

Virtual Reality Applied to Teletesting

T.W. van den Berg^{*a}, R.J.M. Smeenk^a, A. Mazy^b, P. Jacques^c, L. Argüello^d, S. Mills^d

^aTNO Physics and Electronics Laboratory, The Netherlands

^bCentre Spatial de Liège, Belgium

^cSpacebel, Belgium

^dESA/ESTEC, The Netherlands

ABSTRACT

The activity “Virtual Reality applied to Teletesting” is related to a wider European Space Agency (ESA) initiative of cost reduction, in particular the reduction of test costs. Reduction of costs of space related projects have to be performed on test centre operating costs and customer company costs. This can be accomplished by increasing the automation and remote testing (“teletesting”) capabilities of the test centre.

Main problems related to teletesting are a lack of situational awareness and the separation of control over the test environment. The objective of the activity is to evaluate the use of distributed computing and Virtual Reality technology to support the teletesting of a payload under vacuum conditions, and to provide a unified man-machine interface for the monitoring and control of payload, vacuum chamber and robotics equipment.

The activity includes the development and testing of a “Virtual Reality Teletesting System” (VRTS). The VRTS is deployed at one of the ESA certified test centres to perform an evaluation and test campaign using a real payload. The VRTS is entirely written in the Java programming language, using the J2EE application model. The Graphical User Interface runs as an applet in a Web browser, enabling easy access from virtually any place.

Keywords: Virtual Reality, Java, J2EE, Desktop VR

1. INTRODUCTION

As one of the ESA certified test centres, Centre Spatial de Liège (CSL) is specialised in the evaluation of the performances of optical payloads. With current test practices, customer personnel and CSL test facility operators closely co-operate during the execution of tests. Customer personnel – either located at the (remote) client-site or located on-site at the test centre – define the specifications for each individual test. The test facility operator supervises the test execution in accordance with the specifications. At any time, he is the only one who can access the most vital controls of the test facility. If the customer has personnel on-site, the test centre can provide the personnel access to a limited set of less vital controls (e.g. controls that alter the internal state of the payload). For every individual control action to be performed by the customer personnel, the test centre must explicitly provide the personnel the rights to perform this action. This is because the test centre is fully responsible for all incidents that occur during the execution of the tests.

Two main problems with current test practices are:

1. Separation of control between test facility operator and customer operator

The test facility operator monitors and controls the test facility from a control room. From here the operator issues commands to the vacuum chamber and robotics systems inside the vacuum chamber. The operator is not directly aware of the effects his commands may have on the payload inside the chamber. The customer operator monitors and controls the payload through his own dedicated computer systems, with no control over the vacuum chamber or robotics systems inside. The operators use their own unique man-machine interfaces to monitor and control the equipment. Close co-operation between customer and test facility operators is therefore required.

* vandenbergt@fel.tno.nl; phone: +31 - 70 374 0762; <http://www.tno.nl/instit/fel/felmain.html>; TNO Physics and Electronics Laboratory, P.O. Box 96864, 2509 JG The Hague, The Netherlands.

2. Lack of situational awareness

Both the test facility and customer operator cannot see inside the vacuum chamber, even on site. A vacuum chamber has mostly no window. If a chamber would have a window it may be small and the viewing angle limited. The presence of a camera would interfere with the test, therefore no camera is allowed in the chamber either.

Current man-machine interfaces are mostly numerical or 2D schematic. The man-machine interface for the test facility operator in the control room only shows the setpoints of the automatic controllers in the vacuum chamber (e.g. the instrument position controllers) and the raw data from the numerous sensors inside the chamber (instrument position sensors, temperature sensors, etc.). Line graphs are used to show the history of sensor values. In order to remain sufficiently aware of the situation inside the chamber, the test facility operator must mentally construct a visual image of the inside area of the chamber, complicating diagnosing unexpected test results. It is difficult to determine whether unexpected results result from incorrectly configured instrumentation, incorrect test specifications, a malfunctioning payload, or malfunctioning instrumentation. Current man-machine interfaces have no capabilities to visualise the spatial arrangement and movements of instrumentation and payload in the vacuum chamber.

The objective of the activity is to evaluate the use of distributed computing and Virtual Reality (VR) technology to overcome these problems. These technologies enable the visualisation of the inside area of the test chamber and enable test facility and customer operators to simultaneously and possibly remotely monitor and control tests under vacuum conditions. The activity includes the development and testing of a “Virtual Reality Teletesting System” (VRTS). The VRTS is deployed on the computer infrastructure of CSL in order to perform an evaluation and a test campaign using a real payload in a vacuum chamber. The VRTS is entirely written in the Java programming language, using the Java 2 Enterprise Edition (J2EE) application model.

This paper describes the rationale behind the technology used for the development of the VRTS, provides an overview of the VRTS and its operational concept, describes the demonstration and finishes with conclusions.

2. DEVELOPMENT RATIONALE

2.1. J2EE technology

With the Java 2 Enterprise Edition (J2EE)¹ application model, an enterprise application results from the assembly of several *components* distributed across multiple tiers: the *client* tier (with client/server presentation logic), the *middle* tier (with business logic), and the *enterprise information system* tier (see Fig. 1).

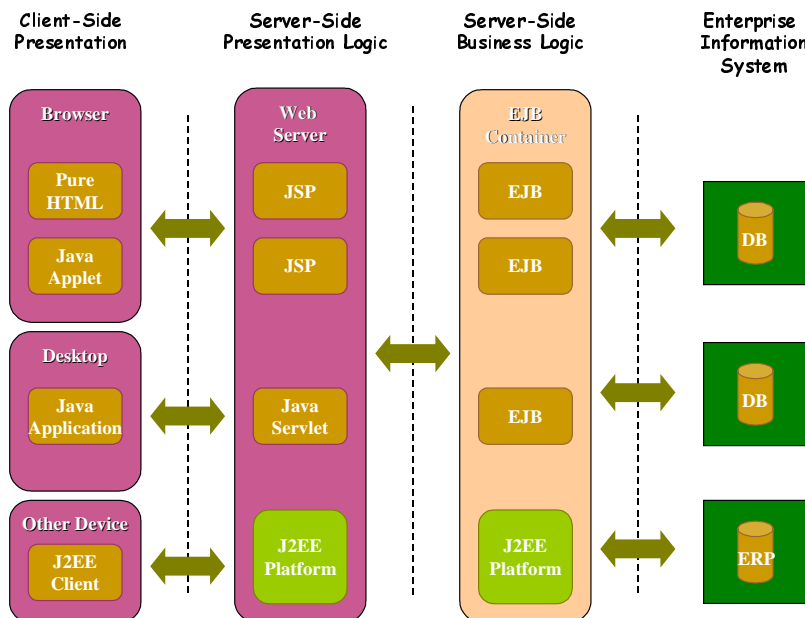


Fig. 1. J2EE application model.

Some main benefits of a J2EE based application are: productivity of Java, development of presentation and business logic only, reuse of existing components, portability of components, unified security model, etc, etc. The VRTS is a distributed system that has to provide monitoring and control functions of a combined test environment (CSL and Customer) to several actors (operators, observers, evaluators) connected through the public Internet (i.e. the World Wide Web) without compromising security. It is a combination of many heterogeneous computing platforms (Web clients, local clients, VRTS servers, ECS systems, ...) each with their own specific requirements but all interconnected using the standard Internet protocol suite. The VRTS must be highly flexible and re-configurable to support a multitude of test campaigns conducted by a variety of customers. The technology of choice for building applications with such demanding requirements is the J2EE technology.

2.2. Virtual Reality technology

Virtual Reality (VR) technology offers the opportunity to visualise payload and instrumentation in 3D and to completely immerse a user in the *virtual environment* (e.g. by means of CAVE technology or a head-mounted display). This *immersive VR* is most useful in applications where it is desired that the users feel as if they are present in the virtual environment (e.g. when practising real-world tasks in a training simulator). During teletests, the test operator and the customer personnel do not have to feel as if they are present inside the test chamber. VR technology is only needed because of its ability to visualise the payload and the instruments in the test chamber in true perspective. *Desktop VR* offers a more practicable solution here. With desktop VR, a desktop computer shows the user the view of the virtual environment. Desktop VR is less expensive than immersive VR. Besides, multiple users (e.g. the customer personnel during teletests) can observe the view of the virtual environment at the same time. Therefore, the VRTS uses desktop VR.

Distributed or networked (desktop) VR is realized by the application of the J2EE application model described earlier. A Commercial Of The Shelf (COTS) Web browser (Netscape or Microsoft Internet Explorer) provides the Desktop VR graphical user interface, where 3D rendering capabilities are added to the browser through a COTS *plug-in* (3D-MasterSuite for Java from TGS²).

3. OVERVIEW OF THE VRTS

The VRTS system and its actors are shown in Fig. 2. At the bottom of the figure is the vacuum chamber and its associated robotic systems which are all controlled by the Equipment Control System (ECS).

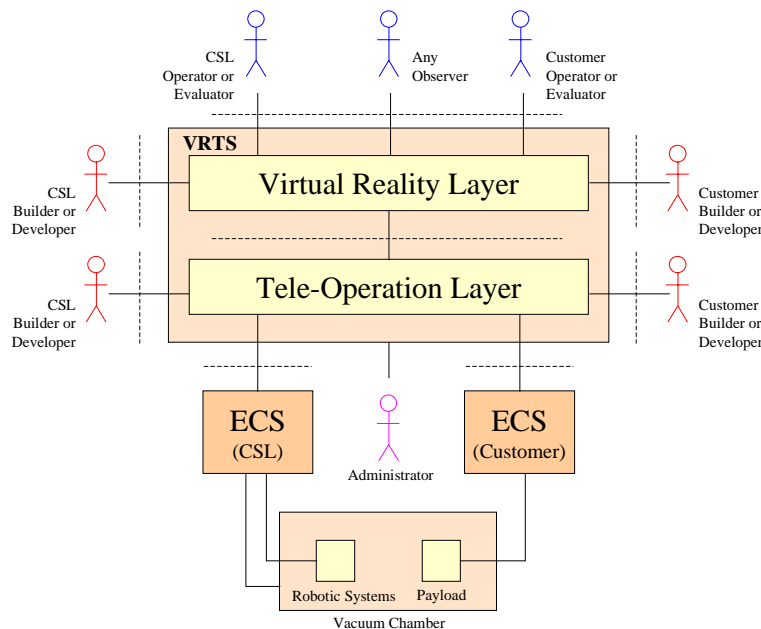


Fig. 2. Overview of the VRTS.

The VRTS consists of two layers:

- The Tele-Operation Layer (TOL) provides a unified monitoring and control interface to the vacuum chamber, robotics and payload equipment. It relies on the underlying ECSs to perform the low-level details of state variable monitoring and command execution and to handle other critical aspects. These ECSs are assumed to be fairly autonomous and capable of quickly reacting in an emergency situation. The TOL is responsible for performing run-time checking and order authorization, device status monitoring and the scheduling of actions.
- The Virtual Reality Layer (VRL) provides the advanced graphical user interface with 2D and 3D rendering capabilities to monitor and control tests. From the VRL user interface the operator can, amongst others, initiate orders, terminate actions, monitor the progress of an action and monitor the device status. Order authorization is left to the TOL, although the possible set of orders an operator can submit is role-dependent.

The TOL covers the J2EE middle and enterprise information system tiers. The VRL covers the J2EE client tier. The VRL consists of a number of VRTS Presentation Clients (COTS Web browsers, each equipped with a VRTS specific VR applet) and a VRTS Presentation Server (a COTS Web server hosting a VRTS specific VR servlet). VR applet and VR servlet communicate via HTTP to enable access through possible firewalls.

The VRTS divides its control functions into:

- Preparation functions: these functions are used during the preparation of a test campaign. Functions include the modelling of the environment, such as the modelling of actions and device features.
- Execution functions: these functions allow the execution, control and monitoring of a test campaign. This includes functions to enter actions and to visualise device status and device status history data.
- Evaluation functions: these functions enable the evaluation of test results, such as device status history data, images and logs.

Both layers include preparation, execution and evaluation functions.

4. OPERATIONAL CONCEPT

The VRTS has three operational phases, in which there are different actors. Summarized there are the following actors:

- “*modeller*” = Builder, Developer (preparation phase)
- “*operator*” = Operator, Evaluator, Observer (execution/evaluation phase)

The operational concept is based on the following key points:

- the *operator* tele-operates via “VRTS Presentation Client”, running on a desktop PC as COTS Web browser;
- the *operator* is provided with interactive control and monitoring functions;
- the *operator* is provided with an easy to use graphical user interface with 2D and 3D rendering capabilities (desktop VR);
- the 3D virtual environment is built out of models of devices and updated on-line by status data via the tele-operation layer;
- the *modeller* is provided with a software-programming interface (VRL and TOL) and database population tools (TOL) to incorporate models of actions, features and devices (which are groups of features). Actions have pre, post, keep and prevail conditions to specify what will hold before, after and during an action³.

4.1. Preparation phase

In this phase the modeller models the environment. Modelling is done with the help of COTS tools, such as 3D Studio Max for the 3D geometry models, JBuilder for the Java class generation and Microsoft Access for database population. Re-use of the various Java classes is stimulated with the help of object-oriented technology. The VRTS provides for example Java base-classes that the modeller only needs to extend and adapt where necessary.

4.2. Execution/evaluation phase

In the execution/evaluation phase the operator tele-operates via the VRTS Presentation Client, running on a desktop PC as COTS Web browser with a VRTS specific applet. The operator is provided with interactive control and monitoring functions. These functions are available via an easy to use graphical user interface with 2D and 3D rendering

capabilities (desktop VR). The 3D virtual environment is built out of models of devices and updated on-line by status data via the tele-operation layer. The main features of the VRTS Presentation Client graphical user interface are described in the next section.

4.3. VRTS presentation client graphical user interface

This section summarizes some of the main features of the VRTS Presentation Client graphical user interface.

4.3.1. Main VRTS presentation client windows

The main VRTS windows are as follows.

- The *System Status window* provides information about the test (name, description, system state, etc.). From the System status window the logging and replay windows can be opened.
- The *Devicelist window* provides a system hierarchy by means of a tree (similar to Windows Explorer), where tree nodes can be expanded or collapsed to hide or show information. From the tree orders may be submitted, actions stopped or device and action properties be requested.
- The *3D Sceneview window* provides a 3D view on the environment with many controls to navigate through the virtual environment (e.g. zoom, pan and set/goto home position). Most actions that can be performed from the Devicelist window can also be performed from the Sceneview window. The virtual environment contains both *symbolic devices* (pumps and valves of the pumping system) and *physical devices* (vacuum chamber, ion beam gun and robotic arm). The spatial arrangement of the physical devices reflects the actual arrangement of the devices in the vacuum chamber and cannot be changed by the user. The user may change the spatial arrangements of the symbolic devices.

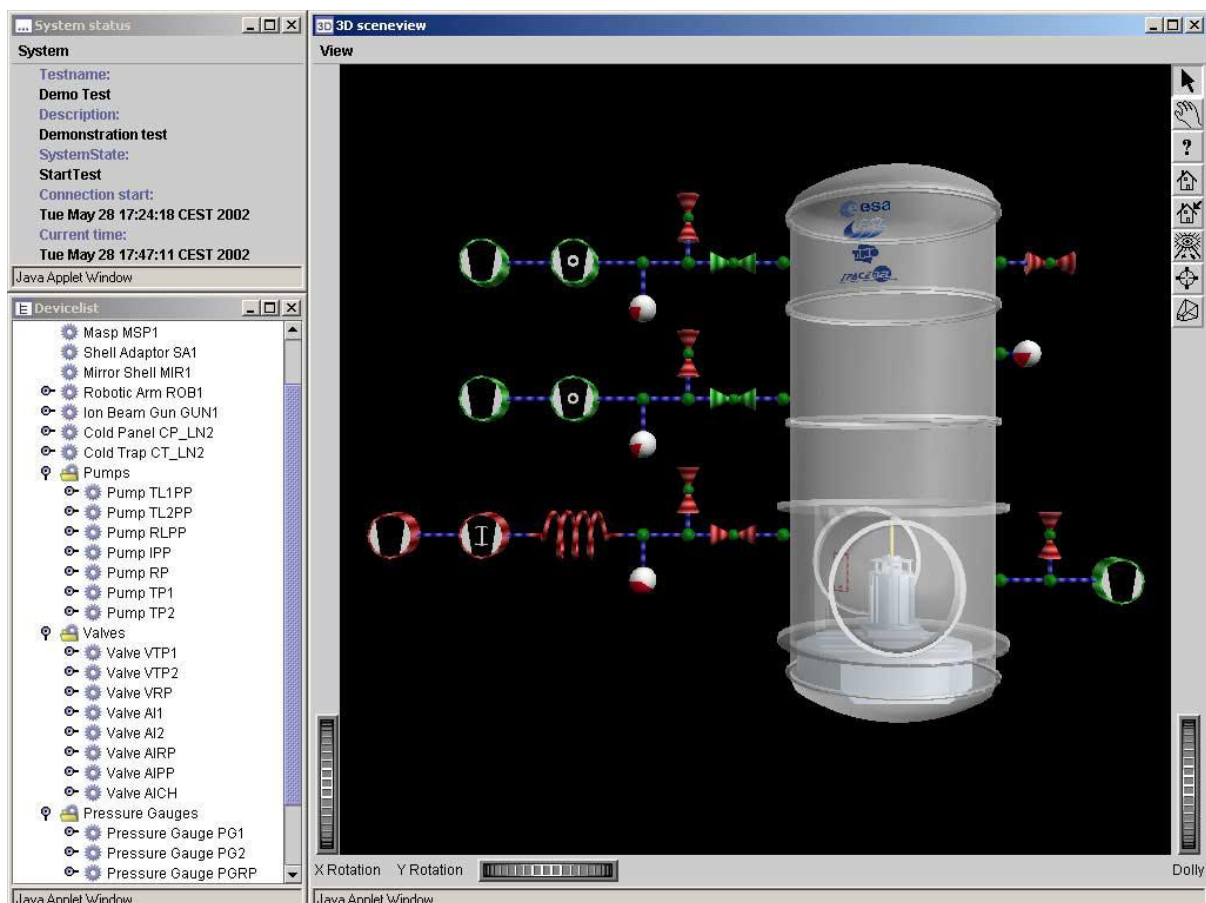


Fig. 3. The VRTS Presentation Client user interface with the following windows: the System Status window, the Devicelist window and the 3D Sceneview window.

4.3.2. Views

The Point Of View (POV) and the 3D arrangement of devices in the 3D sceneview window are called a *view*. It is possible to change the POV and to re-arrange the devices in a view by hiding, unhiding, scaling, rotating or re-positioning devices. A new view can be saved under a name and restored later when required. By using views a user can quickly jump to a predefined arrangement of devices. The views are stored per user on the server so that the user has access to his views from any client. Fig. 4 gives an example of some views.

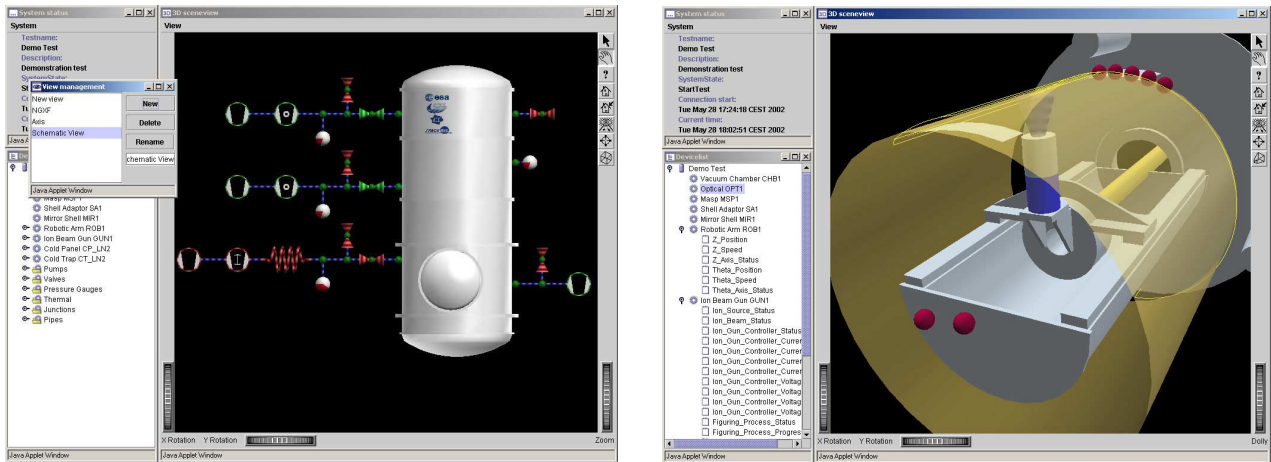


Fig. 4. The schematic view (left) and view on Ion Beam Gun in operation (right).

4.3.3. Complex and nested devices

A *complex* device is a device with a lot of features that cannot be visualised all at the same time. A well-known technique to visualise complex devices is by expanding the device, where the device geometry is replaced by a more detailed geometry of the device.

A *nested* device is a device whose interior contains other devices. A technique that is applied in the experiment (see section 5) uses a combination of device transparency and device hiding to visualise nested devices. By making a device transparent one can see the interior of the device. By hiding a device the exterior of the device becomes invisible. Fig. 5 gives an example of device transparency and hiding.

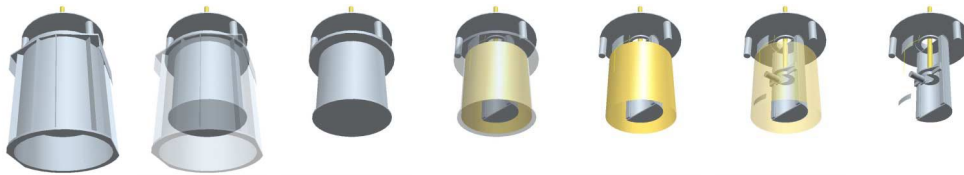


Fig. 5. Example of device transparency and hiding.

4.3.4. Visual clues

Visual clues can be used to increase the understanding of the virtual environment. A visual clue is for example the orientation or bounding box of a device. Other types are visual clues for depth perception, such as distance lines, grids or rulers. Fig. 6 provides an example of visual clues: trail of the Ion Beam Gun and orientation of the Robotic Arm.

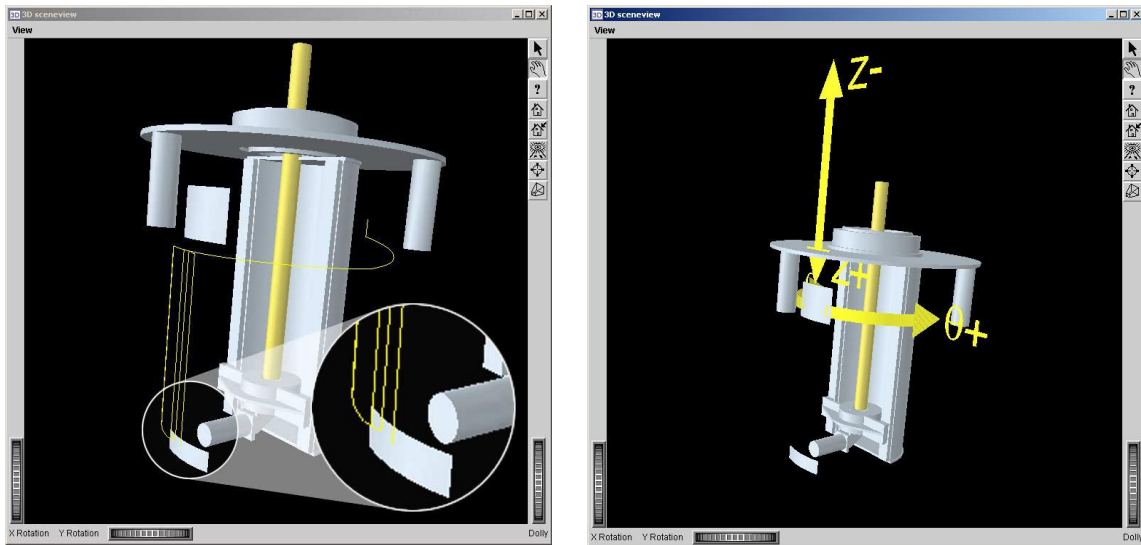


Fig. 6. Visual clues: display of trail of Ion Beam Gun (left) and display orientation of Robotic Arm (right).

4.3.5. Simulation of orders

Predictive simulation is a technique to show the results of an order before it is submitted to the TOL. With predictive simulation problems may for example be detected before the action is executed in the real environment. Fig. 7 gives an example of the simulation of the move arm order for the Robotic Arm device

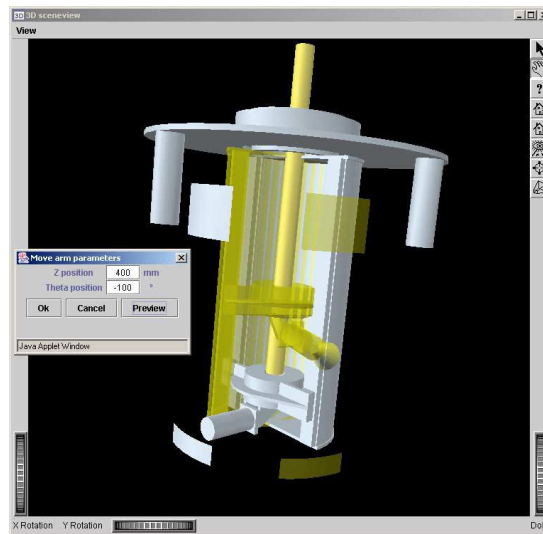


Fig. 7. Preview of Move Gun order.

4.3.6. User roles

The orders that may be submitted are role dependent and based on the login information that a user provides. Fig. 8 gives an example of the Robotic Arm device pop-up menu where non-allowed orders are displayed in gray. Security is based on the J2EE provided security mechanisms.

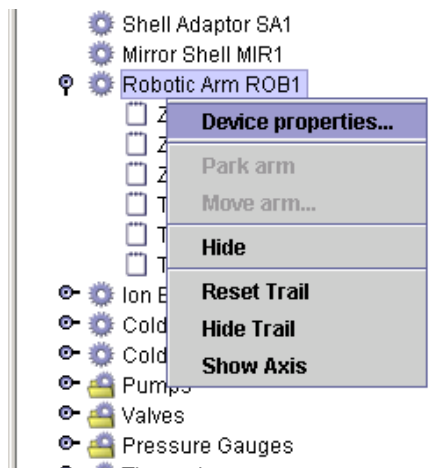


Fig. 8. Role dependent allowed orders.

5. DEMONSTRATION

The objectives of the demonstration can be summarized as follows:

- to exercise the VRTS in a realistic demonstration scenario,
- to evaluate the VRTS functions and performance and
- to assess the benefits of the VRTS.

The experiment that is performed as part of the demonstration is an adaptation and improvement of an already existing Mirror Shell Figuring System, part of the ESA New Generation X-ray mirror manufacturing Facility (NGXF), briefly described here.

5.1. Figuring system

The Figuring System consists in two parts: (1) a Metrology System and (2) an Ion Beam Figuring System. The Metrology System will provide a measurement of the Mirror Shell surface before and after the Ion Beam Figuring process dedicated to improve the shell figure. Ion Beam Figuring process takes place under vacuum condition. The Metrology System will work on atmospheric pressure. Figuring is done in three steps:

1. a metrology run,
2. an ion figuring run,
3. a second metrology run.

The experiment that will be used for the demonstration of the VRTS is the Ion Beam Figuring process. However, we will also shortly present the metrology system.

5.1.1. Metrology runs

A dedicated metrology machine performs the metrology runs. It is performed at the atmosphere. It returns the mirror radius variation from theoretic telescope. The aim of the first metrology run is to determine the amount of matter to remove and compute the dwell-time of the ion source. The aim of the second metrology run is to check if the matter was correctly etched from the mirror. The output of the metrology run is a matrix containing the local radius of the mirror shell sampled every degree on theta axis and every 2mm on Z-axis (TBC). This matrix is then provided to dedicated software that will compute the trajectory of the Ion Beam Gun for the Ion Figuring process.

5.1.2. Ion figuring run

The ion figuring run is performed under vacuum conditions (vacuum conditions are required to produce the ions) in the Focal-X facility (see Fig. 9). The Ion Beam Gun removes matter from the Mirror Shell. The amount of matter removed from the Mirror Shell is proportional to the total time spent at this position. However, the process generates heat. To avoid high temperature that could damage the mirror shell, the Ion Beam Gun is moving continuously and will pass

more often and more slowly where lots of matter must be removed. The speed and trajectory of the gun is computed offline after the metrology run.

The main parameters that influence the ion source's speed are:

- the shape errors, linked to the mirror, and
- the ion beam erosion function, which varies with the ion energy.

5.2. Focal-X facility

The vacuum chamber of the facility is 12 meters high and 4.5 meters in diameter. The facility is monitored by hundreds of sensors of different types (temperature, pressure, valve status, pump status, ...), connected to automatons, dataloggers, etc. These devices are then connected to a central workstation gathering all the information in a real-time database. A large number of devices can be commanded automatically. Only a subset of the available sensors, valves and pumps is used in the experiment.

5.3. Scenario

The demonstration scenario consists of starting a set of pumps in a well-defined sequence, opening a set of valves and starting the cold trap. Once the vacuum chamber has reached the right pressure and temperature the Ion Beam Figuring run is started. When figuring ends, the vacuum chamber is brought back to atmospheric pressure and room temperature.

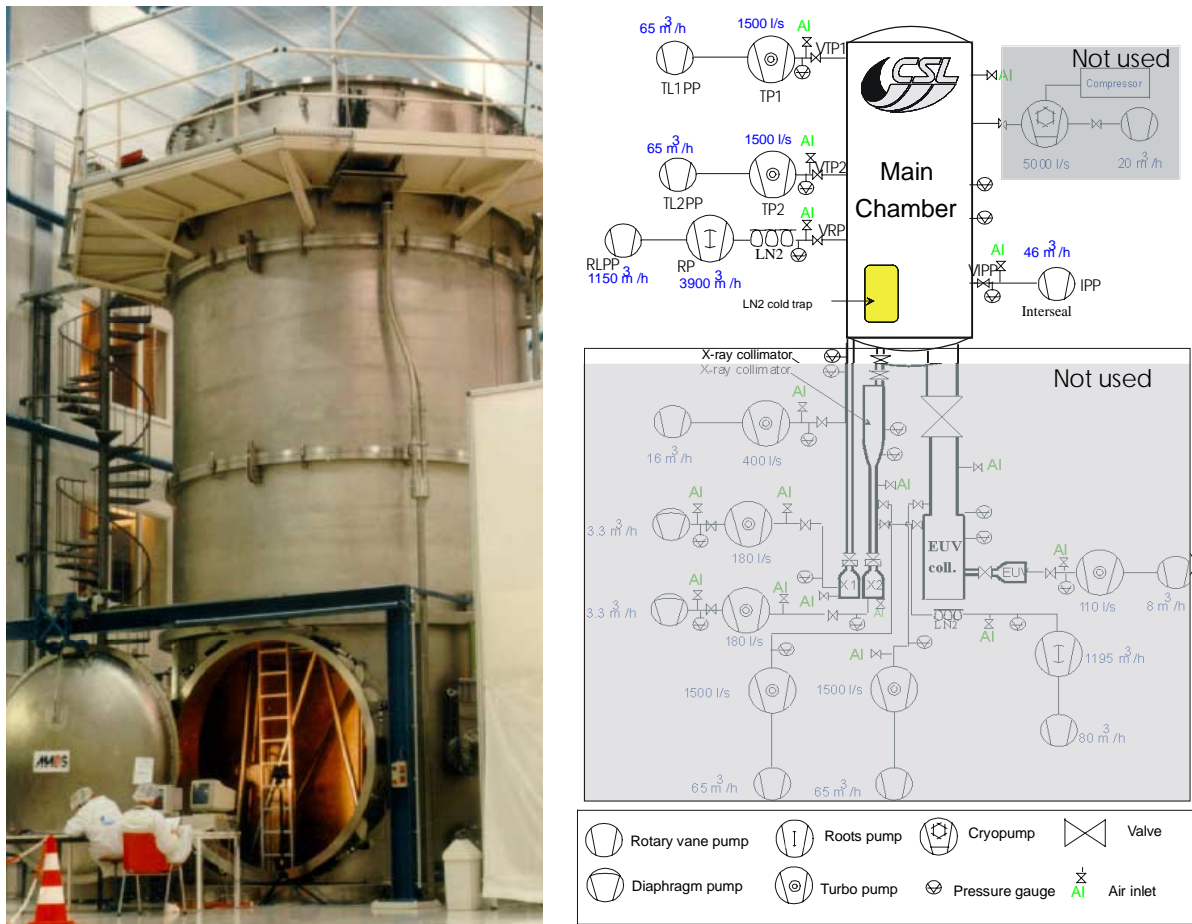


Fig. 9. Focal-X vacuum chamber (left) and schematic overview (right).

6. CONCLUSIONS

The objective of this activity was to investigate and demonstrate the possibility to use distributed computing and Virtual Reality technology to control and to monitor the testing of a payload under vacuum conditions in the Centre Spatial de Liège (CSL). A prototype system based on commercial off the shelf J2EE technology was implemented and demonstrated in 6 months.

A demonstration took place in Liège where it was possible to control the pumping sequence in Focal-X (one of the vacuum chambers at CSL) with a standard desktop PC in CSL and control the NGXF experiment inside the vacuum chamber via another PC accessing the VRTS through the Internet. A number of operational and technical issues as well as possible improvements have been identified. Within the limited scope of the prototype and the test campaign, it was demonstrated that tele-operation and Virtual Reality technology can be effective for the remote monitoring and control of vacuum chamber tests.

The Virtual Reality techniques utilised in the prototype system have shown to be very useful in understanding the situation inside the vacuum chamber. During the demonstration the various graphical user interfaces provided efficient means to obtain an overview of the thermal behaviour of instruments and to visualize the movement of robotic devices. Virtual Reality should not be seen as a replacement of existing interfaces, but as an addition to existing interfaces. It is complementary.

A main challenge for the future will be the application (such as remote monitoring) of a VRTS-like system on other payloads in an operational environment, using the experience gained in this activity.

ACKNOWLEDGMENTS

The Prime Contractor for the activity is CSL, a research Centre for the University of Liège (Belgium) and one of the ESA Coordinated Facilities. Acting as Sub-Contractors are TNO-FEL (Netherlands Organisation for Applied Scientific Research) and Spacebel (a Belgium Information Technology solution provider). Activity funding is by ESA/ESTEC.

REFERENCES

1. Java 2 Platform, Enterprise Edition (J2EE); <http://java.sun.com/j2ee>.
2. TGS, <http://www.tgs.com>.
3. C. Bäckström, *A representation of coordinated actions*, Technical Report, Linköping University, 1988.