

Eurosim and its Applications in the European Robotic Arm Programme

In this paper the commercial RT Engineering tool EuroSim will be introduced. EuroSim is used as the primary simulation framework for the ERA Simulation Facility (ESF) programme. Both the European Robotic Arm (ERA), ERA Simulations Facility and their role in the various development are described. The ESF is used throughout the ERA design, verification and operations phase. In the operational phase ESF forms one of the key elements for the Mission Preparation and Training Equipment (MPTE). The MPTE will be used throughout the operational of the International Space Station (ISS).

INTRODUCTION

Fokker Space is the prime-contractor to ESA for the development of the European Robotic Arm (ERA). ERA will be used for the assembly and servicing of the Russian Segment (RS) of the International Space Station (ISS). Its operational capabilities include the manipulation of large and heavy payloads (such as space station elements), replacement of small and large equipment boxes (ORUs), space station inspection services, and support of cosmonauts during Extra Vehicular Activities (EVA). The ERA will be commanded and monitored by a cosmonaut, either internal (IVA) or external (EVA) to the space station, in several real-time control modes from the control panel Man Machine Interface (MMI). ERA is capable of manoeuvring without a direct cosmonaut view on the scene and with limited operator supervision. Therefore ERA has been designed to be operated both fully automatic and manual.

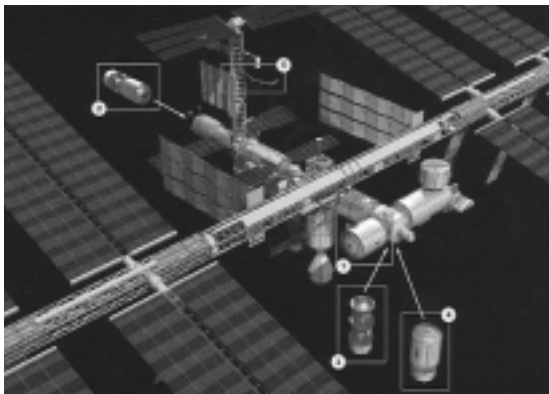


Figure 1. ERA on International Space Station.

The complete ERA system comprises the following elements (see figure: 2):

- a relocatable manipulator
- two consoles for EVA control: EVA-MMIs
- a console for IVA control: IVA-MMI
- a set of basepoints on the Russian Segment of the ISS
- a ground segment with, in particular, the Mission Preparation and Training Equipment

In the design and verification approach of ERA, simulation is considered to be an important technology. Of course, for the design specification of motion control algorithm's, simulation is an obvious technology. Moreover for the verification of space manipulator motion control algorithms in three-dimensional space, simulation is considered to be the only feasible way. In this sense simulation complements the flat-floor tests on the ERA Test Facility (ETF).

Throughout the last decade, Fokker Space has developed the 'European Real-Time Operations Simulator (EuroSim), a unique and powerful RT Engineering Simulation tool. This tool was used as the simulation framework for two major simulators, the ESF and the MPTE, which are used within the ERA programme. These two simulators support design specification, Software in-the Loop and Hardware in-the Loop testing of flight critical components, and the operational use of ERA.

In this paper some details of the EuroSim and the ERA system are described in light of ESF. Furthermore the use of ESF within the ERA programme is discussed, as well as the simulation models resident within ESF. Another EuroSim application in the ERA programme is the Mission Preparation and Training Equipment (MPTE). The MPTE is developed by the Dutch Aerospace Laboratory NLR and will reuse the ESF as well. The concept of the MPTE system and its visualisation approach and possible future extensions are also discussed in this paper.

EUROSIM

Traditionally, simulators are custom built and quite a number of different simulators are required to meet all of the simulation needs of a space project. In order to help reducing the cost of simulation, EuroSim was developed as a standard simulation tool, that can be employed throughout life-cycle of a (space) project and that will support the re-use of simulation models between different (space) projects.

EuroSim is based on the principle that each simulator can naturally be broken down into an invariant tool part and a part that is specific to the subject being simulated: the latter is called the (simulation) model. The model is a software representation of the behaviour of a particular real-world system. By means of a careful

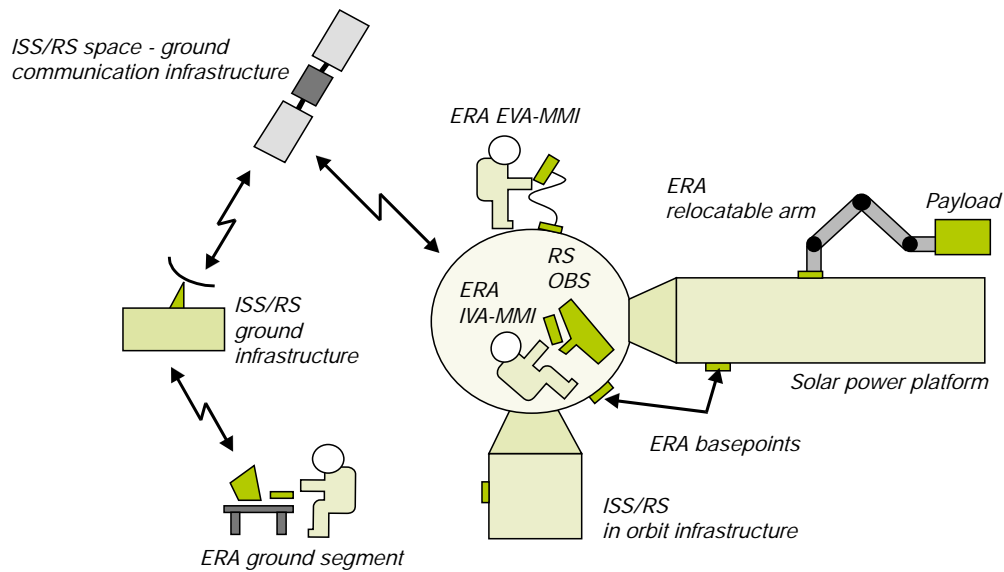


Figure 2. ERA system schematic overview.

design of the tool part, it can be used for both small and large simulators, and it can support the portability of simulation models, allowing them to be re-used between projects and to exchange one model for another, e.g. one offering higher fidelity. Having such a tool available will help to reduce the cost associated with simulation and/or allow it to be used more extensively and earlier in a space project. It will also help to concentrate the effort on model development itself, i.e. the project specific part of the simulation.

Thus EuroSim is a complete real-time simulation environment, supporting the user from the initial task of developing an application model that simulates some real-world system, to the final task of analysing the results (Brandt 1994). The main capabilities of EuroSim are:

simulator development: the user is involved with developing (or porting) source code for an application model and associated files such as geometry data for image generation, operator screen definitions, hardware device drivers; assembling the application/simulation model from individual sub-models, and defining a schedule for execution of that model; creating an executable version of the simulator (or a procedural definition for doing so);

test preparation: the user defines test conditions for a particular simulator, which includes definitions for the scenario (to trigger events such as failures or malfunctions), stimuli (to replace missing real-world inputs), initial conditions (e.g. payload, configuration of robot arm), and data recording and on-line monitoring requirements;

test execution (simulation): the user is responsible for controlling the execution of simulation runs, both in terms of simulator state (e.g. running, paused, stepping), and also by being able to trigger additional events to those planned in the scenario; the user can monitor the simulation-run through display of model data values, and can receive copies of image/audio information which is being presented

at the operator workstation; during test execution, interfaces to man-in-the-loop and hardware-in-the-loop can be active, as can interfaces to other simulator facilities;

test analysis: this is concerned with processing and analysing simulation-run results; also the replay of simulation run images, e.g. camera images, is possible;

facility management: this activity supports the other (mainstream) ones: managing access to the facility; ensuring that the facility remains in an available state and the critical function of providing configuration control of model and tool software within the facility.

In EuroSim, all phases of the simulation life cycle are supported by graphical interactive tools: the user can operate the complete facility through 'pointing & clicking.'

EuroSim is based on client/server architecture and a UNIX operating system. The server runs the actual real-time simulation process, i.e. executes the simulation model according to a user-defined schedule (see "Dynamic scheduling and Scheduling support"), provides the required stimuli and records the selected data variables. It also executes the user-defined scenario ("MissionTool and the Mission Definition Language"). To the server, clients can connect as needed, e.g. to provide an interface to a specific hardware item, another simulator or to a 'man-in-the-simulation-loop'. The latter usually takes the form of an image generation system with displays and control/input devices. Within EuroSim, also the standard facilities such as the (simulation) test control interface and on-line data monitors are implemented as clients.

The client/server design allows for a large amount of flexibility, with all processes (server and clients) running on an entry level, single-processor workstation for a simple 'desk-top' simulation, all the way to having the

EuroSim server running on a top-of-the-line multi-processor 'number cruncher' with the clients running on dedicated workstations/equipment, connected to the server via a high-speed network. Because EuroSim will distribute the load over the available processors in the server, larger simulation models can be accommodated simply by adding more processors, without having to reprogram any of the software.

In case of ESF and MPTE four- and six- processor Silicon Graphics Onyx computers are used in some cases with powerful Reality Engine II/ Infinity Reality visualisation hardware and software extensions.

EuroSim was developed by Fokker Space, Origin and the Netherlands Aerospace Laboratory (NLR) and ongoing development will take place in the next years. It is commercially available through Fokker Space.

Some unique EuroSim features of major importance to the ESF programme are highlighted in the following sections.

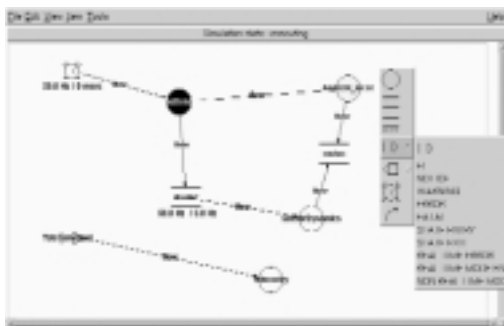


Figure 3. EuroSim's ScheduleEditor.

Dynamic scheduling and scheduling support

As mentioned the application/simulation models are built-up from submodels. Sub-model's entry points (i.e. functions without arguments) can be combined into tasks for sequential execution. The tasks themselves can be scheduled for activation in a number of ways, as is depicted in figure 3. Tasks can be activated depending on a periodic timer, on another task's completion or when a certain (user defined) event occurs. Because EuroSim supports multi-processor computer architectures, there's also a scheduling primitive provided (represented by an icon) to make tasks mutually exclusive, so that they will not run simultaneously.

Tasks can be given various attributes. The minimum, maximum and average task and entry point execution times can be entered manually, but can also be obtained from a previous simulation run with this task. Based on this data, EuroSim can display a time bar, showing the execution of the various tasks within the schedule; an example is provided in figure 4.

The time bar displays the time range and deadlines within which the tasks will execute. As the EuroSim scheduler is a truly dynamic scheduler, the actual execution of tasks can vary between the indicated boundaries, depending on the variation of actual execution times. The scheduler will distribute the 'ready-to-run'

tasks over the available processors (unless of course the user has specified that a task has to run on a specific processor) to make sure that they have been completed before their deadline.

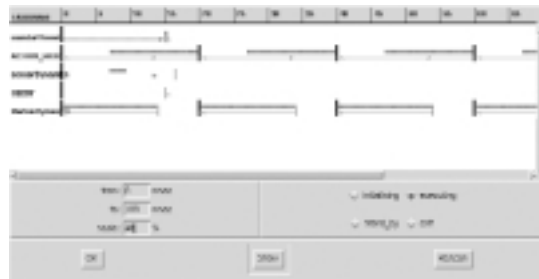


Figure 4. Pre-run display of a EuroSim schedule.

Great care has been taken to minimise the jitter, i.e. the random variation in a task's activation time. On a multi-processor machine, EuroSim can allocate a user-specified number of processors for exclusive (hard real-time) use, giving the EuroSim scheduler full control over them, bypassing the UNIX operating system. On a single processor machine this is not possible, but if the machine is not too loaded, remarkable deterministic real-time performance can still be achieved.

EuroSim can furthermore run in a non and a faster than real-time mode. The former is still trying to meet all real-time constraints, but will not claim as much of the computer resources as the real-time mode does. This mode is therefore ideal for model development purposes and a number of such simulations can be accommodated on a single machine simultaneously. The faster than real-time mode simply executes the model as fast as the processors can manage, disregarding all requirements regarding real (wall clock) time. Timing constraints of tasks relative to each other are - of course - kept intact.

Mission tool and mission definition language

A simulator consists of an invariant tool part and a model. The simulation results can differ from run to run, because of different parameter settings in the simulation model, different inputs to the simulator (either coming from a file, hardware device or human operator) and - within EuroSim at least - because of a different scenario being loaded and executed. A scenario is list of events and corresponding actions. An event is defined as a Boolean condition, that EuroSim is constantly evaluating at a certain (high) frequency. When the condition evaluates 'true', the event has occurred and the corresponding action will be executed immediately.

Both events and actions can be pre-programmed and loaded from file when starting a simulation run or they can be added while the simulation is running. Programming is done using EuroSim's MDL, or Mission Definition Language. This can be used as a C-like language with all the standard language features, such as constants, arithmetic and Boolean operators and flow control statements. The user can create vari-

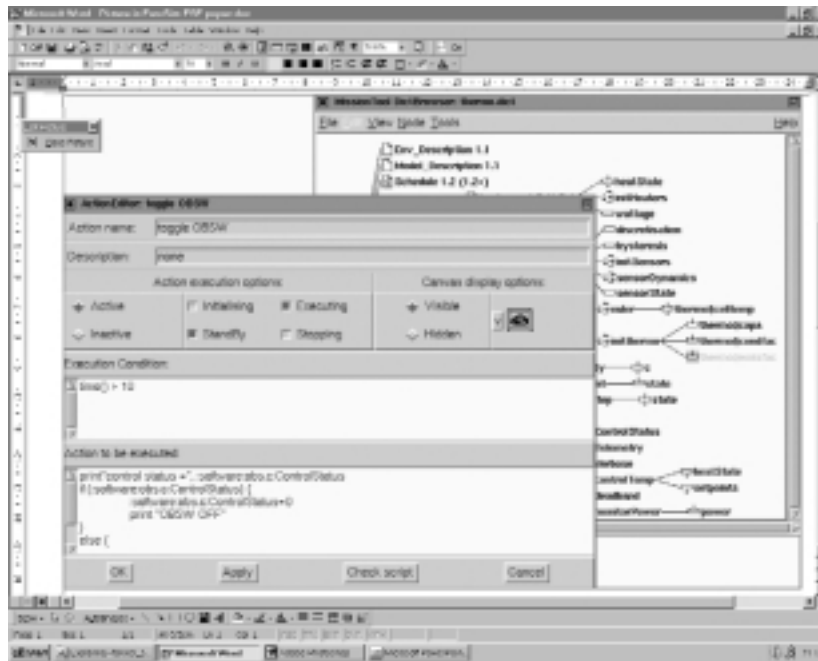


Figure 5. EuroSim's DictBrowser and ActionEditor, displaying an MDL example.

ables as needed, but can also use the variables contained within the model's data dictionary. To complement the MDL, a suite of built-in functions is provided. This suite contains not only the usual mathematical and signal-generating functions, but also functions to control the state of the simulator itself.

The MissionTool makes testing different simulation scenarios very easy, but its unique features also make it possible to run simulations with only a unit or sub-system present of the complete system to be simulated eventually. This allows for the earliest possible use of simulation in project's life cycle as well as for the validation of the different sub-models in the simulation. When a unit/sub-model is run in isolation, it can be stimulated - i.e. receive its inputs - either from one of MDL's signal generating functions or from pre-recorded test data. The latter could e.g. originate from hardware tests on that unit and would allow the validation of that unit's simulation (sub-) model.

External Interfaces

An important aspect of almost any system is the way it can interface and cooperate with other systems. This section discusses EuroSim's on-line interfaces. A number of them are available.

First, EuroSim offers support - through the EuroSim library - for a number of 'physical' or hardware interfaces: serial RS232/422, VME, GPIB, external interrupts and the MIL1553B data bus (both as bus controller or remote terminal). These hardware interfaces make it possible to build mixed hardware/software simulations. Early in a project's life cycle, the simulation will be fully software based. In later stages, when (breadboards of) hardware units become available, software sub-models can be replaced by this actual hardware. This will then

allow the testing of this hardware in a closed loop manner and it will increase the fidelity of the simulation itself.

Secondly, EuroSim offers an important 'software' interface, which is called ExternalSimAccess. This interface allows the user - with remarkably little programming - to map some or all of EuroSim's model data dictionary variables into the address space of his application. This application then effectively becomes a client to the EuroSim simulation server, and can - just as the standard EuroSim clients such as the TestController - be run either locally or remote. When run remotely, the connection between the server and the client is TCP/IP based. ExternalSimAccess makes it easy to realise a distributed application over a network, e.g. for multi-user simulations. By the mapping of data dictionary variables into the client's address space, the client can simply read and write these variables as if they were local variables. EuroSim itself is using the ExternalSimAccess mechanism for the interface (back and forth) with the Image Generation System. This system is of course driven by the data from the simulation model, but can report back events like a collision occurring between two geometry objects.

Finally, EuroSim's DIS - Distributed Interactive Simulation interface should be mentioned. This interface based on the VR-Link package allows EuroSim to take part in simulations performed through the cooperation of a number of simulators, each simulating a particular system (often a vehicle) in the simulation scenario. EuroSim has also been extended to operate in a High Level Architecture (HLA) distributed simulation environment.

THE EUROPEAN ROBOTIC ARM: ERA

The ERA is a 11.3 meter long symmetric relocatable manipulator arm (see figure: 6), with at both hands a shoulder/wrist assembly containing a pitch, yaw and roll joint (Boumans 1998). The two carbon fibre limbs, with in between an elbow pitch joint, connect the two hands. Next to the elbow-joint the ERA Control Computer (ECC) is located. The ECC runs all the software to operate the arm. Both hands are equipped with an end-effector, which can latch to a grapple fixture of a payload, or to a basepoint. The end-effector contains a connector for power, data and video links, an integrated service tool to mechanically drive a mechanism in the grappled object, a torque/force sensor and a camera and lighting unit (CLU), which can also act as a proximity sensor.

The data communication network for ERA comprises two MIL1553 databuses. The external (or RS) bus for the communication between the ECC and the ISS Central Post Computer (CPC), and an internal (or ERA) bus for the communication between the ECC and its subsystems. Telecommands and telemetry from and to ground (the Mission Control Centre) or MMIs are relayed by the CPC via the RS bus to the ECC. The ECC controls the communication (commands and status data) with the ERA subsystems. The software running on the ECC comprises the following functionality (Beerhuizen 1998):

- handling communication with the Space Station (ground and IVA-MMI)

- maintaining the communication with the ERA subsystems (wrists, elbow, end-effectors, EVA-MMI, payload, CLUs)

- guarding the safety of ERA by performing a number of checks

- controlling the motion of ERA

ERA can operate automatically under cosmonaut supervision of the execution of ground-based prepared AutoSequences or manually by EVA-MMI keyboard inputs. The motion control software comprises three modes: free motion, proximity motion and compliant motion. In free motion, open-loop setpoints are generated based on inverse kinematics of the end-effector Cartesian path to a required end-pose or based on a single joint required end-pose. In proximity motion a visual target is approached based on a feedback loop with proximity sensor processing of a camera image. Compliant motion is used to grapple a grapple-fixture of a payload or a basepoint while controlling exerted forces and torques using torque/force sensor feedback.

The ERA is scheduled for a Space Shuttle flight in the year 2001.

ERA SIMULATION FACILITY

The ESF development started formally, with the ERA kick-off, in 1995. However, several preparatory activities have been performed in the years before, either in the context of the HERA Simulation Facility (HERA was the predecessor of ERA) or on national funding.

ESF is developed in an incremental way. ESF started

as a design simulator, which has now reached the status of verification simulator (in the ESF-HILT configuration) and will evolve towards a simulator for training and mission preparation. A major advantage of the use of EuroSim as simulation environment is that reuse can be made of simulation models and the fact that each ESF configuration can be used and maintained within one single environment.

The ERA Simulation Facility is already being used and will continue to be used in several phases of the ERA project:

- For the design specification of ERA motion control algorithms. In this phase the ECC motion control software is simulated and its performance with respect to the ERA operational requirements, timing and accuracy, is evaluated.

- For the verification of the ECC flight software. In this phase the flight software is connected to ERA simulation models as software- or hardware-in-the-loop. The communication over the RS and ERA MIL1553 buses is verified and performance tests are executed.

- Validation of ESF Models by means of flatfloor ETF test results in order to qualify ERA for space environment

- The validated ERA Simulation Facility will become a core element of the Mission Preparation and Training Equipment (MPTE).

The ESF consists of a number of high-fidelity simulators configured in the generic simulation tool EuroSim.

ESF MODELS

The ERA simulation model as used within ESF is subdivided into a number of models. In this section, each model is described in detail.

Manipulator dynamics model

The core model within ESF is a model of the dynamics of the manipulator. This model is based on a recursive method of solving dynamics equations comprising an open-chain of flexible and rigid bodies as described in (Ellenbroek 1994).

As ESF shall be able to simulate the ERA actions CompliantMove and Grapple, the open-chain dynamics model has been augmented with a contact model and a constrained manipulator-tip motion-model to enable closed-chain dynamics.

For the purpose of ESF, the two long ERA limbs are to be modelled as flexible elements, containing two flexible modes per bending plane and one torsion mode. As additional features, a moving base (6 dof) and a torque/force sensor (6 spring elements) is required. Together, this adds up to in-total 29 degrees of freedom (excluding 7 dof for actuator dynamics) and a wide spectrum of eigenfrequencies.

In order to optimise this model for real-time execution, the set-up of the dynamics equations is split into a slow dynamics part (to compute the generalised matrix of inertia and the centripetal/Coriolis vector) and a fast dynamics part (to solve the dynamics equations). The required execution frequency of the slow and fast

dynamics calculations depends on the required accuracy and the inherently required stability to solve the dynamics equations using numerical time-integration.

Actuator model

The ERA Manipulator Joint Subsystems (MJS) for the Shoulder, Wrist and Elbow assembly contain a joint driving unit. The dynamics model of the mechanical part of the MJSs is represented by the Actuator model.

The ESF Actuator model features a flexible gearbox (giving rise to one extra dof in the total dynamics equations), between the motor and joint axis, including hysteresis characteristics. Furthermore friction and stiction on both the motor axis and the joint axis are taken into account.

Sensor models

Sensors of the interoceptive and exteroceptive type are available within ERA. Interoceptive sensors are used to measure the internal state of ERA. The joint resolver and encoder are, as interoceptive sensors, part of the MJSs to measure the motor speed and joint angles. Within ESF, models of the Resolver and Encoder are resident, where the key characteristics of these models are quantification of the output signal and noise.

In order to measure the external state of ERA, two exteroceptive sensors are available: a proximity sensor and a torque/force sensor. The proximity sensor is part of the CLU subsystem; it determines using a camera device and illuminated target dots, the ERA end-effector pose. In fact, the CLU performs run-length encoding of the camera image, which is subsequently sent to the ECC via the ERA bus for decoding. The torque/force sensor is part of the end-effector subsystem; it measures, via strain gauges, the force and torque vector exerted on the end-effector.

Within ESF two different models of the proximity sensor are available: a model for the ESF-HILT configuration comprising functionality to produce run-length code based on camera characteristics, and a model for the ESF design configuration producing quantified ERA end-effector pose output. The ESF torque/force sensor model transforms and quantifies the in the manipulator dynamics model determined spring deformations.

Joint control model

The ERA angular joint position is locally setpoint-controlled via a closed-loop system comprising encoder/resolver, joint control software and actuator. The joint control software is embedded in the MJSs. The ESF joint control model is a direct copy of the embedded joint control software in C source code.

Communication models

Data communication over the two databuses maintains the transfer of commands and status information between ERA subsystems and the Russian Segment computer.

Within ESF, the communication models simulate the

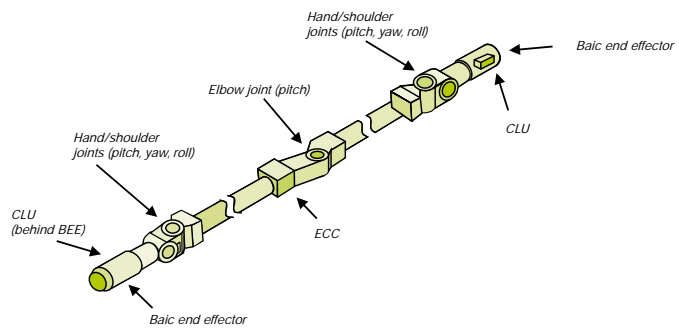


Figure 6. ERA break-down.

MIL1553 databuses via VMIC interface cards in the simulation host computer. On the RS bus the CPC BusController (BC) and the IMMI and EMMI RemoteTerminals (RT) are simulated. The Wrist, Elbow, EndEffector, CLU, Payload, and EMMI RemoteTerminals of the ERA bus are simulated.

In the present context, MIL1553 communication simulation means the transfer of data according to the specified scenario with respect to message lay-out and timing according the ERA specifications. In addition, the BC/RT pertaining subsystems provide and/or accept the data and/or commands involved. For instance, the RS bus BC model, i.e. the CPC simulation model, relays the TM/TC between ground or MMIs and the ECC.

The communications models are used in the ESF-Hardware-in-the-Loop configuration. By means of this model a breadboard of the ECC can be connected to ESF. The application software (ASW), including ERA motion control software, can be run in closed-loop with the ERA manipulator dynamics model.

Onboard software model

The ERA motion control software is part of the Service Layer (SL) of the ASW. The ESF Onboard software model is a functional model of the ERA SL motion control software implemented in C source code.

The Onboard software model can be used without the breadboard ECC and communication models, in non-real-time simulation mode for ESF design simulations.

Hardware-in-the-loop

The ECC breadboard can be connected to the ESF-HILT configuration via the MIL1553 bus interface. In this HILT configuration the functionality of the flight software is verified against the requirements. In this case ESF provides the simulated environment for ECC verification.

Subsequently, the EMMI and IMMI will be connected to the HILT configuration also via the MIL1553 bus interface. This configuration will be used for end-to-end tests with a man-in-the-loop (cosmonaut), and will eventually be the basis for the MPTE.

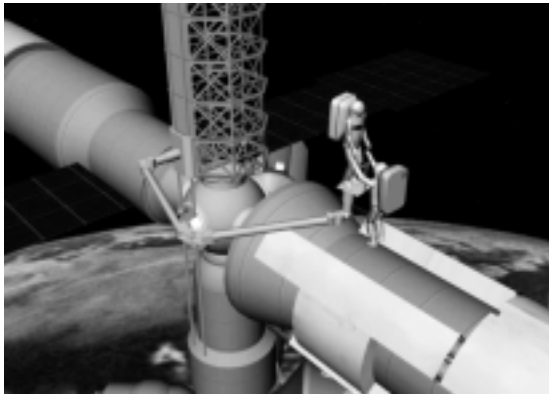


Figure 7. Example of EVA visualisation in MPTE.

MISSION PREPARATION AND TRAINING EQUIPMENT

The MPTE is under development with the National Aerospace Laboratory NLR (The Netherlands), and the Spacebel Trasys Association (Belgium). The MPTE provides the ground support functions for ERA operations, both off-line and on-line:

Mission Preparation

Training of cosmonauts for ERA operation in space

On-line Mission Support

ERA Mission evaluation.

To support mission validation and training the MPTE uses EuroSim, ESF, environmental models representing the space environment and the ISS and the Hardware-in-the-Loop interface with a real ECC, EVA- and IVA-MMI and CPC. It also includes visualisation and facility management. In addition, the MPTE supports flight- and ground operational software maintenance.

The MPTE will be installed at three locations:

Russian Space Corporation/Energia-Mission Control Centre, Korolev, Moscow Region, Russia, to be used for support of flight operations (mission preparation and mission support).

Gagarin Cosmonaut Training Centre, Star City, Moscow Region, Russia, to be used for training of ERA operations.

ESTEC Noordwijk, The Netherlands, to be used for training of Russian instructors, and for maintenance of flight and ground operational software.

MPTE CONFIGURATION

The three MPTE systems will be identical for all three sites, although the MPTE at ESTEC will have been enhanced with flight software maintenance capabilities. Each MPTE has a Silicon Graphics Onyx 2 computer to perform the computational work for the various simulation models and the visualisation image generation. Various images are generated for different purposes:

Single image for training of the cosmonaut during External Vehicular Activities (EVA) with use of the EVA-MMI; this is an image from the cosmonaut point of perspective during the EVA mission

Two images provided to cosmonauts under training for manoeuvring ERA from within ISS through the IVA-MMI; these images are simulations of two cameras at the elbow and the wrist of ERA.

A combination of images provided to the Ground Segment

Overview image for use by the Trainer

One of the images that are generated is intended for training of the EVA cosmonaut operating ERA. This image is projected on a large screen (2.6 meters by 1.8 meters) and must be of a high quality. Other images are presented on conventional monitors.

For the Real-Time visualisation, an Image Generating System (IGS) is used which uses VEGA as visualisation tool while providing customised, enhanced features providing effects such as:

Shadowing

Noise

Eclipse/glare

Focusing

Resolution reduction

Vega is a toolkit based on SGI Performer.

The total Hardware set-up of the MPTE is depicted in the following diagram. The 1553 Space Station communication standard is simulated as Hardware in the Loop, making it possible to use the actual ECC, IVA-

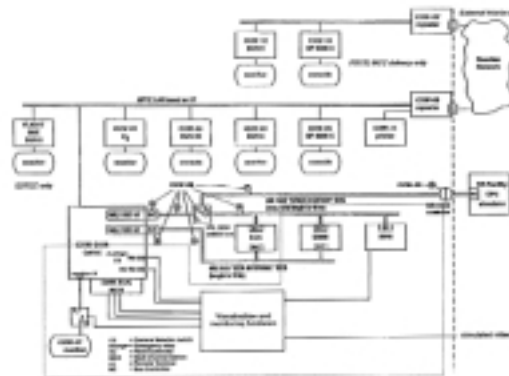


Figure 8. MPTE Hardware set-up.

MMI, EVA-MMI and CPC.

Although MPTE utilises 3D images, extensions to stereo 3D and Virtual Reality are possible for future use in the operational support of ERA.

The MPTE facilities will become available to the ERA project in the beginning of 2000.

CONCLUSIONS

Throughout the engineering, development, verification, validation, qualification and operational phases of the ERA programme simulation has played and will play a major role. During the project life-cycle, EuroSim has been successfully used as the Real-Time simulation framework for a number simulation facilities of which ERA Simulation Facility and Mission Preparation and Training Equipment are highlighted. Especially in the

latter, visualisation plays an important role. This equipment uses high-end SGI Onyx computers which are used to provide high quality visualisation in 3D, with possibility of extension to stereo 3D and in the further future possibly Virtual Reality ■

*U.G. Termote, D. J. Schulten,
R.M. Lieuw and M.J.H. Couwenberg,*

REFERENCES

Beerthuisen, P.G., C. Maegaard and A. Rusconi. 1998. "ERA Safety Strategy." In DASIA 98 (Athens, Greece 25-28 May), European Space Agency. 321-326.

Boumans, R. and C. Heemskerk. 1998. "The European Robotic Arm for the International Space Station." *Robotics and Autonomous Systems* 23: 17-27.

Brandt, A.P., W. Elhorst and A.J. Lander. 1994. "EuroSim - European Real-Time Operations Simulator." In *Proceedings of the Conference on Modelling and Simulation 1994* (Barcelona, Spain, June 1-3). SCS, 566-570.

Couwenberg, M.J.H., R.M. Lieuw and D.J. Schulten. 1999. "The ERA Simulation Facility for the European Robotic Arm Programme".

Ellenbroek, M.H.M. 1994. "On the fast simulation of the multibody dynamics of flexible space structures - The recursive thoughts." Ph-D Thesis. Technical University Twente.

Pronk, Z, Schoonmade, M and Baig, W. 1999. "Mission Preparation and Training Facility for the European Robotic Arm (ERA)"

Technical description, 1999. "EuroSim Features - supporting the use of simulation throughout a project's life-cycle -"

EDITORIAL CALENDAR

1Q00 NETWORKS
(incl. RTOS Section)

2Q00 INSTRUMENTATION & AUTOMATION

3Q00 BUS TECHNOLOGY

4Q00 EMBEDDED INTERNET

previews on the web at:
<http://www.dedicated-systems.com>

ADVERTISEMENT INDEX

ONX SOFTWARE SYSTEMS	2
PHARLAP	13
NATIONAL INSTRUMENTS	17
CCII	22
DEDICATED SYSTEMS ENCYCLOPAEDIA	69
ANNABOOKS	72-73
ACT EUROPE	51

For more information on advertisement opportunities, call 32-2-520.55.77 or send an e-mail to info@dedicated-systems.com

EDITOR'S NOTE

Inclusion in the Company Directory of Dedicated Systems Magazine should not be taken as a Dedicated Systems Magazine endorsement or recommendation. Likewise, omission from the company directory should not be taken negatively. The information here was believed to be accurate at the time of writing, but Dedicated Systems Magazine cannot be responsible for omissions, errors or changes that occur after compilation.