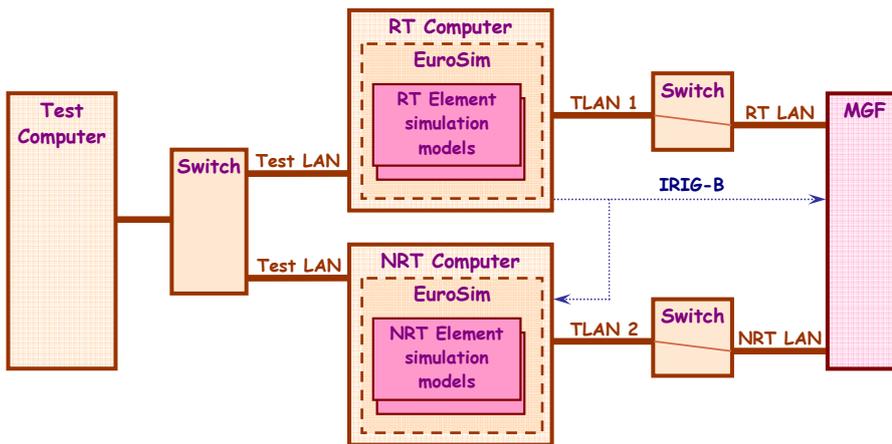




## Executive summary

# AIV Platform for the Galileo Message Generation Facility



### Problem area

The Message Generation Facility (MGF) is an element of the Galileo Mission Segment (GMS) and is responsible for distribution of the navigation, integrity and SAR messages from the processing facilities (OSPF, IPF, ERIS, RLSP) to the Up-Link Stations (ULS).

The main objective is to route a message to the correct ULS in time for on-board update of navigation data and integrity data for dissemination to users. The MGF element is being developed by Deimos Space S.L. (Spain).

### Description of work

To perform the Assembly, Integration and Verification (AIV) activities of the MGF, a dedicated test platform, named MGF-AIVP, is developed by NLR.

The MGF-AIVP simulates other elements in the GMS that are connected to the MGF, in real-time. Its focus is to verify the main objective of the MGF.

### Applicability

Galileo is a new GNSS developed by Europe to expand and enhance the current American GPS. It offers several navigation services: free-of-charge open service, safety-of-life service (with integrity guarantee), public regulated service (military), commercial service.

The MGF is a crucial part of the Galileo ground system. Hence, the AIVP plays an important role in the acceptance of the system, requiring knowledge of real-time simulations.

### Report no.

NLR-TP-2007-464

### Author(s)

B.A. Oving  
T. Zwartbol  
S. Denham  
M. Rennie

### Report classification

UNCLASSIFIED

### Date

May 2008

### Knowledge area(s)

Space

### Descriptor(s)

test system  
Galileo  
EuroSim  
simulation  
verification





NLR-TP-2007-464

## AIV Platform for the Galileo Message Generation Facility

B.A. Oving, T. Zwartbol, S. Denham<sup>1</sup> and M. Rennie<sup>1</sup>

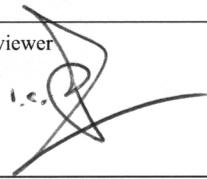
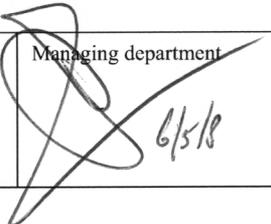
<sup>1</sup> Deimos Space S.L.

This report is based on a presentation held at the DASIA 2007 conference, Naples, 31 May 2007.

The contents of this report may be cited on condition that full credit is given to NLR and the authors.

Customer                      National Aerospace Laboratory NLR  
Contract number            ----  
Owner                         National Aerospace Laboratory NLR  
Division                      Aerospace Systems & Applications  
Distribution                 Unlimited  
Classification of title      Unclassified  
                                      May 2008

Approved by:

Author 	Reviewer 	Managing department  6/5/08
---	---	--



## Summary

The Message Generation Facility (MGF) is an element of the Galileo Mission Segment (GMS) and is responsible for real-time distribution of the navigation, integrity and SAR messages from the processing facilities (OSPF, IPF, ERIS, RLSP) to the Up-Link Stations (ULS).

The main objective is to route a message to the correct ULS in time for on-board update of navigation data and integrity data for dissemination to users. The MGF element is being developed by Deimos Space S.L. (Spain).

To perform the Assembly, Integration and Verification (AIV) activities of the MGF, a dedicated test platform, MGF-AIVP, is developed by the National Aerospace Laboratory, NLR (the Netherlands). The MGF-AIVP simulates other elements in the GMS that are connected to the MGF, in real-time. Its focus is to verify the main objective of the MGF.

## Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>Test system description</b>	<b>7</b>
<b>3</b>	<b>Test system architecture</b>	<b>8</b>
<b>4</b>	<b>Test system interfaces</b>	<b>9</b>
<b>5</b>	<b>Simulation framework</b>	<b>10</b>
<b>6</b>	<b>Simulation models</b>	<b>12</b>
<b>7</b>	<b>Simulation behaviour</b>	<b>14</b>
<b>8</b>	<b>Acknowledgements</b>	<b>15</b>
<b>9</b>	<b>References</b>	<b>15</b>

## Abbreviations

AIVP	Assembly, Integration and Verification Platform
ASN.1	Abstract Syntax Notation (nr. one)
CRC	Cyclic Redundancy Check
CS	Commercial Service
ERIS	External Regions Integrity System
FTP	File Transfer Protocol
GACF	Ground Assets Control Facility
GCC	Galileo Control Centre
GCS	Galileo Control Segment
GMS	Galileo Mission Segment
GNSS	Global Navigation Satellite System
GSS	Galileo Sensor Station
GST	Galileo System Time
ICD	Interface Control Document
IP	Internet Protocol
IPF	Integrity Processing Facility
KMF	Key Management Facility
LAN	Local Area Network
MDDN	Mission Data Dissemination Network
MDL	Mission Definition Language
MGF	Message Generation Facility
MUCF	Mission & Uplink Control Facility
NRT	Near Real-Time (LAN)
OSPF	Orbital Synchronisation Processing Facility
PTF	Precise Timing Facility
RLSP	Return Link Service Provider
RT	Real Time (LAN)
SAR	Search And Rescue
SIS	Signal In Space
SMP	Simulation Model Portability (standard)
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
ULS	Up-link Station
UML	Unified Modelling Language

# 1 Introduction

The Galileo system provides several services to users around the globe. The satellites in the space segment transmit signals containing time, navigation, and integrity information with which users are able to determine their position.

A network of ground stations all over the world containing Galileo receivers continuously measures the satellites' positions<sup>1</sup>. These measurements are sent to the Galileo Control Centres in Europe, to be processed by the Orbital Synchronisation Processing Facility (OSPF) and the Integrity Processing Facility (IPF). The OSPF is responsible for updating the navigation data, such as satellite position parameters and clock parameters, whereas the IPF produces integrity status flags for each satellite.

Figure 1 shows this continuing process as a closed-loop system, where the ground segment is composed of Galileo Sensor Stations (GSS), Mission Data Dissemination Network (MDDN), Ground Control Centre (GCC), and Up-Link Stations (ULS).

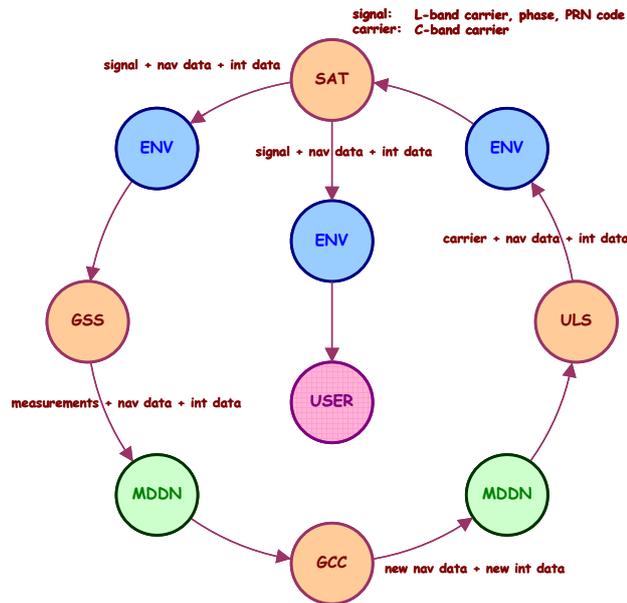


Fig. 1 Overview of Galileo navigation & integrity service

The navigation and integrity data is disseminated by the Message Generation Facility (MGF), residing inside the GCC, to the various UpLink Stations (ULS) that uplink the data to the satellites.

<sup>1</sup> Actually, the pseudo-range from satellite to receiver is measured as a time difference.

For redundancy reasons, there are multiple instances of each GMS Element in the GCC.

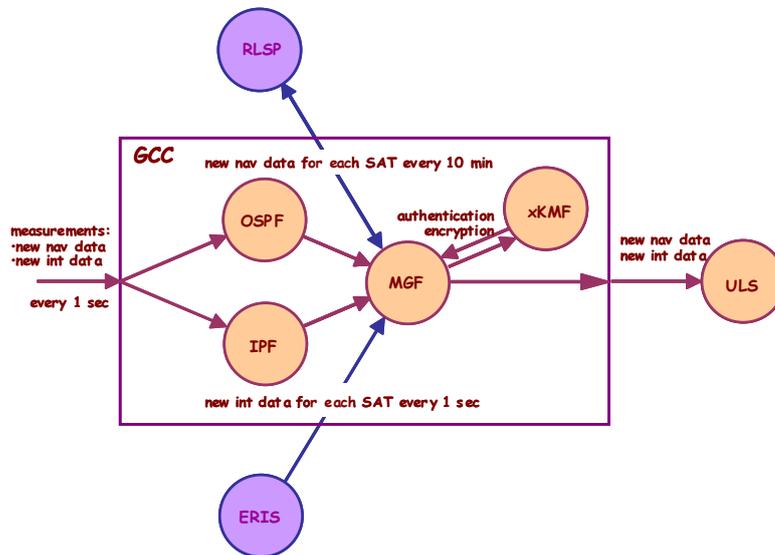


Fig. 2 Real-time data flow through MGF

Figure 2 shows a simplified picture of the real-time data flow, with interfaces from the processing facilities to the MGF and from the MGF to the ULS. For real-time authentication and encryption of data, the MGF has dedicated connections to the Key Management Facilities (xKMF).

As Galileo features an extra service called Search And Rescue (SAR), complementing the COSPAS/SARSAT system, the Return Link Service Provider (RLSP) provides data for the return link in answer to a distress signal from a user. Furthermore, integrity data coming from a (commercial) External Regional Integrity Service (ERIS) is inserted by the MGF into the integrity messages.

But the MGF has interfaces to other GMS Elements, as well:

- As the satellites orbit around the planet, a visibility schedule is needed to route a message for a particular satellite to the ULS that has that satellite in view. This visibility schedule is produced by the Mission Uplink and Control Facility (MUCF), which sends it periodically to the MGF.
- The MGF has a monitoring and command interface with the Ground Assets Control Facility (GACF).
- The MGF has a timing interface to the Precise Timing Facility (PTF) for synchronising its local time with the Galileo System Time (GST).
- In case the normal C-band uplink via ULS cannot be used, the navigation service switches to S-band uplink via the Galileo Control Segment (GCS).

- For Commercial Services (CS), the MGF interfaces with the Service Provision Facility (SPF).

As the GMS Elements are connected in dedicated Local Area Networks (LAN), real-time interfacing is based on the UDP/IP protocols over the RT-LAN. The Non-real-time LAN (NRT-LAN) is used for all other interfacing (e.g. file transfers). The MGF is built as an embedded system with a large number of active physical network ports and runs a real-time certified operating system.

## 2 Test system description

To verify MGF requirements, test scenarios will be defined in the MGF AIV plan. The Assembly, Integration & Verification Platform (AIVP) must be able to play these scenarios. The role of the AIVP is to act as the counterpart of the MGF, *simulating* the operational environment, in this case a part of the Galileo Mission Segment (GMS). To enable this, the AIVP contains *simulation models of all the elements*<sup>2</sup> that communicate with the MGF. In this way, the MGF thinks it is running in an operational environment.

As the MGF is communicating in real-time, the AIVP must be able to test the level of real-time guarantee, and hence must be real-time as well. Timing measurements will show the delay of the MGF processing and the interface delays in order to verify MGF performance requirements.

The simulation models are able to set up and maintain the connections to the MGF. Furthermore, depending on the required level of simulation, the models implement (part of) the behaviour of the corresponding element, following the protocols as defined in the GMS Interface Control Documents (ICD). This behaviour is essentially a mix of synchronous (cyclic) messaging by sending models and asynchronous reception of messages by receiving models, everything in real-time.

When a simulation/test run has finished, the results will be analysed to assess the pass/fail criteria of the test. Depending on the complexity of a test scenario, pass/fail criteria are checked automatically in a test script (and logged during the simulation run) or verified in a post-processing step using a dedicated tool.

Another important capability of any test system is the injection of errors for testing the robustness of the unit-under-test. The AIVP features a scripting language to implement test scenarios that command and control the simulation models.

---

<sup>2</sup> This means that nearly the complete Galileo Mission Segment (GMS) is modelled on interface level.

### 3 Test system architecture

The design of the MGF-AIVP is based on separation of the real-time and non-real-time domains, see figure 3. It has the following hardware components:

- Test Computer, responsible for test monitoring and control.
- Real-Time Computer, responsible for running the real-time simulation models.
- Non-Real-Time Computer, responsible for running the non-real-time simulation models.
- Switches, responsible for LAN conversion from the Ethernet ports of the MGF to the (limited number of) Ethernet ports of the AIVP.

The Test Computer is a standard PC, whereas the RT/NRT Computers are rack-mounted dual-processor servers that run the simulation models of the GMS Elements using the EuroSim real-time simulation framework.

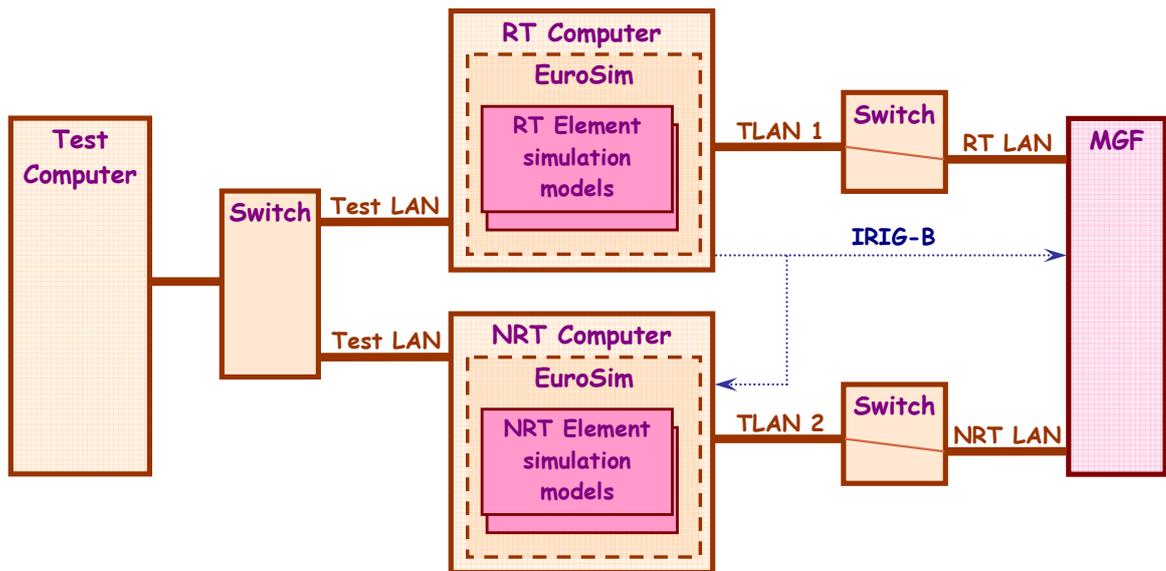


Fig. 3 MGF-AIVP architecture

The Real-Time Computer includes a “PTF simulator” that generates the master time reference (GST reference via IRIG-B 1pps) to which the simulators as well as the MGF are synchronised. The uniform time reference is achieved by slaving the EuroSim scheduler directly to the IRIG-B clock.

All simulator components of the AIVP will be slaved to this GST reference to obtain *unambiguous time-stamping* when recording values of variables, messages, and events. This



ensures time-related issues to be *traced* when combining the recordings of the AIVP machines and the MGF.

#### 4 Test system interfaces

Four interface types can be discerned for the MGF:

- a message interface for real-time data exchange, based on the GMS protocol over TCP/IP or UDP/IP over Ethernet;
- a message interface for status management, based on the Simple Network Management Protocol (SNMP) over Ethernet;
- a file transfer interface using the File Transfer Protocol (FTP) over TCP/IP over Ethernet;
- a time synchronisation interface using a *1-pulse-per-second (1PPS) IRIG-B* standard.

A two-layer approach separates the models into a Common Layer and a Specific Layer, where the Common Layer is responsible for socket handling (UDP/IP, TCP/IP, SNMP) and for the encoding and decoding of GMS messages. The Specific Layer deals with behavioural aspects, such as states, (automatic) responses, command handling, and keep-alive mechanisms.

<b>Specific Layer</b>	<i>Application- specific behaviour</i>		
<b>Common Layer</b>	GMS data messages	files	GMS m&c messages
	NRT	RT	FTP SNMP
<b>Network Layer</b>	TCP	UDP	
<b>Transport Layer</b>	IP		
<b>Data Link Layer</b>	Ethernet		
<b>Physical Layer</b>	Ethernet		

Fig. 4 Layered approach for GMS element models interfacing with MGF

All events, such as reception & transmission of messages, errors & warnings, script execution, are written to logs with a timestamp.

The implemented UDP interfaces support both uni-cast and multi-cast transmission of messages. Apart from the different addressing in the corresponding sockets, no distinction is made in the design between uni-cast and multi-cast.

When multiple instances of a model are available, they all send to the same port; no restrictions are imposed to prevent the interleaved arrival of messages at MGF. It is however possible to

configure the start slots of the cyclic models in such a way that each model has its own start slot. In this way, sequential arrival of messages can still be simulated.

## 5 Simulation framework

The MGF-AIVP relies for a large part on the simulation framework EuroSim. EuroSim is used to build, schedule and execute the simulation models.

Important EuroSim features are its real-time scheduler, with which tasks (containing entry-points to model functions) can be executed, and the data dictionary: a shared-memory pool containing model data.

On multi-processor machines, EuroSim is able to reserve a processor (provided there are more than one processor in a machine) for real-time execution. Only the necessary interrupts are handled by that processor, guaranteeing known performance at operating system and simulation level.

Furthermore, EuroSim allows models to be scheduled on either the real-time processor (with given boundary conditions for the maximum allowed processing time) or the non-real-time processor. This enables the user to easily change the domain of a simulation model, e.g. from non-real-time to real-time, by simply editing the EuroSim schedule.

In the EuroSim schedule, tasks can be connected to interrupts or to timers with a user-defined frequency. The latter option is to be used for repetitive actions, such as sending messages every second with a real-time guarantee.

EuroSim also features a real-time scripting language that allows the user to precisely and repetitively execute actions to influence the simulation. Via EuroSim External Simulator Access (ExtSim), these scripts can be accessed externally by another test scripting language, e.g. Perl. External Simulator Access provides access and control to a running simulation, as well.

Figure 5 shows the functioning of a EuroSim simulation with the Data Dictionary as the central part and the real-time scheduler controlling the access to the Data Dictionary. The models publish (selected) attributes as variables, and (selected) methods as entry-points in the Data Dictionary. The scheduler updates the time with every clock tick and calls the models according to the schedule, which then perform the activity to update the variables.

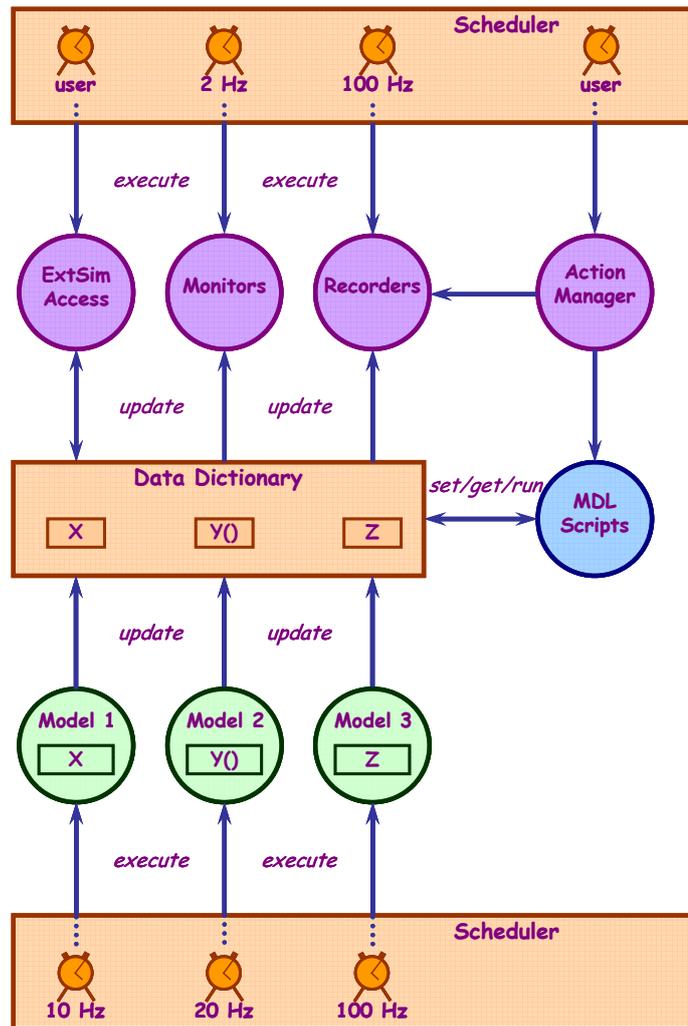


Fig. 5 Overview of EuroSim functioning

The real-time scripting engine of EuroSim is based on the Mission Definition Language (MDL), which has a C-like syntax with the obvious built-in statements (if, while) and functions. An MDL script can also execute entry-points, i.e. functions of a model that are put in the Data Dictionary. For instance, the *send* method of a model is put in the Data Dictionary as an entry-point to allow the user to create a test script that calls the method at a specific Simulated Time.

The monitors are used to display the values of selected simulation variables with a frequency of 2 Hz. A recorder logs selected values with a user-definable frequency. Just like MDL scripts, they can be created and executed during a simulation run.

The external simulator access allows other software applications (such as a scripting language like Perl) to connect to the (initialised or running) simulation.

## 6 Simulation models

The simulation models are designed using the Unified Modelling Language (UML) to allow inheritance and multiple instantiation. As the implementation is done mostly in C++, the models need to comply with the Simulation Model Portability (SMP) standard, so that they can be imported in the EuroSim framework.

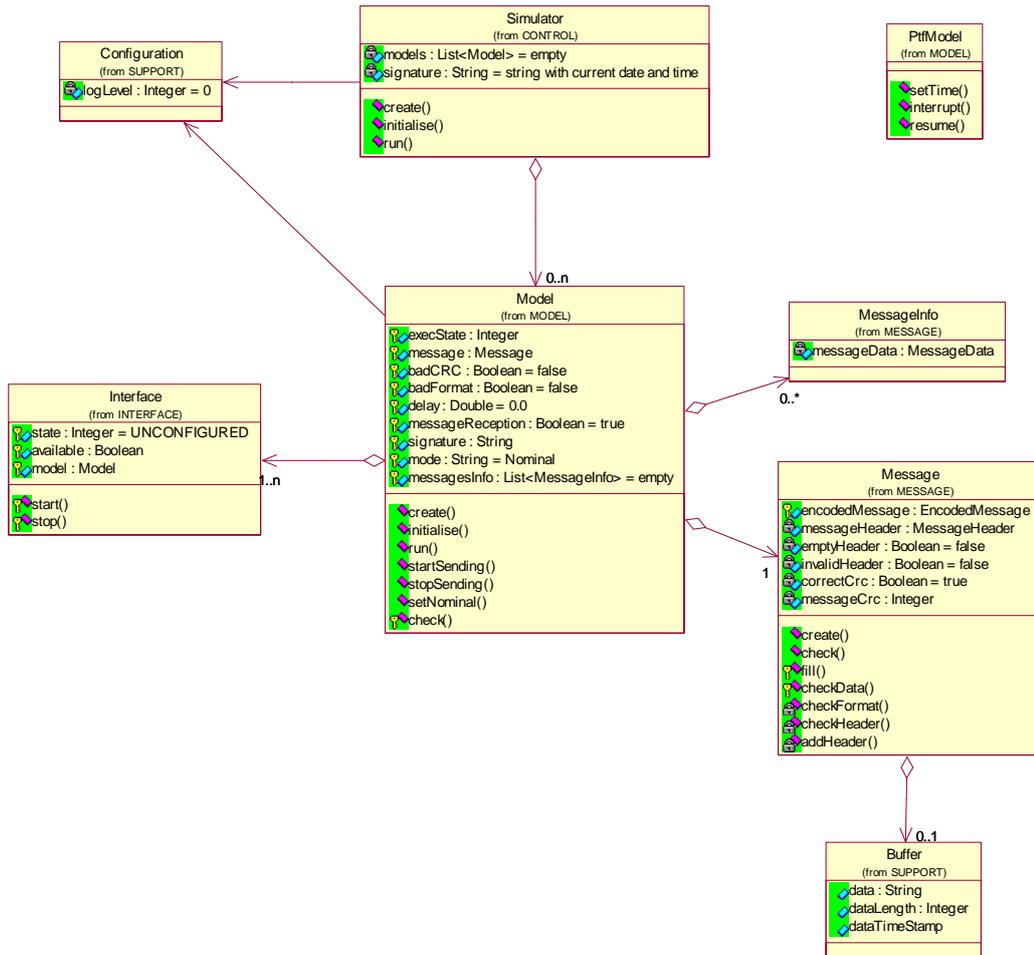


Fig. 6 Top-level class diagram for the simulator

Each server runs the exact same simulator, but through initial configuration, only those models that need to be connected will be active. By setting the state of the RT models to ON and those of the NRT models to OFF, the AIVP can be configured for the RT Computer. The same approach holds for the NRT case.

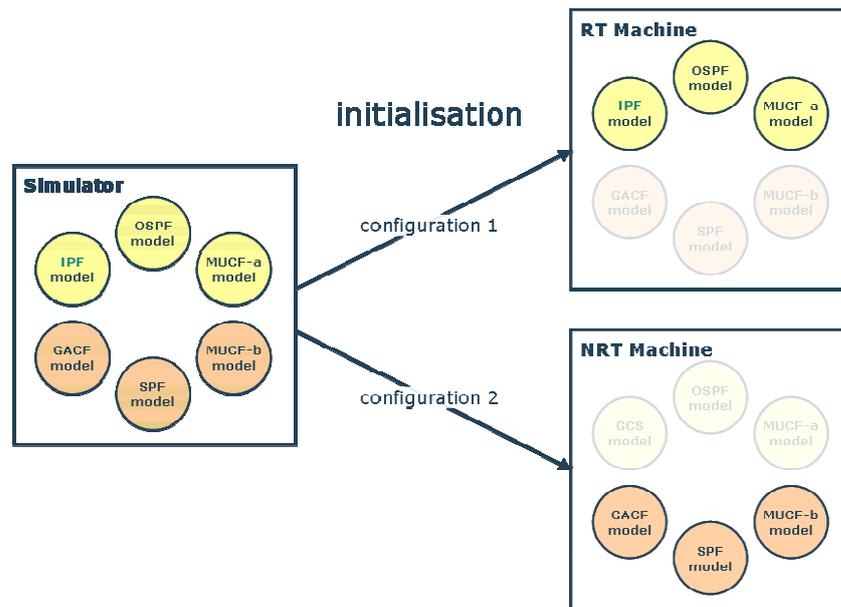


Fig. 7 Model deployment

The example shows that the simulator configuration 1 on the RT Machine only activates the RT models, whereas configuration 2 activates the NRT models on the NRT Machine.

Note that several instances of each model are created during initialisation of the simulator, e.g. 10 instances of the ULS model to represent the complete ULS network.

The ‘sending’ models can be configured for failure injection by sending corrupted messages or not sending messages at all:

- insert a bad or correct CRC in the message;
- use an incorrect format for the message;
- delay the sending of the messages;
- switch sending of messages on or off.

These substates are independent of each other. Furthermore, the scheduling of each model task can be changed during run-time, so that the models can send their messages faster or slower than they should.

The ‘receiving’ models will verify the correct behaviour of the MGF by checking the routing, the timing and the contents of the messages. Checks of the message contents are performed at the models where the data originated (e.g. IPF or OSPF), whereas checks of the routing are performed by the MUCF model.

The real-time messages are defined in Abstract Syntax Notation (ASN.1) files. The MARBEN ASN.1 toolkit is used to generate C++ source code from these definitions. Dedicated wrapper classes are defined to ‘insert’ the generated code into the EuroSim project.

## 7 Simulation behaviour

The communication protocols between MGF and GMS Element are described in the MGF ICD. The simulation models implement that behaviour. Figure 8 shows the sequence diagram for the handling of navigation data between the MGF-AIVP simulation models and the MGF.

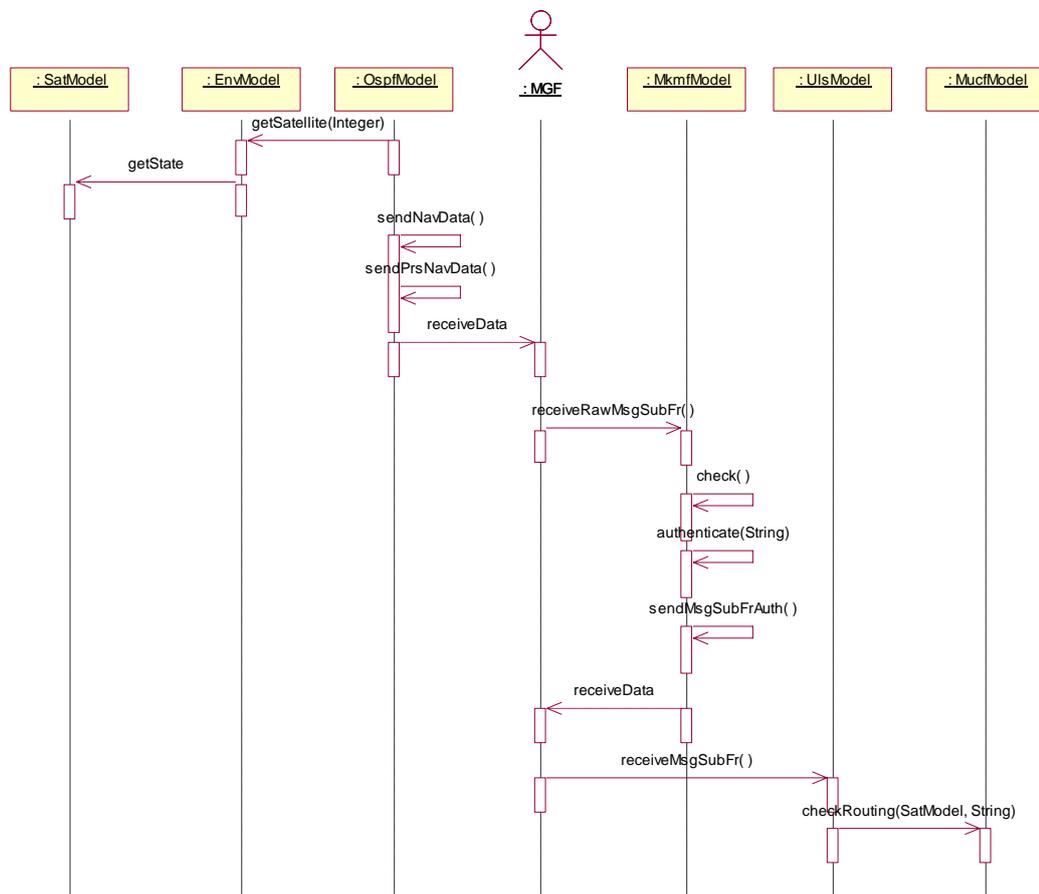


Fig. 8 Navigation data processing (Open Service)

The OSPF model first retrieves the state of the satellites, which can also be ON or OFF. Then, the navigation data for the satellites is sent to the MGF, according to the ICD. The MGF creates a Galileo message from the navigation data and asks the MKMF to authenticate the message.

Then, the MGF sends the authenticated message to the ULS model, which in turn calls the MUCF model to verify whether the MGF routed the messages to the correct ULS.

## **8 Acknowledgements**

NLR thankfully acknowledges the fruitful co-operating with Deimos Space and their contributions to this paper.

## **References**

1. EuroSim, [www.eurosim.nl](http://www.eurosim.nl)