EAGLE: An Extensible, End-to-End Simulation and Evaluation Framework for Planetary EDLS

Guy Johns[(1)], Ender St. John-Olcayto[(2)], Alastair Pidgeon[(2)], Christian Philippe[(3)]

(1) SciSys U.K. Ltd., 23 Clothier Road, Bristol, Avon BS4 5SS, United Kingdom
Tel.: +44 (0) 1179 165 192
Guy.Johns@scisys.co.uk

(2) SciSys U.K. Ltd, Methuen Park, Chippenham, Wiltshire SN14 0GB, United Kingdom
Tel.: +44 (0) 1249 466 467, Tel.: +44 (0) 1249 466 310,
Ender.Olcayto@scisys.co.uk, Alastair.Pidgeon@scisys.co.uk

(3) ESTEC, Keplerlaan 1, Postbus 299, 2200 AG Noordwijk, The Netherlands
christian.philippe@esa.int

1. Keywords

Entry, Descent, and Landing; Entry, Descent, and Landing Systems; End-to-End Simulator; Systems Engineering; Simulation Fidelity.

2. Extended Abstract

Entry, descent, and landing (EDL) is one of the most critical parts of a surface planetary mission following the launch sequence. It is also a mission phase that is most reliant on spacecraft autonomy with little scope for human intervention. For this reason, EDL is subject to high levels of simulation-based evaluation at multiple design phases. Each of these phases use different levels of fidelity with the fidelity increasing as the mission concept and associated design matures.

The Entry and Guided Landing Environment (EAGLE) is a MATLAB / Simulink-based environment for systems engineers and domain specialists to design, develop, and evaluate EDL mission concepts and the systems and algorithms to support these concepts. It consists of the following elements (see Figure 1):

1. The simulation kernel,
2. Analysis tools,
3. Design tools,
4. User interface.

The initial design goal of EAGLE is to provide a platform for the design and evaluation of guidance, navigation, and control (GNC) algorithms for Mars EDL but it can be extended to other similar applications such as Earth, Venus, Titan, and Lunar EDL (such as Lunar NEXT) but also non-EDL applications such as aerobraking and asteroid rendezvous and landing.

EAGLE provides an iterative simulation workflow (see Figure 2) that starts with the simulation needs of phase 0 and phase A studies and provides data products for the

simulation needs of subsequent systems engineering phases, culminating in the validation of flight software.

To support the above workflow, EAGLE provides four macro-levels of simulation fidelity that are mapped to the following simulation classes:

1. System concepts simulator (SCS),
2. Mission performance simulator (MPS),
3. Non-real-time functional engineering simulator (NRT-FES),
4. Avionics test bed (ATB).

SCS: The SCS is built in a desktop computing environment. The purpose of the SCS is to determine the feasibility of the selected mission from the vehicle's final trajectory correction manoeuvre to the lander touch-down point. A key feature of the SCS is that it operates much more quickly than wall-clock time. This is achieved using relatively simple spacecraft, environment, and algorithmic models (for example, point mass / single body representations of the spacecraft and a simple atmospheric model.) This simulation speed permits mission concepts to be evaluated, rejected, or modified quickly.

MPS: Like the SCS, the MPS is built on the desktop. The MCS is a refinement of the SCS and is used to elaborate the spacecraft and environmental models. Aspects of the MCS include full 6 DOF modelling of all the spacecraft bodies, a realistic atmospheric model, and fault injection capability. After the necessary refinements and elaborations have been performed, the MCS provides a platform for the definition of a GNC architecture. The MCS permits parameter trade-offs and robustness tests (through Monte Carlo methods) of the GNC concept to be performed.

NRT-FES: The NRT-FES represents another increment in simulation fidelity and is derived from the MCS. It is used to focus on specific mission phases and events (such as parachute / airbag / landing leg deployments) and also to confirm GNC robustness in a full fidelity simulation. The NRT-FES may be bypassed, with the MPS being converted to the ATB directly.

ATB: The ATB is used to evaluate the GNC algorithms in a real-time environment that represents the spacecraft computing resources as closely as possible (LEON 3 processors.) The ATB is especially useful in determining if the GNC algorithms satisfy timing, memory, and fault recovery constraints. The fidelity level of the spacecraft (dynamics, sensors, actuators, etc.) and the environment is equivalent to the MPS and is converted to real-time code and compiled and linked to run on the dSpace platform. This processor-in-the-loop architecture as shown in Figure 3 provides end-to-end simulation capability.

This paper will show representative use-cases of EAGLE in order to demonstrate the EDLS GNC design workflow and how the different levels of simulation fidelity aid the design. It will also show the component libraries that the designer can choose from and how they can be extended by conforming to EAGLE interfaces. The paper will further show how EAGLE is extensible to other scenarios including multiple planetary missions and non-planetary missions.

ACKNOWLEDGEMENTS

SciSys U.K. Ltd. is grateful for the support of ESTEC contract 21286/07/NL/EK under which EAGLE is being developed. SciSys also acknowledge the help of EADS Astrium, Vega (now SELEX Systems Integration), and Vortcity Ltd.

FIGURES

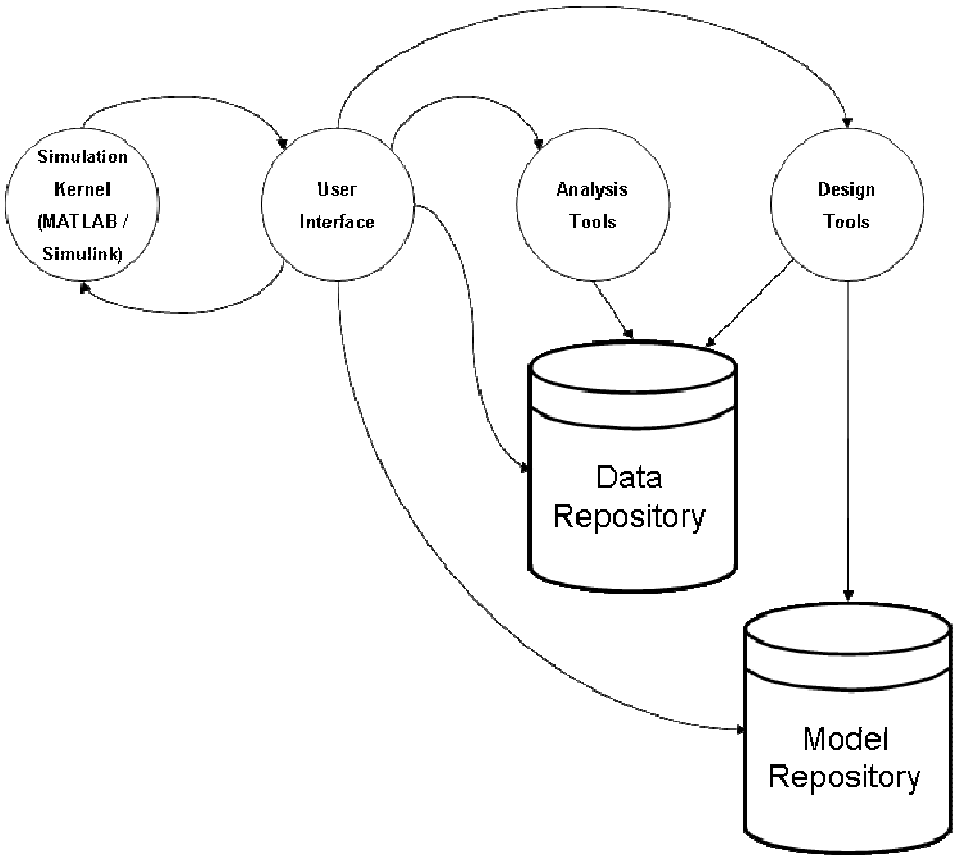

Figure 1: The EAGLE framework

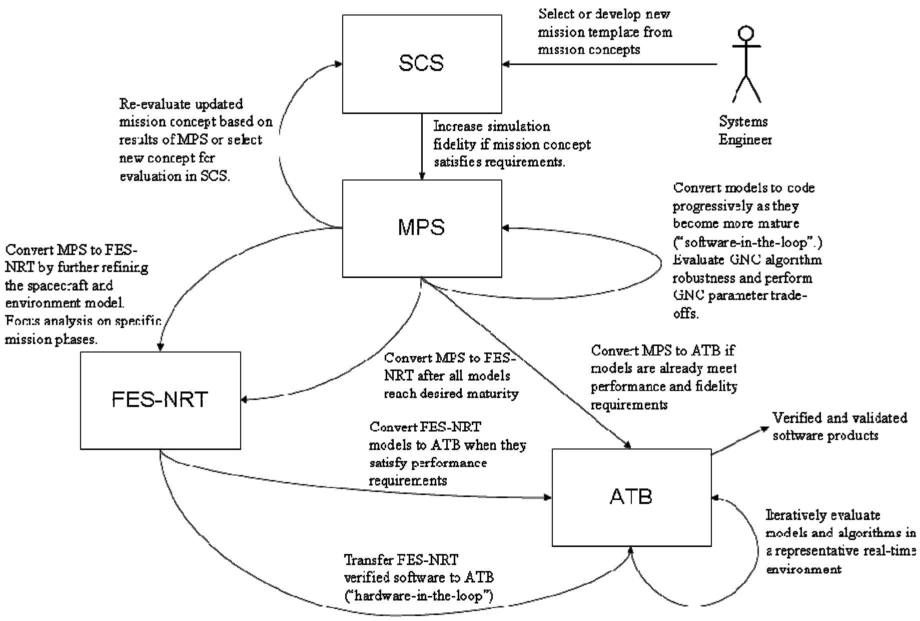

Figure 2: The EAGLE simulation development workflow

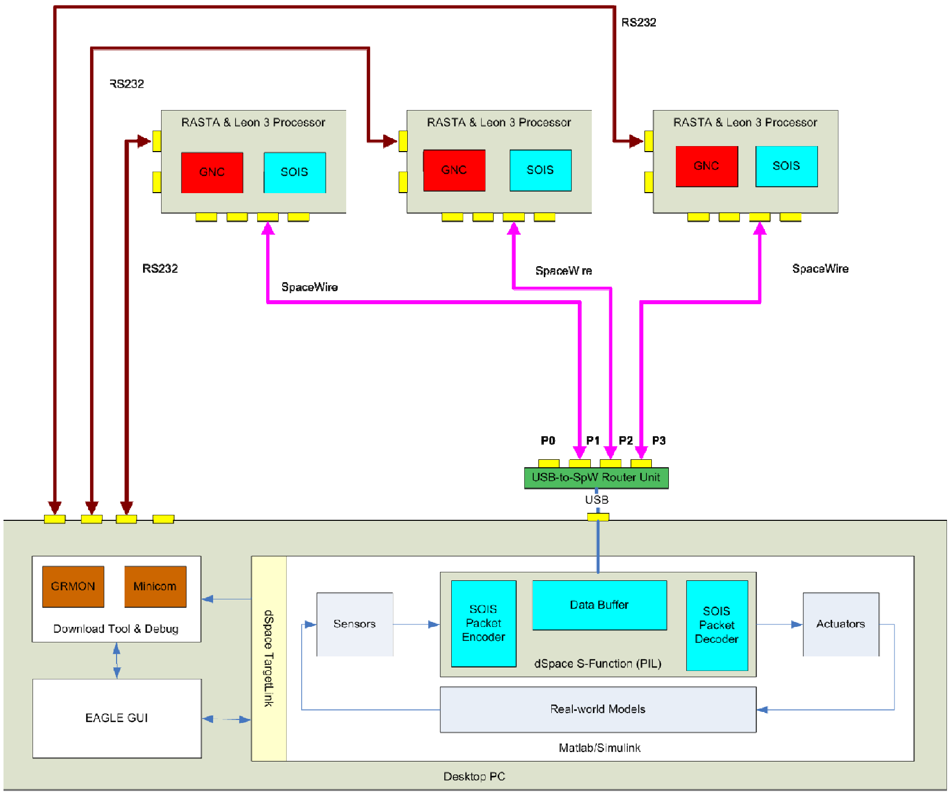

Figure 3: The EAGLE ATB architecture