Space dynamics software ELECTRA

Philippe Pavero, 15/03/2016
Table of Contents

- Introduction
- ELECTRA Functionalities
- Comparison of the Fortran/Java architecture
- Validation of ELECTRA Java
- Conclusion
Introduction

- 2004 : French Space Operations Act
- 2007 : Start of ELECTRA development
- 2010 : Deployment at the Guyana Space Center
- 2014 : Start of the Java version
Introduction

- ELECTRA Fortran
  - Linux SUSE 10sp4

- ELECTRA Java
  - Linux SUSE 10sp4, Redhat 6.4, Windows 7

- ORESTE : impact viewer
Table of Contents

- Introduction
  - ELECTRA Functionalities
- Comparison of the Fortran/Java architecture
- Validation of ELECTRA Java
- Conclusion
ELECTRA functionalities

- Principle
  - Fragment impact locations => different modes available
  - Impact probability of occurrence
  - Population distribution
  - Habitat protection

- Many dispersions are available:
  - Environment
  - Initial vehicle position
  - Fragment characteristics
  - Fragmentation altitude
  - Maneuver characteristics

=> Monte-Carlo method
ELECTRA functionalities

- Uncontrolled re-entry: analytic computation
  - Orbit inclination
  - Fragment lists
  - => Risk by latitude band

- Every other mode computes trajectories using:
  - Numerical propagator
  - Force models:
    - Earth potential model
    - Atmospheric forces
    - Sun and Moon attraction
    - Solar radiation pressure
ELECTRA functionalities

- Controlled re-entry
  - Failure during re-entry maneuvers

- Launching
  - Failure during rocket launching
ELECTRA functionalities

- **Final orbit**
  - Exact re-entry point and ballistic coefficient are unknown
  - No maneuver
  - List of re-entry points called final orbit

- Anchor points and angular portion

- Shifting the anchor’s impacts
Table of Contents

- Introduction
- ELECTRA Functionalities
  - Comparison of the Fortran/Java architecture
- Validation of ELECTRA Java
- Conclusion
Comparison of the Fortran/Java architecture

- Language benefits
  - Portability
    - Linux SUSE 10sp4
    - Redhat 6.4
    - Windows 7
  - Unified workshop made by CNES
    - Eclipse
    - Maven
    - Quality standards and tools: Checkstyle, PMD, Findbugs
  - CIP
    - Unit testing
    - Crucial to a team of 5 people
    - Sonar
Comparison of the Fortran/Java architecture

- **Simplification**
  - CNES standard libraries evolution: mainly BIBMS => Sirius
    - MSLIB, MECASPA, PSIMU, COMPAS => Patrius
    - MAGE => Java
    - GENESIS and MADONA => GENIUS

- **Maven**
  - Chain of dependencies
  - Generation and installation
Comparison of the Fortran/Java architecture

- **Specific improvements**
  - Dispersions handling
    - Fortran: drawn at usage
    - Java: centralized drawing

- **Parallelization**
  - Fortran: Open-MPI
  - Java: Executor Framework

![Diagram](image-url)
Table of Contents

- Introduction
- ELECTRA Functionalities
- Comparison of the Fortran/Java architecture
- Validation of ELECTRA Java
- Conclusion
Validation of ELECTRA Java

- Testing specific functions
  - Example: uniform distribution
    - 10,000,000 drawings, 50 boxes
    - Expectation: 200,000 drawings per box

![Number of drawings per box deviation](image)
Validation of ELECTRA Java

- **Unavailability of models**
  - Most BIBMS models absent from Patrius
  - Model interfaces
  - If each model is validated, validation of ELECTRA only requires one model

- **Example**
Validation of ELECTRA Java

- Changes in the dispersion of variables
Validation of ELECTRA Java

- Changes in the dispersion of variables

![Graph showing risk evolution wrt the simulation number](image-url)
Conclusion

- New CNES referential: Patrius, Genius

- Java, Eclipse, and Maven => Easier to develop, maintain, generate, test.

- **BUT**: need careful conception phase

- Delicate points in validation

- **Future**: more complex modes, other tools