



Deliverable D4.1

Inventory of Practice: Integration

Deliverable

<i>Number and title:</i>	D4.1 – Inventory of Practice: Integration		
<i>Work package:</i>	WP4 (Integration and Validation)		
<i>Lead author:</i>	Holger Hermanns (USAAR)		
<i>Contributors:</i>	Gregory Stock (USAAR), Juan A. Fraire (UNC)		
<i>Reviewers:</i>	Roberto Cibils (INVAP), Zhana Imbrosh (GOM), Felix Walter (D3TN)		
<i>Due date (GA):</i>	M6 (2022-03-31)	<i>Dissemination level:</i>	Public
<i>Due date (revised):</i>	2022-08-31	<i>Version:</i>	1.0 (final)
<i>Submission date:</i>	2022-08-24	<i>Pages:</i>	8

Version history

<i>Version</i>	<i>Date</i>	<i>Notes</i>
1.0	2022-08-24	First official release

Project

<i>Title:</i>	Models in Space Systems: Integration, Operation, and Networking		
<i>Acronym:</i>	MISSION	<i>Start date:</i>	01-10-2021
<i>GA no.:</i>	101008233	<i>Duration:</i>	48 months
<i>Call:</i>	H2020-MSCA-RISE-2020	<i>Website:</i>	mission-project.eu



This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 101008233. This document reflects only the authors' view. It does not represent the opinion of the European Union, and the European Union is not responsible for any use that may be made of the information it contains.

1 Introduction

This deliverable presents a summary of techniques/tools used at our industrial partners, together with methods mandated or suggested by authorities, as well as requirements or wishes of the industry. The information presented here was mainly collected by means of an electronic questionnaire distributed to all industry partners of MISSION (Ascentio, INVAP, D3TN, and GomSpace).

1.1 Electronic Questionnaire

The questionnaire was comprised of the following three questions.

Question Q1

Give an overview of the *techniques* and *tools* that you (as an industrial partner) are using (or of which you are aware that other industrial partners are using it).

Question Q2

Describe *methods* that are mandated or suggested by public authorities.

Question Q3

Give an overview of the *requirements* that exist and the *wishes* that the industry has, in the context of modelling techniques, tool support, and technology platforms.

Q1 Techniques and Tools in Use at Industrial Partners.

- **Planning:** This service allows the planification of activities to be carried out in flight and on ground to comply with the scenarios defined by the mission. It allows the generation of activities to be planned, the programming of station services, the optimization of the plan to be executed, and the feasibility calculation (memory data, power and thermal balances, orbit dynamics). We use the following software tools: PyEphem [36], SimPy [42], NumPy [31], Cesium [10], Orekit [33].
- **RAMS:** We use Reliability Block Diagrams (RBD) model [44] to represent the logic relationships between component and system failures. RBDs are modelled using failure distribution data as input in PTC Windchill Quality Solutions 10.1 software [51] and then simulated to obtain the availability value.
- In addition, we use the following software tools: GitLab [18], Docker [13], Grafana [21], InfluxDB [22], Django [12], Kubernetes [25], Robot Framework [39].

-
- Reliability Modeling of Systems and Subsystems (Ansys Sherlock [2], Windchill Risk and Reliability (formerly Relx) [52])
 - Computer Aided Mechanical Design Modeling (SOLIDWORKS [45])
 - Computer Aided Electronic Design and Production Modeling (Altium [1])
 - Programmable Logic Design Modeling (Vivado [49])
 - Radiation Environment Shielding (FASTRAD [15])
 - Spreadsheets
 - MathCAD [35]
 - GNU Octave [20]
 - Bayesian Doctor [3]
 - FMEA [6]
 - FMECA [6]
 - Fault Trees [40]
 - Reliability Block Diagrams [44]

-
- **Discrete-event-based simulation:** We develop and use tools to simulate and emulate data transmission through delay-tolerant networks using different routing techniques and implementations of the DTN protocols (tools: DtnSim [17], aiodtnsim [50], STK [46])
 - The DTN architecture (RFC 4838 [47]) and protocol stack (RFCs 5050 [41], 9171–9174 [7, 4, 5, 43]) and its implementations (tools: especially uD3TN [16], ION [8])
 - Development techniques, tools, languages: GitLab [18], Continuous Integration (GitLab CI [19]), Static Code Analysis, Fuzzing [26], C [23], Rust [24], Python [48]
 - Other tools and protocols: XMPP [54], OpenPGP [9], Matrix [27] / Element [14], Docker [13], Nix [30], OpenStack [32], Kubernetes [25], Prometheus [34], WireGuard [53], Nextcloud [29]

Q2 Methods Mandated or Suggested by Public Authorities.

- FMEA: FMEA (Failure Mode and Effects Analysis) [6] is the process of reviewing as many components, assemblies, and subsystems as possible to identify potential failure modes in a system and their causes and effects. In the aerospace industry, this is usually applied as a method for a reliable hardware design. Sometimes FMEA is extended to FMECA (Failure Mode, Effects, and Criticality Analysis) [6] to indicate that criticality analysis is performed too.
- RAMS (Reliability, Availability, Maintainability and Safety) analysis [37] is used to characterize a product or system.
- FDIR: Fault detection, isolation, and recovery (FDIR) is a subfield of control engineering which concerns itself with monitoring a system, identifying when a fault has occurred, and pinpointing the type of fault and its location. We define the critical operations considering the mission FDIRs.
- Moreover, we use other international standards of the space industry, such as MIL and NASA Standards & Handbooks [28], CCSDS recommendations [11], and ECSS Standards.

-
- FMEA [6]
 - Fault Trees [40]

We do not yet engage in projects with strict requirements from public authorities but are using various techniques in order to reduce the chance of issues/bugs/vulnerabilities within our software. Besides the tools and techniques mentioned above (such as the development of most new components in Rust [24] and leveraging CI, static analysis, and fuzzing [26]), we aim to minimize the codebases and the number of transitive dependencies of all software we develop.

Q3 Requirements or Wishes of the Industry in the Context of Modelling Techniques, Tool Support, and Technology Platforms.

- Planning Optimization: Currently, we have a planning service running for a simple satellite mission, but we are aiming to upgrade the service for a satellite constellation. Also, we need to improve our planning system for mission resources optimization (memory, power, others).
- Power Modelling: We aim to upgrade and scale the power modelling from single-satellite systems to satellite constellations.
- DevOps: Historically, our software was designed and implemented to be run on the clients' premises, and now we are starting with the DevOps philosophy by upgrading our software to apply for the micro-services design and cloud deploys. Some techniques used are: Docker [13], Ansible [38], Kubernetes [25]. *WISH*
- We are developing a new product: GSaaS (Ground Segment as a Service). GSaaS is a group of services aimed to solve the needs of space missions that require a simple, flexible, scalable and robust Ground Segment. It is grounded in the experience obtained by missions where one of the partners of MISSION was involved: SAC-C, SAC-D, SAOCOM, SABIAMAR. The partner highlights that this product is in an early stage and could profit from collaboration.

The partner highlights that they are not only involved in space projects but also nuclear. In this last subject, there is a special interest in advanced techniques and tools that permit more efficient management of nuclear reactors and facilities maintenance and operation planning.

It would be perfect to have easy-to-use tooling to further enhance the correctness and reliability of real production codebases, e. g., to automatically and quickly check (subsets of) Rust [24] programs within a Continuous Integration pipeline. It is important that the introduction of such tools and processes does not require rebuilding huge parts of the existing software or having years of experience in the modelling field as a developer working on the code. The possibility of incremental deployment (e. g., one function at a time) would be a huge advantage. An overall introduction of modelling approaches and tools for software engineers not specialized in theoretical computer science would be extremely helpful as a next step towards joining forces within the MISSION project.

References

- [1] *Altium Designer*. URL: <https://www.altium.com/altium-designer/>.
- [2] *Ansys Sherlock*. URL: <https://www.ansys.com/products/structures/ansys-sherlock>.
- [3] *Bayesian Doctor*. URL: <https://www.spicelogic.com/Products/Bayesian-network-software-20>.
- [4] Edward J. Birrane and Kenneth McKeever. *Bundle Protocol Security (BPsec)*. RFC 9172. Jan. 2022. DOI: [10.17487/RFC9172](https://doi.org/10.17487/RFC9172).
- [5] Edward J. Birrane, Alex White, and Sarah Heiner. *Default Security Contexts for Bundle Protocol Security (BPsec)*. RFC 9173. Jan. 2022. DOI: [10.17487/RFC9173](https://doi.org/10.17487/RFC9173).
- [6] Abdelkader Bouti and Daoud Ait Kadi. “A state-of-the-art review of FMEA/FMECA”. In: *International Journal of Reliability, Quality and Safety Engineering* 01.04 (1994), pp. 515–543. DOI: [10.1142/S0218539394000362](https://doi.org/10.1142/S0218539394000362).
- [7] Scott Burleigh, Kevin Fall, and Edward J. Birrane. *Bundle Protocol Version 7*. RFC 9171. Jan. 2022. DOI: [10.17487/RFC9171](https://doi.org/10.17487/RFC9171).
- [8] Scott C. Burleigh et al. “Delay-tolerant networking: an approach to interplanetary Internet”. In: *IEEE Commun. Mag.* 41.6 (2003), pp. 128–136. DOI: [10.1109/MCOM.2003.1204759](https://doi.org/10.1109/MCOM.2003.1204759).
- [9] J. Callas et al. *OpenPGP Message Format*. RFC 4880. Nov. 2007. DOI: [10.17487/RFC4880](https://doi.org/10.17487/RFC4880).
- [10] *Cesium*. URL: <https://cesium.com/>.
- [11] Consultative Committee for Space Data Systems (CCSDS). *Schedule-Aware Bundle Routing (SABR) (Blue Book, Recommended Standard CCSDS 734.3-B-1)*. July 2019. URL: <https://public.ccsds.org/Pubs/734x3b1.pdf>.
- [12] *Django*. URL: <https://www.djangoproject.com/>.
- [13] *Docker*. URL: <https://www.docker.com/>.
- [14] *Element*. URL: <https://element.io/>.
- [15] *FASTRAD*. URL: <https://www.fastrad.net/>.
- [16] Marius Feldmann et al. “Ring Road Networks: Access for Anyone”. In: *IEEE Commun. Mag.* 60.4 (2022), pp. 38–44. DOI: [10.1109/MCOM.001.2100835](https://doi.org/10.1109/MCOM.001.2100835).
- [17] Juan A. Fraire et al. “DtnSim: Bridging the Gap between Simulation and Implementation of Space-Terrestrial DTNs”. In: *2017 6th International Conference on Space Mission Challenges for Information Technology (SMC-IT)*. 2017, pp. 120–123. DOI: [10.1109/SMC-IT.2017.27](https://doi.org/10.1109/SMC-IT.2017.27).
- [18] *GitLab*. URL: <https://about.gitlab.com/>.
- [19] *GitLab Continuous Integration (CI)*. URL: <https://about.gitlab.com/stages-devops-lifecycle/continuous-integration/>.
- [20] *GNU Octave*. URL: <https://www.gnu.org/software/octave/>.
- [21] *Grafana*. URL: <https://grafana.com/>.
- [22] *InfluxDB*. URL: <https://www.influxdata.com/>.
- [23] *Information Technology – Programming Languages – C*. Standard ISO/IEC 9899:2018. Geneva, CH: International Organization for Standardization, June 2018. URL: <https://www.iso.org/standard/74528.html>.
- [24] Steve Klabnik and Carol Nichols. *The Rust Programming Language*. USA: No Starch Press, 2018. ISBN: 978-1-59327-828-1. URL: <https://nostarch.com/Rust2018>.
- [25] *Kubernetes*. URL: <https://kubernetes.io/>.

- [26] Jun Li, Bodong Zhao, and Chao Zhang. “Fuzzing: a survey”. In: *Cybersecur.* 1.1 (2018), p. 6. DOI: [10.1186/s42400-018-0002-y](https://doi.org/10.1186/s42400-018-0002-y).
- [27] *Matrix*. URL: <https://matrix.org/>.
- [28] *NASA Technical Standards System*. URL: <https://standards.nasa.gov/>.
- [29] *Nextcloud*. URL: <https://nextcloud.com/>.
- [30] *NixOS Linux*. URL: <https://nixos.org/>.
- [31] *NumPy*. URL: <https://numpy.org/>.
- [32] *OpenStack*. URL: <https://www.openstack.org/>.
- [33] *Orekit*. URL: <https://www.orekit.org/>.
- [34] *Prometheus*. URL: <https://prometheus.io/>.
- [35] *PTC Mathcad*. URL: <https://www.mathcad.com/>.
- [36] *PyEphem*. URL: <https://rhodesmill.org/pyephem/>.
- [37] *Railway Applications – The Specification and Demonstration of Reliability, Availability, Maintainability and Safety (RAMS) – Part 1: Generic RAMS Process*. Standard EN 50126-1:2017. Brussels, BE: European Committee for Electrotechnical Standardization (CENELEC), Oct. 2017.
- [38] *Red Hat Ansible*. URL: <https://www.ansible.com/>.
- [39] *Robot Framework*. URL: <https://robotframework.org/>.
- [40] Enno Ruijters and Mariëlle Stoelinga. “Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools”. In: *Comput. Sci. Rev.* 15 (2015), pp. 29–62. DOI: [10.1016/j.cosrev.2015.03.001](https://doi.org/10.1016/j.cosrev.2015.03.001).
- [41] Keith Scott and Scott C. Burleigh. *Bundle Protocol Specification*. RFC 5050. Nov. 2007. DOI: [10.17487/RFC5050](https://doi.org/10.17487/RFC5050).
- [42] *SimPy*. URL: <https://simpy.readthedocs.io/>.
- [43] Brian Sipos et al. *Delay-Tolerant Networking TCP Convergence-Layer Protocol Version 4*. RFC 9174. Jan. 2022. DOI: [10.17487/RFC9174](https://doi.org/10.17487/RFC9174).
- [44] Carlos Guedes Soares. “Reliability engineering and risk analysis: a practical guide: M. Modarres, M. Kaminskiy and V. Kritsov, Marcel Dekker Inc., New York, 1999, 542 pages”. In: *Reliab. Eng. Syst. Saf.* 77.2 (2002), pp. 207–208. DOI: [10.1016/S0951-8320\(02\)00008-X](https://doi.org/10.1016/S0951-8320(02)00008-X).
- [45] *SOLIDWORKS*. URL: <https://www.solidworks.com/>.
- [46] *Systems Tool Kit (STK)*. URL: <https://www.agi.com/products/stk>.
- [47] Leigh Torgerson et al. *Delay-Tolerant Networking Architecture*. RFC 4838. Apr. 2007. DOI: [10.17487/RFC4838](https://doi.org/10.17487/RFC4838).
- [48] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. ISBN: 978-1-4414-1269-0.
- [49] *Vivado*. URL: <https://www.xilinx.com/products/design-tools/vivado.html>.
- [50] Felix Walter. “Prediction-enhanced Routing in Disruption-tolerant Satellite Networks”. Doctoral dissertation. Technische Universität Dresden, 2020. URL: <https://nbn-resolving.org/urn:nbn:de:bsz:14-qucosa2-721622>.
- [51] *Windchill 10.0 – What’s new*. URL: <https://support.ptc.com/products/windchill/windchill-10-0/quality/whats-new.htm>.
- [52] *Windchill Risk and Reliability 12.0.1.0*. URL: <https://support.ptc.com/help/wrr/r12.0.1.0/en/>.

- [53] *WireGuard*. URL: <https://www.wireguard.com/>.
- [54] *XMPP*. URL: <https://xmpp.org/>.