Rail4Future

Project title:	**Resilient Digital Railway Systems to enhance performance**
Start date:	**01/04/2021**
Time duration:	**42 months**
Project number:	**882504**
Announcement:	**8. Ausschreibung COMET Projekte 2019**

Deliverable D1.1.4
Interface and Adapter Specifications

Due date	30.10.2022
Submission date	08.02.2023
Submitted by	TU Wien MIVP

Version	Date	Edited by	Description
0.1	31.01.2023	Ozan Kugu	Creation
1.0	31.0.1.2023	David Kern	Complements

Deliverable released

Dr. Manfred Grafinger

by Area Scientific Manager

Deliverable released

Dr. Filip Kitanoski

by Project Scientific Manager

1 Executive Summary

In this deliverable, interfaces and adapters are shown and described, which are implemented into the multi domain model, mentioned in the previous Deliverable 1.1.3 (M18), to improve and optimize the model and data integration process in the Rail4Future (R4F) platform for the digitalization of the holistic large-scale railway infrastructure system. First, the free standard FMI and its simulation unit FMU are defined, which is to be used to make different digital models more tool-independent, easy-to-use and platform-adaptive compared to their previous version, in this document. Additionally, the SSP standard is shortly mentioned, which is to be used to describe the sub-models with input parameters and outputs in common. Lastly, the essential adapters consisting of server- and UI-based software tools and the hardware components are shown and described, which are to be used to build the conceptual virtual and physical environment of the R4F platform. In addition, their parts in the process chart of the whole asset integration process are shown in some figures.

2 Table of Content

3 Abbreviations and Acronyms

Abbreviations / Acronyms	Description
R4F	Rail4Future
MBS	Multi body simulation
FMI	Functional Mock-up Interface
FMU	Functional Mock-up Unit
RLT	Residual Lifetime
UI	User Interface
VTI	Vehicle Track Interaction
AIT	Austrian Institute of Technology
ViV	Virtual Vehicle Research GmbH
CLI	Command Line Interface
VPN	Virtual Private Network
VM	Virtual Machine
SSP	System Structure and Parametrization
GUI	Graphical User Interface
OWL	Web Ontology Language
XML	Extensible Markup Language
DLL	Dynamic Link Library
JSON	JavaScript Object Notation
SSH	Secure Shell

4 Problem Description / Objectives

4.1 Problem Description

In order to translate the subcomponents of the holistic large-scale railway infrastructure system into the virtual world, sub-models such as MBS, VTI and RLT-Bridge were created and described before. The next step would be to adapt the sub-models into the R4F platform, where the integration and interoperation environment is to be equipped by using different software tools, hardware and interface standards adequately in consideration of user-friendliness, intellectual property, platform-adaptivity and software license management. As a result, the sub-models and data provided by different project partners are to be used with the adapters and interfaces in the R4F platform, which contributes to the digitalization of the railway infrastructure system significantly.

4.2 Objectives

In this deliverable, it is aimed to show the asset integration methodology by introducing and specifying the adapters and interfaces. Different digital assets are to be integrated into the R4F platform, where they are then to be interacted with each other in harmony. First, the FMI standard is proposed to increase easy-to-useness, tool-independency and platform-compatibility of the sub-models. For example, the residual lifetime calculation algorithm, written in a Python script and provided by the AIT, was packed into the FMU format, which was then executed in a conceptual integration environment. As another example, the MBS model, provided by the ViV, was packed into the FMU, which was also executed as well. Additionally, the SSP standard is shortly mentioned as well, which is used for the description and common parametrization of the sub-models. Second, the adapters such as further software tools and VMs are shown and specified, which assist the proper connection, semantical description and interaction of the assets with each other as integrated and interoperated in the R4F platform. Lastly, the proposed interfaces and adapters are addressed to both the conceptual integration environment and process chart of the designed integration process, which are then to be implemented into the R4F platform in future.

5 Significance for the overall Project

After virtualizing the subsystems of the holistic large-scale railway infrastructure system (rail vehicle, track, turnout, tunnel, sensors etc.) by applying the multi domain model as an IT system architecture, the methodology to integrate and interoperate all the assets in the R4F platform must be found out and extracted from knowledge, competence and expertise. Otherwise, it would be meaningless to put all the assets into the platform, because any visual output from the models wouldn't come out to the end user, which would prevent the user interaction with the platform. Besides, the sub-models and data need to be well-described semantically in order to reduce the data exchange complexity and therefore help different stakeholders to easily discover information about the assets of the railway infrastructure system in the platform. Therefore, the proposed interfaces and adapters are to be provided into the platform, which then enables the digitalization of the railway infrastructure system and bring benefit for the stakeholders as a result of the R4F project.

6 Description

6.1 FMI-Standard

The FMI is a free interface standard between software tools and dynamics simulation models, which also helps to exchange the models and is already supported by more than 170 tools [1]. Therefore, the simulation models get more tool-independent, easy-to-use and platform-compatible. In spite of that there are still challenges to be faced for complete adaption of all the sub-models into the R4F platform. For example, there are commercial software licenses to be connected, VPN for secure network connection and dependency on operating system (Windows & Linux) to be considered. Besides, some necessary software tool packages and additional plugins are to be provided to pack and execute the FMU of the sub-models, which is the simulation unit of the FMI consisting of an XML file (modeldescription.xml), binaries and C code [1]. For example, there are Python packages such as pythonfmu for FMU packaging from Python codes and fmpy to bring the FMU file into the simulation process, which are to be used for some use cases (e.g. RLT calculation of a steel bridge) in the R4F project. Furthermore, there are simulation software such as Simpack, provided from Dassault Systemés, where direct FMU-packaging of the MBS model in version FMI 2.0 for Co-Simulation is possible as discovered before. The model is to represent the rail vehicle drive on a designed track, which is surely essential to build the virtual illustration of the holistic large-scale railway infrastructure system. Of course, it is important to consider the valid license server (possibly needed VPN), existency of a Simpack solver and the kernel version of the VM due to the operating system for the FMU-simulation of the Simpack model. For Linux, Red Hat Enterprise Linux and SUSE Linux kernels are supported as mentioned in the Simpack Hardware and Platform Requirements section of the Simpack documentation. For Windows, both the Simpack and FMU simulation of the model would work in Windows 10 as shown in the documentation.

In future, it is also planned to co-accelerate and -deccelerate the MBS model by interacting it with another model, which is traction control system and will be modeled with MATLAB/Simulink. The FMU packaging of the control model can be realized from Simulink by using either FMI Kit for Simulink or Simulix for Co-Simulation export as an open-source example mentioned in the Tools section of [1].

6.2 SSP-Standard

The SSP [5] is a free interface standard to describe different sub-models with one or more FMUs belonging to different use cases of the railway infrastructure system. Besides, the SSP standard is used to parameterize all the models between many different software tools in common. To apply it to the platform, there are various potential software tools, where the available FMUs are to be imported and then interacted with each other with help of the SSP standard. Model.Connect [6] is one of the high potential tools for this purpose, which is provided by the AVL and commercial. The software is also used and proposed by the ViV to interoperate different dynamic models in a virtual environment.

6.3 Integration Environment with Adapters and Interfaces

First, Fig. 2 shows the conceptual design of the integration environment, where the typically used software tools are to be interconnected in hardware by using adapters and interfaces defined in thus subsection as well. The figure also shows the simple approach to be followed for the design of the integration process in the R4F platform. First, the main useful software tools such as Jenkins, Git and MySQL are specified. Jenkins is open-source, user-friendly and tool-independent software with useful plugins and used to provide the Jenkins pipeline technology, which is durable, pausable and enables the continuous integration of different digital sub-models and dataset into the platform as an automation engine. Besides, Git provides GitLab as version control system, which helps asset providers and integrators to work with different codes, input and output files belonging to the sub-models and dataset of different use cases collaboratively. All the files can also be uploaded into a Git repository and Jenkins pipeline can directly be connected to the repository of the Git server through the SCM system, which also provides the interface between the GitLab repository and Jenkins pipeline. In addition, MySQL is implemented in the integration environment as a relational DBMS, which can extract, archive and store raw data from the sub-models and their belonging input and output files, so that the whole dataset can be prepared for the visualization to the end user. The interface between MySQL and Jenkins is ensured by SQL query codes, which are executed by the pipeline to edit tables in a database located in the relational DBMS. Moreover, all the three mentioned software tools are containerized through the Docker engine by pulling Docker containers from Docker Hub [7], which is a library for container images. After that, the installation of all the containers can then be controlled and configured in a Docker-Compose file in YAML format. The Docker containerization is useful to isolate the tools, so that no collision between various software packages occurs and system resources of the tools are used efficiently.

All the three containerized software tools are implemented in a Linux VM with RedHat 64-bit kernel under consideration of software licensing for simulation purposes (e.g. Simpack for vehicle dynamics). The VM is provided from the open-source tool called VirtualBox Oracle, which helps to centrally apply more than one VM in a software-based environment. Besides, the SSH technology is applied to access to the VM from the hardware for network system security purposes. The direct access would also be possible by using a SSH-client called MobaXterm, where the user can easily upload and download different files and directly use the CLI of the VM user-interactively and -friendly.

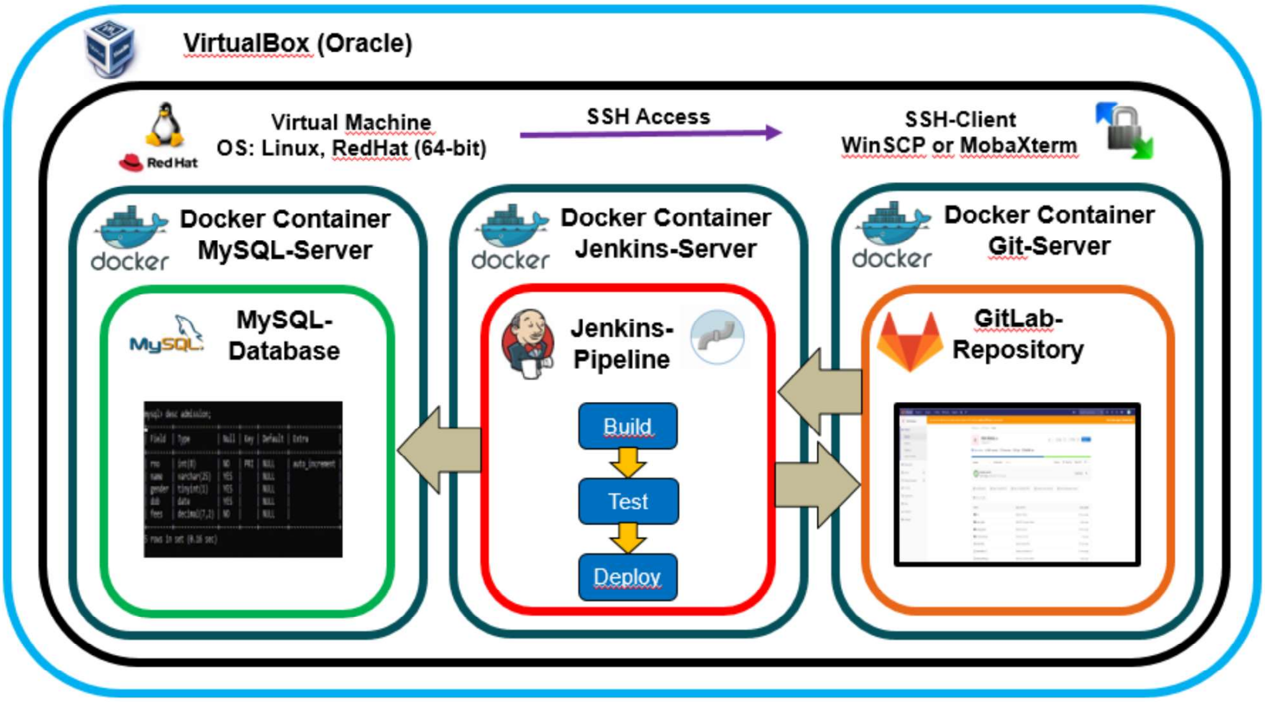

Fig. 2: A conceptual design of the integration environment. (© TU Wien MIVP)

6.4 Integration Process

In this subsection, the above mentioned interfaces and adapters are addressed into the previously proposed integration process of the multi domain model to integrate all the available models and data into the R4F platform.

In Fig. 3, the typical integration process by applying the above mentioned conceptual integration environment is demonstrated, which includes graph DBMS as Docker containerized, the proposed interfaces and adapters, different digital sub-models and datasets of different use cases and automated model visualization additionally. The graph database provides optimal searching functionality by extracting many different sub-model characteristics from the provided digital assets by the Jenkins pipeline and then generating the pipeline with different sensibly linked graphs. The FMUs of the assets, including the XML and DLL formats, are described with annotations by using the ontology with the OWL format (see section 6.5 for further information). Moreover, the FMUs are to be directly uploaded into the GitLab repository, from where the Jenkins pipeline can execute their simulation with help of the SCM interface as mentioned before. Besides, the simulation results coming from the assets can be written back into the repository via git push command for reviewing purposes. On the other hand, SQL queries can be executed by the pipeline to accomplish the data exchange between the MySQL relational DBMS, the Jenkins server and the automated visualization platform called Aardvark. Besides, the JSON format in the output files is likely used to visualize the simulation outputs, because it is easy to be read by the visualization tool.

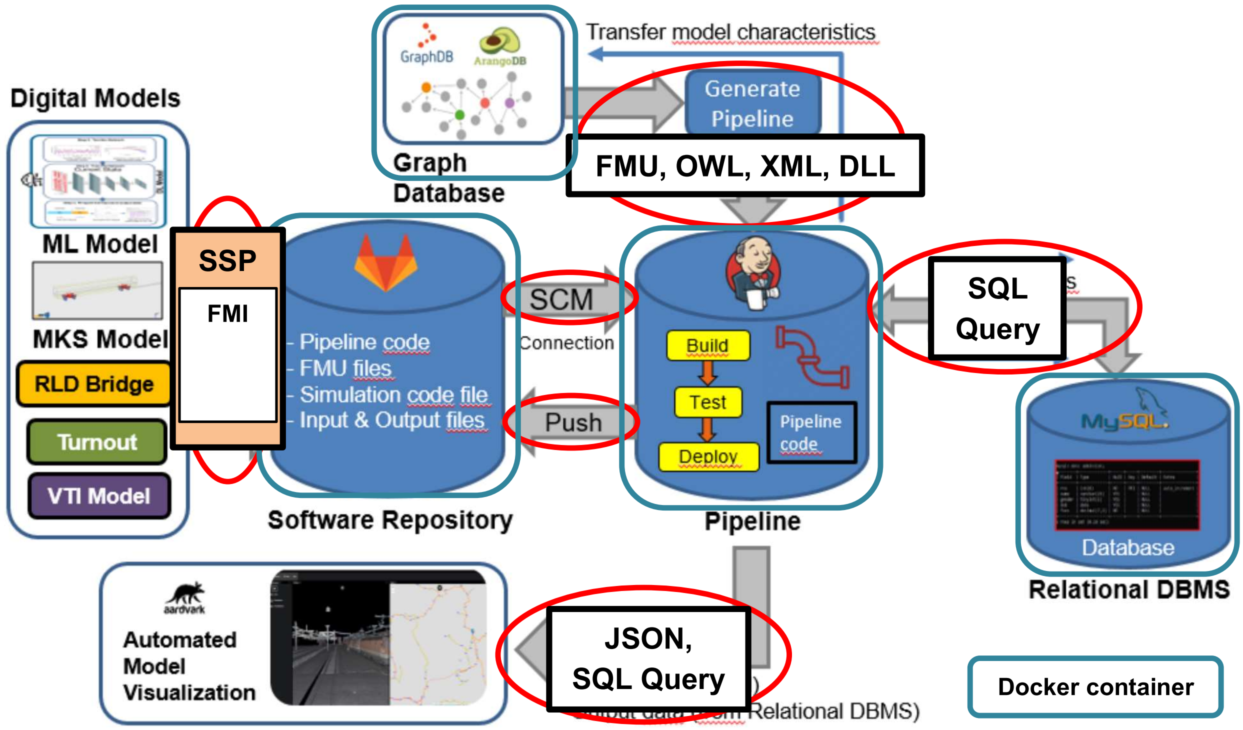

Fig. 3: The demonstrated integration process of different assets. (© TU Wien MIVP)

In future, all the above mentioned Docker containers, including the main software tools, are to be orchestrated by using Kubernetes, so that they can be interconnected through master and node clusters, which directly connect to the VM. It would also help to withdraw the isolation disadvantages by directly connecting different license servers such as Simpack, Matlab, etc. to the containers caused by the containerization technology. Finally, the whole application is planned to be deployed by using Terraform for provisioning and managing the Kubernetes and for infrastructure automation as well.

6.5 Ontology-based Annotation and Information

The FMI Standard is an ideal tool when it comes to assemble full-system simulations from individual simulations. However large-scale systems of multiple domains have a various number of elements and interconnections and therefore can make the combination of models complicated between domains. An ontology is one possible solution to this problem. In fact, an ontology can manage knowledge of input and output variables, their type and unit in order to connect multiple FMUs to a simulation topology. To assist the automation process, the ontology described in Deliverable 1.1.3 is used and a Graphical User Interface (GUI) is developed to assist the manual annotation process. The GUI is able to obtain some data from the FMU itself, by reading and interpreting the modelDescription.xml document contained within the FMU. However not all data can be automatically annotated to that, so the user can add additional information, based on the ontology, and is able to check whether the read data is correct. The resulting annotation is stored as an OWL file within the FMU zip-file alongside the originally existing (.xml and .dll) files. When setting up the modular simulation consisting of several FMUs, the semantic descriptions of the

FMUs within the OWL files are extracted and merged. An algorithm based on inference matches the required FMUs. Fig. 4 provides an overview of the process.

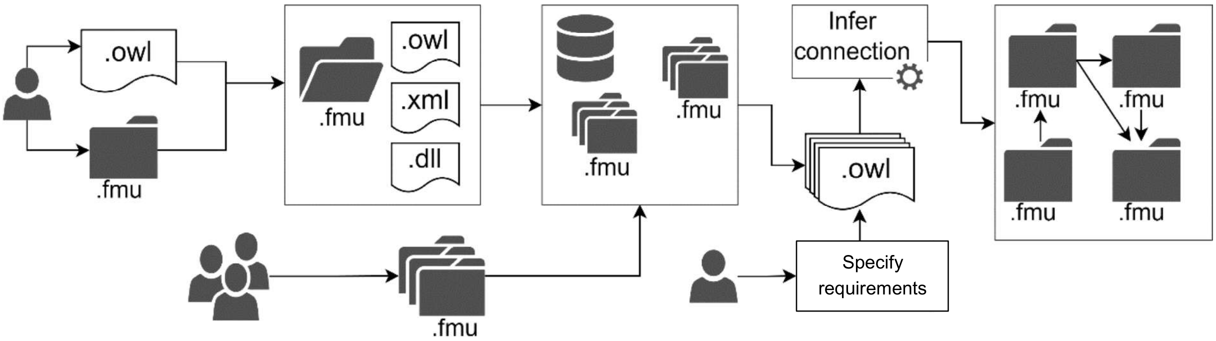

Fig. 4: Semantic process overview with interface standards and adapters. (© TU Wien MIVP)

7 Conclusion

This deliverable aimed to give an idea about the proposed interfaces and adapters, which help to improve, optimize and accelerate the integration, interoperation and ontology engineering processes of all the digital assets belonging to the holistic large-scale railway infrastructure system in the R4F platform. These interface standards also play a crucial role to build the bridge between the software tools and simulation models and dataset in terms of platform-independency, -compatibility and user-interactivity. Besides, the adapters such as the VM, SSH-client, Docker containerization technology and GUI tool for the ontology application assist to communicate and interact all the software tools with each other in the proposed conceptual virtual environment.

In future, the conceptual environment is to be enhanced with containerized graph DBMS, container orchestration and application deployment processes, so that all the Docker containers including the previously mentioned software tools and digital assets can be properly interoperated in the R4F platform.

8 References

[1] Functional Mock-up Interface (FMI): https://fmi-standard.org/

[2] PythonFMU: https://pypi.org/project/pythonfmu/

[3] fmpy: https://pypi.org/project/FMPy/

[4] Simpack: https://www.3ds.com/de/produkte-und-services/simulia/produkte/simpack/

[5] System Structure and Parametrization (SSP): https://ssp-standard.org/

[6] Model.Connect: https://www.avl.com/-/model-connect-

[7] Docker Hub: https://hub.docker.com/