Simulation Model Reference Library: A new tool to promote simulation models reusability

Workshop on Simulation for European Space Programmes (SESP)
24-26 March 2015

ESA-ESTEC, Noordwijk, The Netherlands

‘Yago Isasi[1], Ramón Noguerón[1], Quirien Wijnands[2]

[1] *LuxSpace Sàrl*
9, Rue Pierre Werner Betzdorf 6832 Luxemburg
Email:isasi@luxspace.lu, nogueron@luxspace.lu

[2] *ESA/ESTEC*
Keplerlaan 1 2200 AG Noordwijk The Netherlands
E-mail: quirien.wijnands@esa.int

ABSTRACT

The Simulation Model Reference Library (SMRL or SiMoReLi) activity is an initiative in the General Support Technology Programme (GSTP) in order to promote and increase the reuse and exchange of simulation models between different stakeholders in the space domain. In this paper, we discuss several aspects related to the construction of this library of simulation models. First, a definition of the metadata for simulation models is provided. This helps to understand which “knowledge” should be preserved in the library in terms of necessary simulation model attributes and artefacts for creating a truly reusable model. Second, we describe the software engineering processes involved in the creation of a simulation model and how they can benefit from having a common library that provides not only models, but also guidelines and templates to facilitate and promote reuse. Finally, the paper provides a list of requirements that the library shall fulfill, to become an open and collaborative environment for the reuse and development of simulation models. Also a preliminary architectural description is provided.

INTRODUCTION

The use of modelling and simulation for supporting the design and validation of space systems is an area of constant growth, evolution and technical innovation. At the same time, the design, development and verification of the required simulation models is a specialized field of expertise and moreover a time-consuming and costly process. Unfortunately, tools for managing and sharing the knowledge related to simulation models are often not **publically** available. For that reason, the definition and initial population of a reference library of simulation modelling is meant to contribute to the increase, reuse and exchange of simulation models between different stakeholders of the space industry and research institutions. It is expected that the SMRL will help in creating simulation models more efficiently by reusing previous knowledge. At the same time the SMRL templates and guidelines can help in creating qualitatively good model that contain all the metadata and knowledge that are truly reusable.

The reusability of simulation models has been treated in the literature from several perspectives. In the article “Reusability of simulation models” ([1].), the concept of a model fragment library is introduced. The use of generic ontologies of the physical systems and the proper definition of the model metadata are described as a critical part to preserve and enhance reusability. In the “Simulation model reuse: definitions, benefits and obstacles” ([2].) different types of reuse are described, going from full model reuse down to component reuse, function reuse or code scavenging. Also, the relevance of having deployed a proper reuse strategy that can evaluate costs and benefits before reuse is formalized by defining an average cost/use ratio. Focusing on reusability in the space system domain, the “Collaborative Development of a Space System Simulation Model” ([3]), proposes an interesting reuse flow: the flow browses through different model artefacts, following a bottom-up approach to evaluate its reusability. The approach departs from the analysis of the implementation and flows down to the requirement definition. In addition, initiatives such as the Simulation Model Portability Standard (SMP) and the definition of Reference Architectures such as the Reference Architecture (REFA) [4] or the Space Simulation Reference Architecture (SSRA) [5] have been put into place, as mechanisms to increase the portability and reuse of simulation models between different projects as well as between different infrastructures within the same project. In the same direction the “System modelling and simulation” [6] ECSS Technical Memorandum presents a number of system simulation facilities together with some hints to help reusing models between them.

The SMRL is expected to support the development processes of a simulation model. For example, in the requirement engineering process, requirements from other simulation models can be used to build a new model requirement specification, by directly reusing or adapting them. For that reason, the preservation of knowledge and life-cycle data (e.g. model requirements, design, parameterisation, test procedures, reference results) of a model within the library is seen as equally important as the model sources. In some cases, however, simulation model reuse is not straightforward or maybe not possible at all. Issues can occur like the lack of available information of the simulation model, a steep learning curve to familiarize and use the model and the technical constraints or the validity of the model. Therefore, an initial task of evaluation of the reusable candidates is necessary. This task will estimate whether it is more cost effective and lower risk to develop a new model or to reengineer and validate an existing model for a use which was not originally foreseen.

SIMULATION MODELS METADATA

Simulation modelling and in particular simulation models, constitute the core knowledge domain used by the SMRL. A simulation model is a digital prototype, as a software component, that represents, in part or completely, a physical object, a system, a process, an idea or a particular behavior of the real world. Its typical objective is to mimic the real or original behavior as accurately as possible. Through this, designers and engineers can analyze and study the modelled systems without having them in hand.

Model attributes

A simulation model can be characterized by a set of attributes that will constitute a part of its metadata. The attributes can provide information in many different contexts of the model. Within the scope of this project a set of model attributes has already been identified. The current model attributes are:

- **Identification attributes**: attributes that can be used to identify the model. Some examples are name, description, author, creation date, support contact information or model author website.
- **Taxonomy attributes:** attributes for model classification. To define the taxonomy scheme (criteria of classification) a hierarchical tree structure is used. This tree represents a hierarchical decomposition of all possible models following some concrete criteria. This decomposition helps reduce the complexity in the management of the model classification and will be a key element for searching simulation models within the library. The diagram in Fig. 1 is one decomposition of the “Universe” tree, whose branches represent all the possible categories of simulation models :

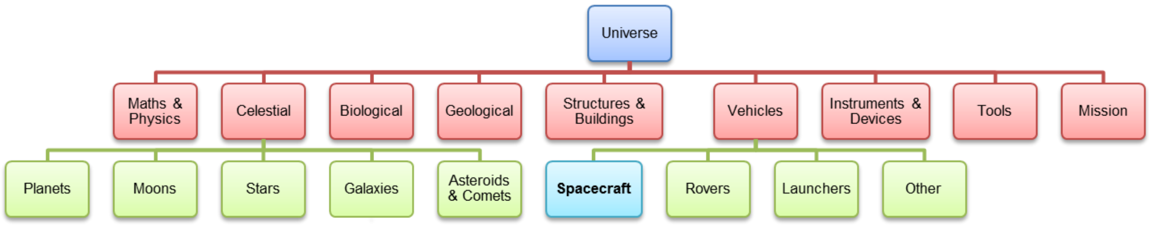

Fig. 1. Hierarchical tree representing the Universe

Fig. 2 represents the spacecraft hierarchical tree indicating a further decomposition of the Spacecraft element that includes (satellite, orbiters, probes, etc…) using physical set division criteria into systems and subsystems.

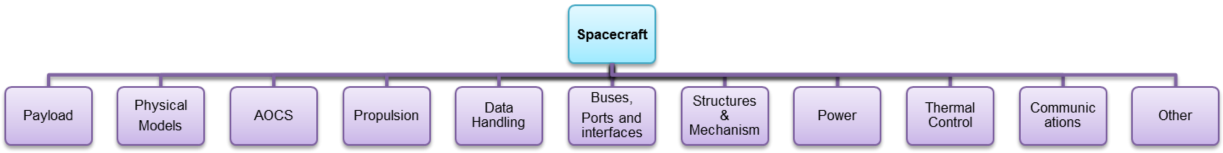

Fig. 2. Spacecraft Tree. A hierarchical decomposition of a Spacecraft element

- **Simulation facility**: This attribute holds the facilities where the model has been used according to the definition in [6]. Some values are System Concept Simulator (SCS), Mission Performance Simulator (MPS), Functional Engineering Simulator (FES), Functional Validation Test-bench (SVT), Software Validation Facility - Software Configuration (SVF-SW), Software Validation Facility - HIL Configuration (SVF-HW), Spacecraft AIV Simulator (AIV), AIV Simulator with S/C Hardware in the Loop (AIV-HW) or Ground System Test Simulator (GSS), Operations, Training and Maintenance Simulator
- **Simulation environment:** The runtime environment, on which the model has been implemented, executed and/or tested (used). Known simulation environments are for example Matlab/Simulink, SIMSAT, Basiles, SimTG, SimVis or Eurosim.
- **(Abstract) Model Architecture:** the software architecture reference used (i.e. object model). Known architectures are SMP2, SSRA, Reference Architecture (REFA) or ISIS[8].
- **Functional context:** a context being a field of application. These fields can be related, but not limited, to branches or groups of physics. However possible keywords are not generally too formal in this case. Typical contexts are: dynamics, power, thermal or magnetic.
- **Fidelity:** refers to the degree to which a simulation model reproduces the state and behavior of a real world object, feature or condition. To capture the fidelity of the model, levels given in [6] are used as a starting point. Those include exact, accurate, emulated, functionality, plausible or realistic, representative, static.
- **Time modelling:** Continuous versus discrete time models.
- **Output response:** Deterministic versus stochastic output responses.
- **Implementation language:** Software programming language used to implement the model. Typical values are Matlab/Simulink, C++, C, Java and FORTRAN.
- **Operating system:** The target OS of the simulation model, if not generic. There are simulation models which are only valid for a certain platform or a number of platforms.
- **Target hardware:** The target hardware for which the simulation model was targeted, if any.
- **Dependencies:** The model dependencies on other simulation models or software components.

Other attributes are linked to the availability of certain artefacts considered essential to achieve true reusability. Those artefacts include documentation artefacts such as a requirement specification, a detailed design, a test plan, a test result or a user manual among others.

The complete set of available attributes, defaults values and explanations details are further described in [9].

Model artefacts

A simulation model comes with a set of associated data and documentation artefacts. These artefacts have many purposes such as describing the function, architecture, and design of the model; supporting its configuration and the execution or supporting the building process. The preservation, guidance and support to all of these artefacts in a consistent way is one of the main objectives of the SMRL. These possible artefacts can be divided in several groups, such as:

- **Documentation artefacts**: General model documentation such as formal ECSS documents, readme files, software release notes, Change Log files, Model manufacturer supporting documents(if any), webpages, wikipages, etc.
- **Design/Development artefacts**: Includes supporting diagrams that are directly linked to the model design or development. This type of artefact typically includes standard modelling/design diagrams e.g. UML diagrams, SysML diagrams, flowcharts or the proprietary files generated by a design or IDE tool used to develop the model.
- **Configuration artefacts:** Under configuration artefacts we include all types of files that are used for the initialization of parameters for configuring a simulation model. The nature of these artefacts could be very diverse. Some examples are: XML, property, Excel, database (e.g. SQL) or CAD/CAM files. Those artefacts are used by the model in order to setup its initial state or to define or control its behavior.
- **Model Sources artefacts:** This group of artefacts includes the source files constituting the model and the output of the compilation processes, if any, including object files and executables.
- **Build and install related artefacts:** This group of artefacts includes the support files necessary for the building process of the model or the project where the model is included. Typically they are: Ant, Maven files or shell, bat, python, TCL scripts among others.

- **Testing artefacts:** Model development involves a number of test activities. As a result of these processes a number of artefacts related to testing are generated. These artefacts may be: test source code, test configuration files, files used by the test code to execute the test and to configure the pass/fail criteria, test reference files, files that are used by the test to validate the results of the test execution, test result files, files generated by the tests including the results of the test execution and the pass/fail output. Script files to run the tests.
- **License related artefacts:** artefacts related to licensing information applicable to the model. License artefacts are: a license contract agreement information, license keys, serial numbers or license disclaimers.

THE SMRL FRAMEWORK VISION

Objectives

The SMRL Framework is a knowledge management tool in the Simulation Models domain. This infrastructure/tool acquires and provides available simulation model knowledge to the users of the library in order to facilitate its reuse. As such the SMRL can play a relevant role in different processes during the development of a simulation model. Not only will it provide model information and artefacts, but also guidelines and templates, e.g. documentation, that are needed to develop new and reusable simulation models. As such the SMRL can serve as a starting point for the creation of simulation models in a more efficient manner. In the next sections, the different commonly known engineering processes that occur during the development of a simulation model are mapped to the usage of the SMRL. This analysis will bring some key artefacts that should come together with the model source code for quality and reuse reasons. Those artefacts constitute the model lifecycle data and shall also be stored in the library. The model development processes were defined using the ECSS software engineering standard [12] as starting point.

Model search / discovery process

The model discovery process is the first step when searching for potential reusable simulation models (Fig.3). The SMRL user gives explicit model requirements by providing a list of relevant model attributes. These attributes will be used as search keywords. At this point the library engine searches for models to find a match; the search process is driven by a similarity metric that computes the similarity of the problem description, i.e., the current user requirements and the available models. When the list of models is retrieved from the SMRL storage, the results are displayed to the user through the SMRL user interface. If the user is not satisfied with these suggestions, the attributes can be modified in the query and a new search cycle is started. Related models can also be returned by the search engine.

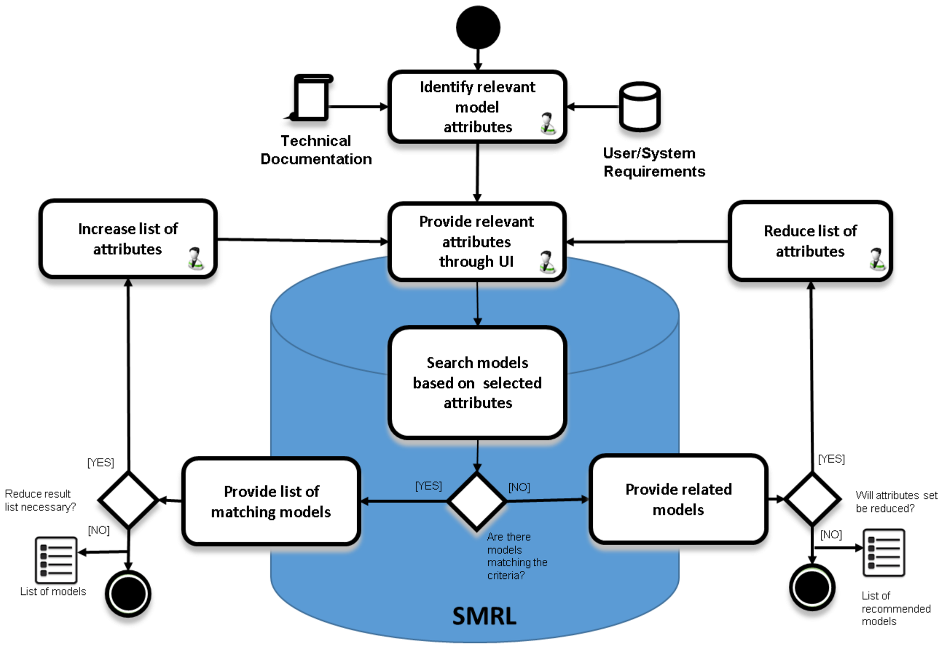

Fig.3. Model discovery process

Requirements engineering process

The requirements engineering process encompasses the set of tasks undertaken by the engineering team to define the simulation model software requirements. This process happens in the early stages of the software development before the architecture definition and detailed design. The main output of this task is a Software Requirement Specification (SRS), the software engineers' understanding of the high level system requirements and dependencies. An Interface Control Document (ICD) could also support the interface requirement definition. The requirement engineering flow is presented in Fig. 4 and follows the "Software requirements analysis" described in [12]. The SMRL will be supporting this process as follows: When the search engine returns model candidates, the user can review and evaluate their specification, interface definition and logical model and then, subjectively, decide which parts can be reused. Finally, the SMRL will be providing templates for the standard ECSS documentation, tailored to the needs of simulation models. Those include the SRS, ICD documents and the Software Verification Report (SVR) that will be used when verifying the requirements and the logical model. These documents will be part of the artefacts stored and managed by the library.

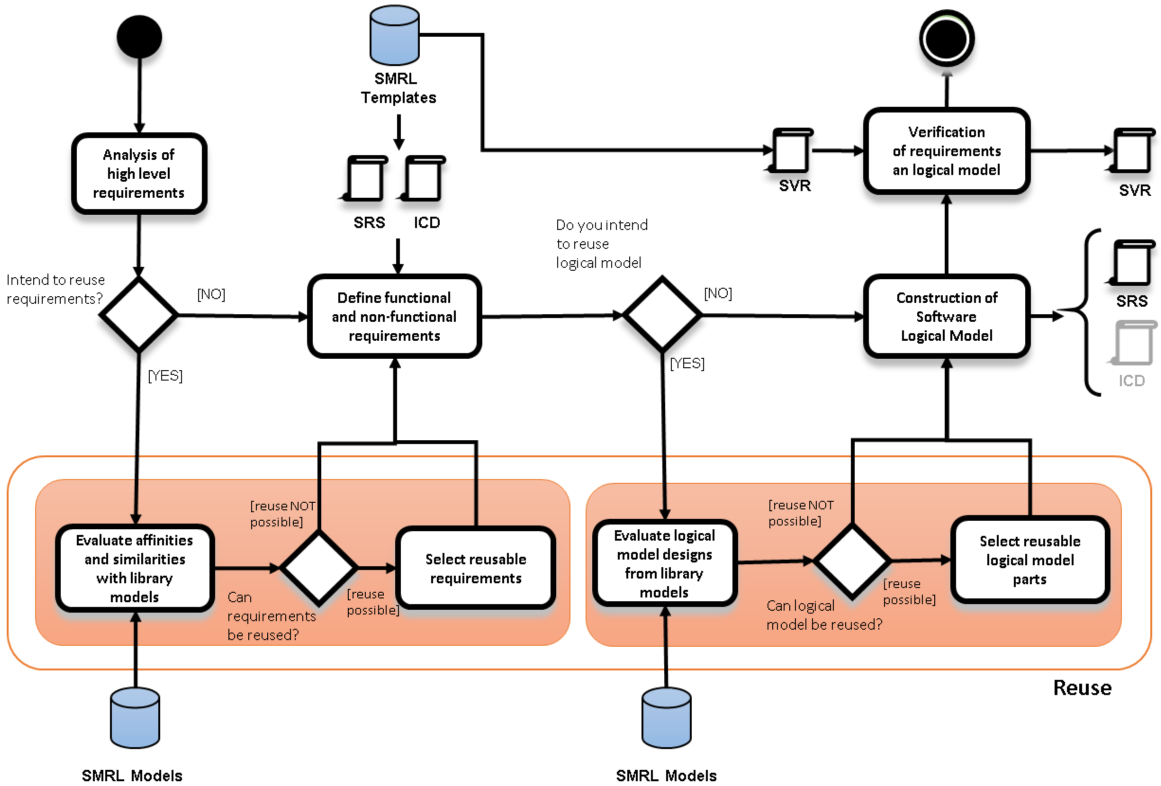

Fig. 4. Requirement engineering process with reuse tasks

Design process

The design process consists on the set of activities undertaken by the engineering team in order to derive the model design from available requirements. We considered aspects of the "software architectural design" and the "software design" processes describe in [12], and adapted them when defining the architecture and design of a simulation model. The SMRL will be supporting this process as follows: it will provide design artefacts of the model candidates to support the reuse of the architecture, interfaces and detailed designs. Also, it will provide templates and guidelines for the realization of the standard design documentation, i.e. Software Design Description (SDD), Software User Manual (SUM) and Software Reuse File (SRF).
This process comprises the definition of the model architecture, interfaces and the detailed design; the development of the architecture and detailed design can take advantage of reuse by evaluation and selection of the applicable parts of

the reused models. The reusable parts of the design will be added when creating the detailed design of the new simulation model. From this activity the new SDD, SUM, SRF, and optionally the ICD design, of the new simulation model are relevant artefacts that will be stored and managed by the library (See Fig. 5).

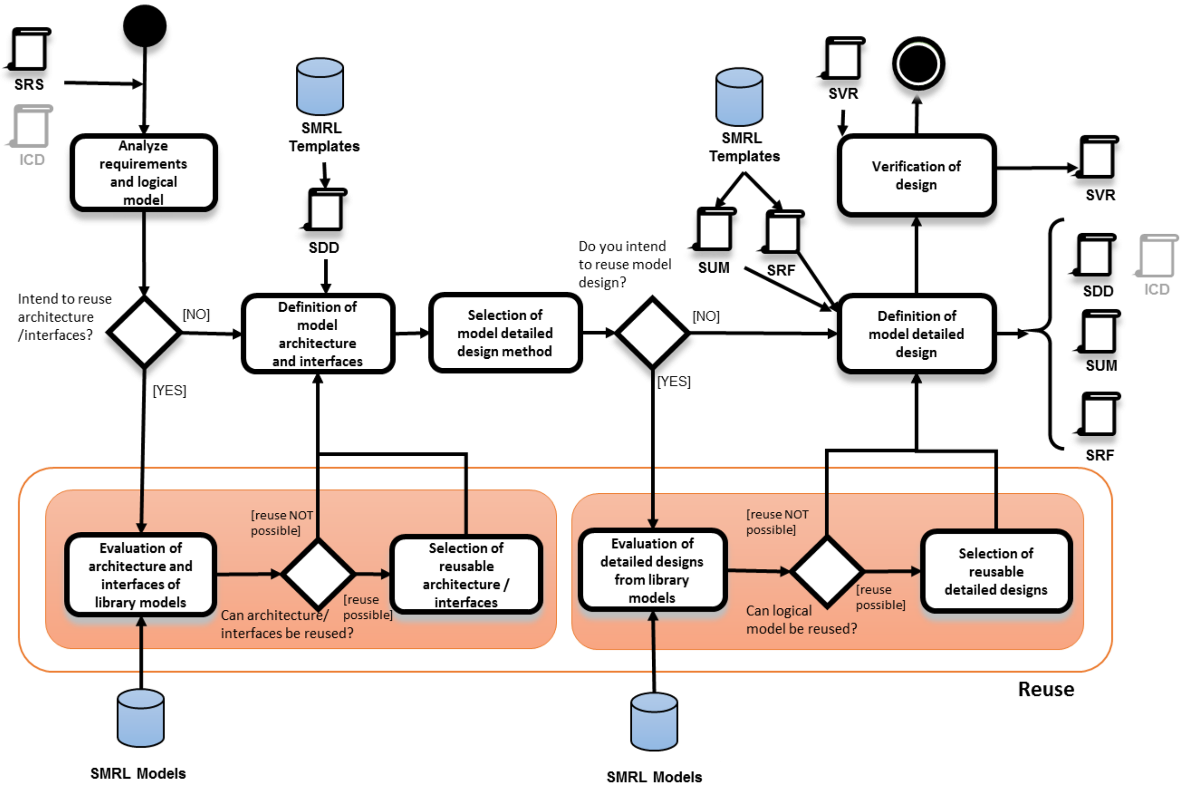

Fig. 5. Simulation model design process with reuse tasks

Construction process

Under the term "construction process" we include the implementation and testing processes. In the coding process, the implementation of the simulation model and the build procedures to compile and link the simulation model shall be created. The SMRL helps to provide model source codes, build scripts and documentation (e.g. SDD, SUM) of the reused models. The engineers shall evaluate and select which parts could be reused by the model being developed. In the testing process, the SMRL provides test documentation such as a test plan, including test cases, e.g. inputs, outputs and test criteria, as well as test procedures, test results and the supporting artefacts such as test codes, test data and test scripts. Reusing these artefacts could be a starting point for the testing activities of the new model. The engineers considering reuse shall evaluate and select which of those artefacts are applicable and can be reused. The availability of the test documentation and artefacts is important to generate confidence and credibility in the behaviour of the models managed by the library.

Validation and Verification

As defined in [15], the simulation model validation is the process of determining the degree to which a simulation model and its associated data are an accurate representation of the real object or idea, from the perspective of the intended uses of the model. Part of the validation will be covered by the tests applied to the simulation model during the construction process. When testing is not possible, the validation will be carried out by analysis, inspection or review of design.
The verification is the process of determining that a model implementation and its associated data accurately represent the engineer's conceptual description and specifications [15]. The verification tasks that occur during the engineering processes are described in detail in [12].

SMRL FRAMEWORK REQUIREMENTS AND FUNCTIONALITY

Based on the above and taking into account the original objectives of the activity, the following high level requirements were identified for the library and its supporting infrastructure. The SMRL shall:

- Provide a repository to store, manage and catalogue the simulation model and its artefacts.
- Provide a search engine, for searching and retrieving simulation models from the library.
- Provide model insert and modification features for authorized users.
- Address the license terms and conditions applicable to the use of a simulation model.
- Address the management of intellectual property rights (IPR´s).
- Provide an access control for user identification and profile management.
- Support bug track and reporting systems linked to the available simulation models. Support the Collaborative development environment, i.e. using version control systems and discussion boards.
- Support the user in creating or adapting simulation models by providing guidelines and template artefacts.
- Support the usage of space and software engineering standards.
- Accept any data format or modelling and simulation tooling.
- Define the responsibility of the SMRL infrastructure, and guarantee of non-liability on the host of the SMRL in case of breach of IPR.

To support all the previous functionality, the foreseen SMRL solution will be based on a web application that harmonizes the available backend resources. Those resources include the storage of the model information, its artefacts, the version control system and external infrastructure and services (e.g. the ESTEC Collaborative Software Development Environment (CSDE)). Fig. 6 shows the SMRL server and the different layers of the SMRL application. The model we used is a multi-layered architecture with presentation, service, business and data tiers. The (server) presentation layer contains the views and controllers for the client flows. It communicates with the business layer, which is in charge of the fundamental business functionality of the application such as the search, insert or retrieve of the simulation models. The data access layer embraces all the components and connectors necessary to communicate with the data sources, such as metadata information or model repositories, and with the external services. The service layer is in charge of defining the interfaces and message formats used by external system willing to access the library information.

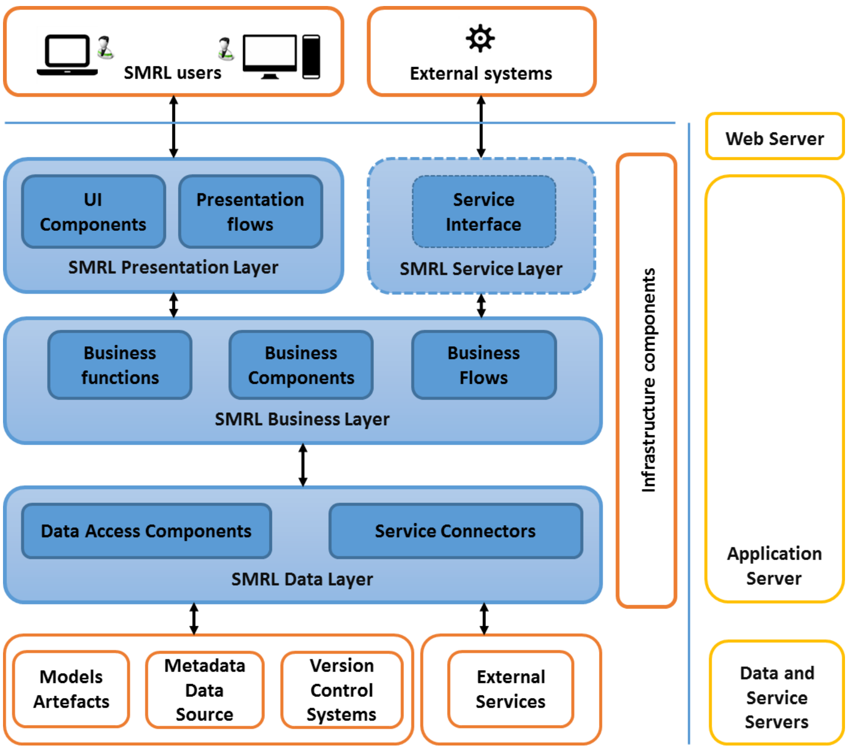

Fig. 6. Preliminary solution architecture for the SMRL

CONCLUSIONS AND STATUS

In this paper we have introduced the Simulation Model Reference Library (SMRL), a tool for the management of model knowledge and reuse of simulation models in the space domain. Model metadata has been defined by using multiple attributes and provides the foundation of the SMRL concept. The software engineering processes involved in the creation of a simulation model have also been discussed and taken into account. In this respect the SMRL can be used to support and speed up those processes, in particular, when reuse flows are taken into consideration.

The library is expected to be a portal for all simulation model users in the space domain and an open simulation model knowledge management tool. The concepts presented in this paper will be used for the development of the knowledge management tools. In The SMRL activity, the work is currently focussed on the SMRL infrastructure definition and implementation phase, including aspects of licensing and security. After this phase, an initial model population of the SMRL is foreseen. It is expected that the SMRL will be available for public use as from early 2016 onwards.

REFERENCES

[1] Anita Pos, Pim Borst, Jan L. Top, Hans Akkermans: Reusability of simulation models. Knowledge-Based Systems 9(2): 119-125 (1996)

[2] Simulation model reuse: definitions, benefits and obstacles S Robinson, RE Nance, RJ Paul, M Pidd, SJE Taylor Simulation modelling practice and theory 12 (7), 479-494 (2004)

[3] Daniel Ludtke, Jean-Sebastien Ardaens, Meenakshi Deshmukh, Rosa Paris Lopez, Andy Braukhane, Ivanka Pelivan, Stephan Theil, Andreas Gerndt, "Collaborative Development and Cataloging of Simulation and Calculation Models for Space Systems", WETICE, 2012, 22nd IEEE International WETICE Conference (WETICE 2013), 22nd IEEE International WETICE Conference (WETICE 2013) 2012, pp. 244-249, doi:10.1109/WETICE.2012.64.

[4] SMP 2.0 Handbook, EGOS-SIM-GEN-TN-0099, Issue 1.2, 28-Oct-2005 [2] Spacecraft Simulator Reference Architecture Specification, REFA, Software Requirements Specification Volume 1 (EGOS-SIM-REFA-SRS-1001), Issue 1.2, 12-Dec-2008.

[5] The Space Simulation Reference Architecture – Reference Architecture Specification Volumes 1-3, (SSRA.REP.001, SSRA.REP.002, SSRA.REP.003), Issue 1.0, 27-May-2010.

[6] ECSS-E-TM-10-21A: System Modeling and Simulation. First issue, April 2010. [Online]. Available: www.ecss.nl

[7] V. Schaus, K. Grossekatthofer, D. Lüdke, and A. Gerndt, Collaborative development of a space system simulation model, in Proc. 20th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE ’11), 2011,pp. 164-169.

[8] ISIS Training Operation and Maintenance Interface Specification (ISIS-SIM-IF-305-CNES), Issue 4.0, 9-Sept-2011.

[9] Y. Isasi, R.Nogueron, SMRL: Models Analysis and Classification Report. Unpublished.

[10] Y. Isasi, R.Nogueron, SMRL: Models Knowledge and Life-cycle Data report. Unpublished.

[12] Y. Isasi, R.Nogueron, SMRL: Guidelines and Processes Report. Unpublished.

[13] Y. Isasi, R.Nogueron, SMRL: Concept and Specification of the Simulation Models. Unpublished.

[14] ECSS-E-ST-40C, Space Engineering: Software ESA Std., March-2009 [Online]. Available: www.ecss.nl

[15] DoD Modelling and Simulation (M&S) Verification, Validation, and Accreditation (VV&A),), DoD Instruction 5000.61, December 9, 2009.