Role-based Simulation Object Management as an Enabler for Democratisation in Collaborative Virtual Engineering

Clemens Hepperle, Christian Gnandt (TESIS DYNAware GmbH, Germany)

1 Democratisation in collaborative virtual engineering

Within virtual development simulation gains more and more importance to support the dimensioning, validation and optimisation of technical systems. Especially when dealing with complex systems such as vehicles or aircrafts integrating manifold control units, simulation is a key enabler to early reflect upon the interaction of highly variant and interrelated system components. The considered electrical and mechanical components are mostly developed and administered within distributed environments, which means involving different departments in and outside the company. An example for such a constellation is a Multi-ECU HiL-system for virtual characteristic rating of vehicle dynamics control systems [1]. In this example different control units – such as a damper control unit, a Haldex-clutch control unit and a ESC control unit – in combination with relevant simulation models are evaluated within tests along the product evolution process from early planning until the product is launched. Looking at vehicle dynamics simulation multiple challenges are associated with such development environments [2]. Besides the engineering experience of setting up test environments, also more administrative issues such as the coordination of communication within collaborative working set-ups as well as the comfortable and user-specific tool and data access have to be considered in order to increase the quality and acceptance of simulation results. In this context there is an increasing demand for instruments to enhance democratisation of simulation [3]. Such aspects are not only relevant to vehicle dynamics but also to other areas like the energy management for vehicles [4] and other dynamic simulation set-ups in manifold industrial sectors. The similarity of such set-ups is that different stakeholders share their knowledge and data to be able to create respective simulation environments capable of evaluating the behaviour on different system levels.

Answers to the described challenges from a data and process point of view are given by simulation object management [5]. This methodology emphasizes the relevance of the data context in order to provide the engineers and managers in cross-departmental collaboration the right information at the right time. Both the focus on traceable and flexible working with system models and parameterisations as well as the data allocation based on flat data hierarchies and individualised data structures for compatibility with task-dependent toolchains play an important role. Thereby the definition and usage of meta-information to increase the valid reuse of existing data is addressed.

This paper focuses on the support of specific roles in simulation environments by using mentioned mechanisms for goal-oriented and structured access to tools and data. The efficient, convenient and quality conformant data allocation is a key for the democratisation of simulation environments. In consequence users can be more individually guided in their daily work without having to deal with time-consuming data collection and tool setup for their respective tasks. Benefits for the company by an increasing efficiency and quality in elaborating on respective tasks are also given. In section 2 the client-server concept and implementation of role and user dependent information access is presented. In addition the generation of individualised, structured working environments on a flat object structure using rule sets is addressed. Section 3 then focuses on an example for role-based simulation management by providing insights into a multi-user workflow from creating model architectures throughout definition and usage of model configurations. In section 4 conclusions are drawn and an outlook on further activities in the context of role-based virtual engineering is given.

2 General mechanism of role-based simulation object management

Within simulation environments, people take in different roles with certain authorizations and responsibilities. In smaller simulation teams singular persons represent multiple roles – or the other way around one role consists of a heterogeneous set of responsibilities. An example is that a simulation engineer takes in both the role of a model developer, a tester and the project manager. In larger simulation teams, especially in collaborative and distributed environments, there is a tendency to associate certain roles with certain persons with clearly defined tasks. There, an own project manager coordinating tasks, costs and communications exists beside a test developer, a requirements engineer and further persons. All of these constellations have their eligibility depending on the pursued goals and organisational setup. One demand here is that a company or simulation organization needs the flexibility to define their own roles. Especially in IT-supported domains like simulation, the mentioned persons can be seen as users working together in a collaborative tool environment. As

there is a growing demand for simulation accompanied by increasing simulation resources, there is also a need for structured and in consequence more formalized procedures, which are embedded into the collaborative tool environment. Issues like access control in collaborative systems have been discussed for years now [6] and are widely spread from a file-based perspective within simulation. Nevertheless the requirement of really supporting roles in simulation environments and functional development with its specific needs is by far not satisfied yet. On the one hand this yields at providing data in an individually definable structure, on the other hand it means that users are guided within the services they perform on the respective relevant data. This supports the democratisation of simulation and in consequence increases the user acceptance in participating in huge simulation environments.
Providing the right data with the needed structure – particularly to run specific toolchains in local development environments (LDE) – comes along with big efforts in preparing the data setup. Thereby often new files and folders are redundantly generated or even copied. When simulation was run, result files are copied from the local machine to network drives and embedded into manifold reports.
To prevent from simple file handling, databases within a client-server setup are a common approach. Nevertheless there is the conflict that flat data structures in databases are difficult to handle in combination with the data and file structures needed by a simulation system or by the user to keep the overview and quickly comprehend the status of relevant data. For instance in simulation, certain parameterisation files have to be provided in a specific folder of the file structure in order that the used simulation tool is run with the intended model configuration. These issues are addressed by a comprehensive simulation object management considering the data context by defining and making use of meta-information [5]. Besides the extra information for the respective user, the meta-information provides the capability of being used in rule set based configurations for browsing and LDE setups. Individual nodes and multi-layered data structures can be generated still having a common data set. These configurations can be defined by singular users, but it is also possible to setup respective data structures to be applied by specific roles. In consequence a model developer works generally on the same data as the model architect, but the data is provided and structured in the respective role-appropriate way.

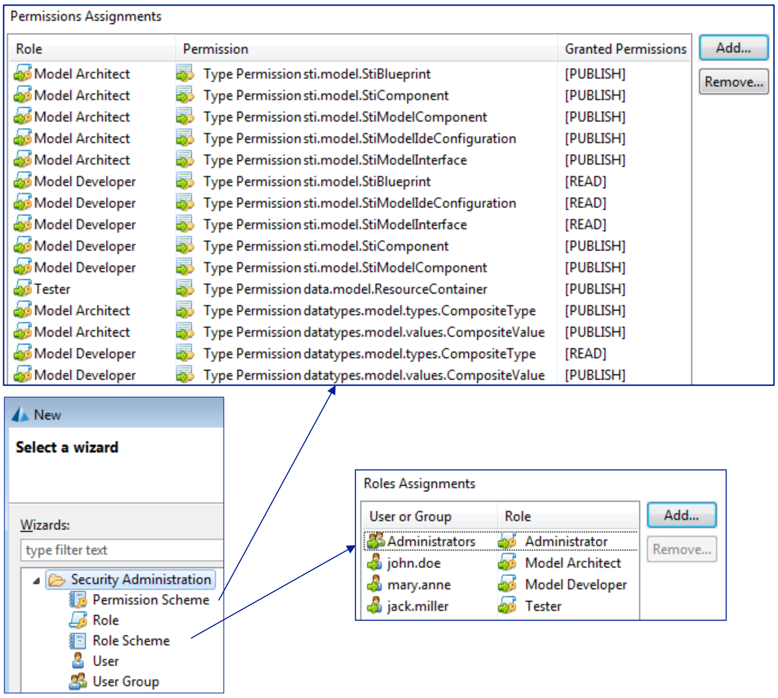

Figure 1: Role-based permission handling

In addition flexibility in role-based virtual engineering is addressed by giving the organization the possibility to setup and handle their own users, roles and associated permissions. One topic is to provide an authentication mechanism for singular users to gain traceability about who is responsible for which artefact. Besides having singular users, also the definition of user groups to deal with high numbers of users is essential. Then there should be the possibility of defining roles. In order to keep the association – what a role is allowed to do – definable for the respective development context, two schemes are used. As shown in Figure 1, one scheme addresses the assignment of users/user groups to roles. The other scheme defines the assignment of roles to service permissions respectively the access to certain object types. In consequence a user can select a role he is associated with and respective object type and service access is provided.

3 Example: working with model architectures and deduced model configurations

Looking at a currently focused scenario at a major automotive OEM, there consist several tasks from defining models (e.g. Simulink), parametrise and in consequence deal with model configurations and further generated artefacts which can be run on respective execution platforms for predefined tests. While all of this work was previously done by one or few persons, with the increased amount of models, parameterisations and test cases due to the variance of product lines – concerning both complete vehicle configurations but also certain subcomponents – also more simulation personnel is involved. Additional roles with certain responsibilities to give simulation models standardized but still customizable structures are considered. From a role perspective such "model architects" focus on the methodological side to define modularized system architectures particularly defining standardized interfaces for exchangeability of (sub-)component models. One big challenge is to provide the right data support and service access for such model architects. Further, role conformant and intuitive tools are needed to be able to deal with the task of defining model architectures. And to make the defined architectures really applicable for further roles and users downwards the workflow and toolchain of configuring and using specific models, also the (re-)usability from a technical point of view has to be given. This means that a model architect should work in a format that the role of e.g. a "model developer" can directly build upon.
Figure 2 shows schematically the data model of a model architecture meeting the mentioned requirements concretised by a simplified model setup for an F14 jet [7]. The model architecture embeds blueprints and slots with predefined obligatory interfaces which then can be detailed by further slots and in consequence also filled with specific model components and assigned code fragments (e.g. Simulink code fragments). Thereby an integrated approach from interface definition throughout to model configurations for certain simulation execution tools is defined. The modularity of model configurations leads to flexibility in exchanging subcomponents and provides a basis for systematic variant handling. Working with model architectures and deduced configurations one can see that a visualization of such dependencies by means of a graphical modelling workbench is very helpful when defining such artefacts. Thus tools which model architects and developers can intuitively work with are provided.

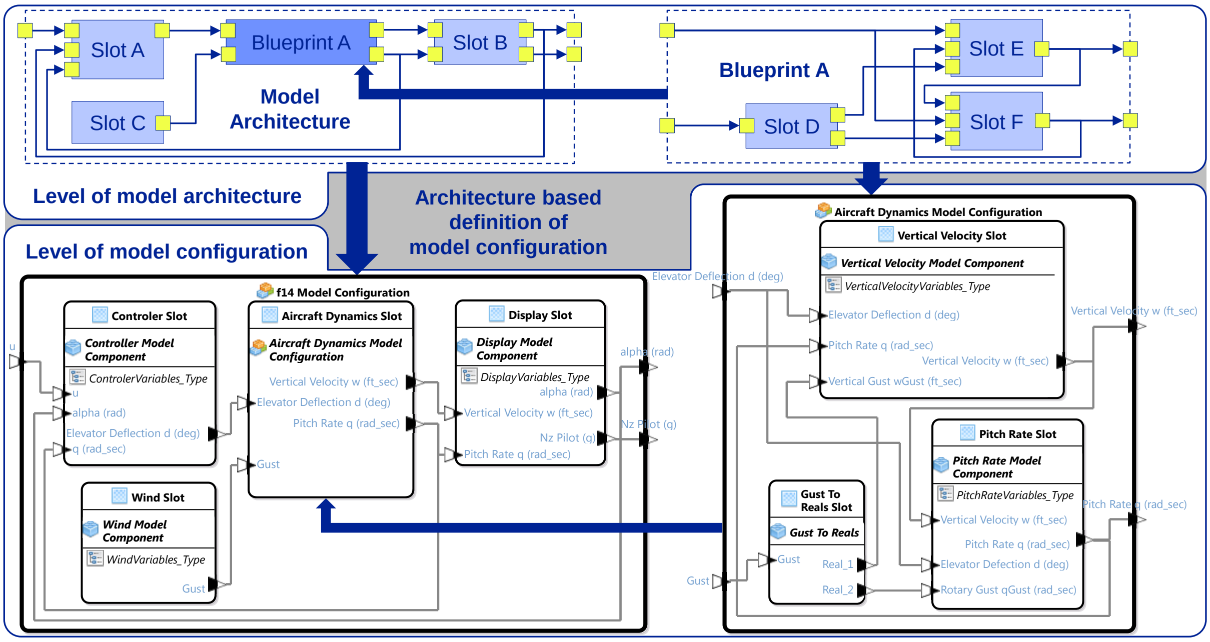

Figure 2: Concept of defining and detailing model architecture

As there are important role-related data and service settings as well as role-conformant tools and user interfaces given, Figure 3 shows an integrated simulation workflow. In this example there are three users, one having the role of a model architect, one being a model developer and one being a tester. Thereby it is also shown how different (also 3rd party) tools are part of the manual but also automated process. Traceability is given from model architecture throughout to the simulation results which are also administered within an object-oriented data management. All artefacts created within the process are handled as objects consisting of history and further relevant meta-information, also created files handled in so called resource containers. As the respective users have both data and tools access matching their demands for solving conveniently and efficiently their respective tasks, this workflow also demonstrates how democratisation within virtual engineering can be established.

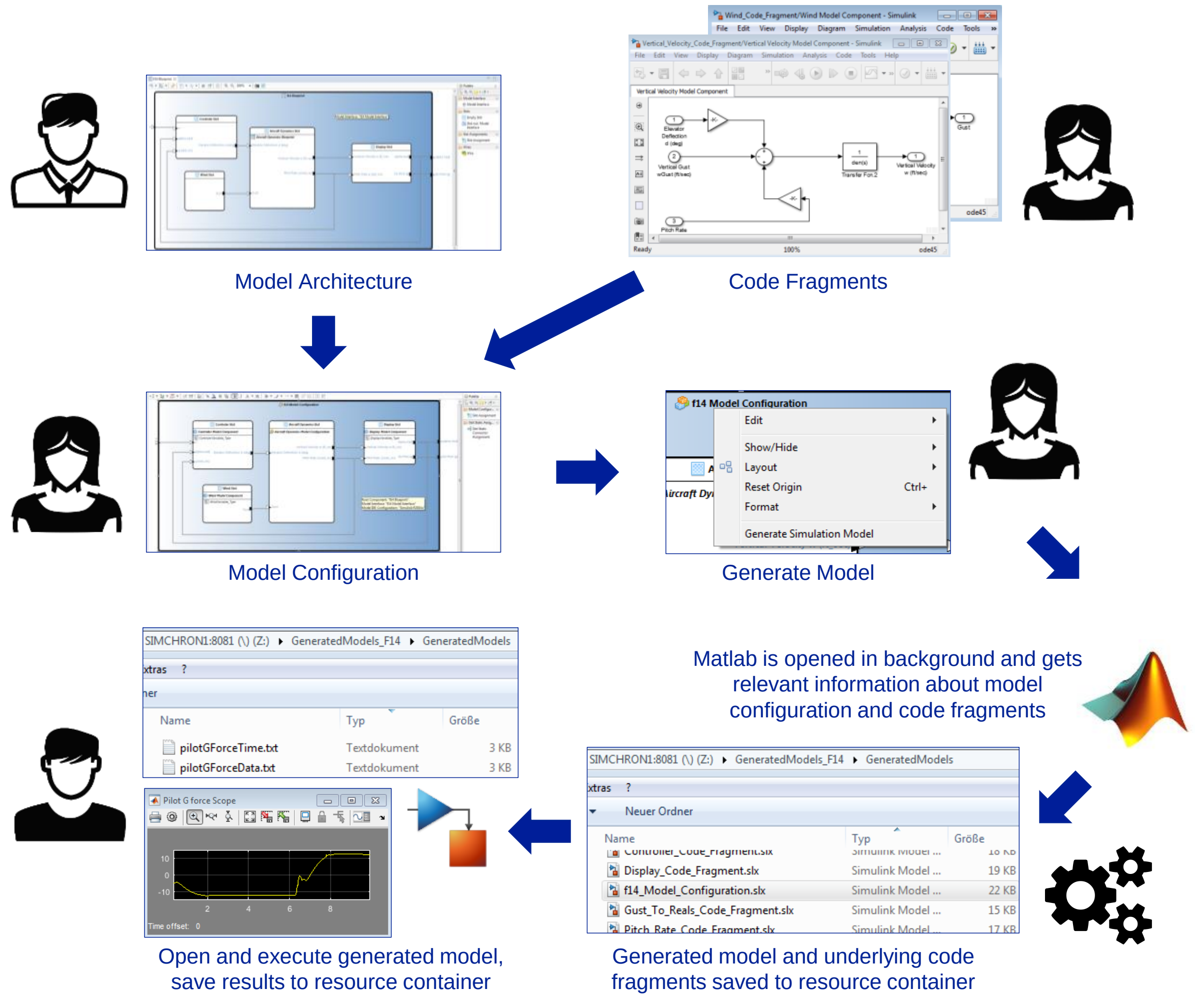

Figure 3: Role-based simulation workflow from model architecture to test execution

4 Conclusions and further work

This paper provides an overview about methods increasing efficiency, traceability and convenience in collaborative virtual engineering focusing on role-based workflows. Besides general approaches to enhance democratisation for simulation, a corresponding multi-role/-user workflow with respective toolchain to derive simulation results based on a reusable model architecture is shown. Currently next steps to link flexible role-based measures to automated process execution are carried out. In this context, the increase of user acceptance in both data access but also in respect to role/user definition and selection from a graphical user interface perspective is crucial.

5 References

[1] Dessort, R., Simon, P., and Pfau, J. 2013. Multi-ECU HiL-Systems for Virtual Characteristic Rating of Vehicle Dynamics Control Systems. In *VDI-Conference Simvec Spezial Simulation Fahrdynamik 2013*, Baden-Baden.

[2] Chucholowski, C. 2014. *The challenge of vehicle dynamics simulation – a critical assessment. ATZ Extra,* 2014/01. http://www.tesis-dynaware.com/fileadmin/Downloads/Presse/vehicle_dynamics_simulation_challenges.pdf. Accessed 22 October 2015.

[3] Faber, B., Davey, C., Felice, M., Diachun, A., Alanoly, J., and Weiss, A. 2015. The Future of Simulation Collaboration in the Automotive Industry. In *ProSTEP iViP Symposium 2015*, Stuttgart.

[4] Zhou, W. 2015. Thermal Design and Simulation of Traction Batteries in Vehicles. In *ELIV – Congress Electronics in Vehicles 2015*, Baden-Baden.

[5] Gnandt, C. and Hepperle, C. 2015. Global Architecture-Based Simulation Object Management with Integration of Local Toolchains. In *NAFEMS World Congress 2015 incorporating the 2nd International Conference on SPDM*, San Diego.

[6] Tolone, W., Ahn, G.-J., Pai, T., and Hong, S.-P. 2005. Access control in collaborative systems. *ACM Computing Surveys (CSUR)* 37, 1, 29–41.

[7] The MathWorks Inc. *Introduction to Profiling Models – F14 example*. http://de.mathworks.com/help/simulink/examples/introduction-to-profiling-models.html. Accessed 30 October 2015.