

Model-driven Data Storage Using Dynamic Object-Relational Mapping

PIET VAN DER PAELT, Department of Mathematics, Royal Military Academy, Belgium

BEN LAUWENS, Department of Mathematics, Royal Military Academy, Belgium

BEAT SIGNER, Web & Information Systems Engineering Lab, Vrije Universiteit Brussel, Belgium

Many programming languages offer discrete-event simulation frameworks, each of them varying in completeness of offered features and ease of use. This research focusses on the SimJulia package (Lauwens et al.), which is an open source discrete-event simulation framework, available within the Julia programming language ecosystem. In the domain of Simulation Tools and Platforms, we present a novel transparent data persistence architecture as an extension of SimJulia. We integrated the PostgresORM (Laugier et al.) package into ResumableFunctions (Lauwens et al.), on which SimJulia depends heavily, by using Julia's language features (metaprogramming). As such, we were able to remove the dependency on a user's knowledge on architectures for persistence. Our contribution aims to improve the usability of the SimJulia package, whilst demonstrating the power of Julia's macro expansion to move towards a dynamic object-relational mapping (ORM) configuration. This intermediary result is part of a broader research project in which we aim to realise a web-based architecture for modelling, simulation and visualisation in support of research and education simulation requirements. SimJulia is offered through Julia's general package repository and currently, must be used within a development environment of choice. As such, it is a widely available open source technology. However, the current implementation lacks functionality to transparently store state variables of a simulation. The exploitation of the data generated by a SimJulia simulation depends entirely on the user's knowledge on technology for persistence and how to interact with it from within the simulation model. To mitigate this shortcoming, we extended both the SimJulia and the ResumableFunctions packages by implementing a transparent probing and data persistence architecture, employing Julia's metaprogramming support. As such, the user is liberated from technological and infrastructural concerns to realise data persistence. Furthermore, the dynamic aspect of our implementation reduces the required configuration with regard to the persistence aspect to a strict minimum. Aside from the aforementioned Julia packages and language features, the persistence infrastructure is realised through the PostgreSQL Relational Database Management Systems (RDBMS). As a result of our implementation, the end-users have the possibility to store data which characterises the evolution of a running simulation transparently. Aside from the requirement of having an RDBMS instance running, there is no additional configuration burden. The data is available in the database and can be exploited using any technology of choice. Apart from the gain in functionality, there is also a technological gain which is situated at the usage of metaprogramming to divert from a static ORM configuration as often seen in web applications, to a transparent and dynamic configuration. As a part of our broader research project, the current proof-of-concept implementation also demonstrates the exploitation of the stored data in a visualisation which shows the tables in a web interface. The latter is realised using the VueJS package. Through the intermediary of an abstraction layer which employs the dynamic ORM, we were able to provide VueJS with the required view on the data. We are currently considering the possibility to externalise the data based on a REST API for the presented architecture.

Authors' addresses: Piet Van Der Paelt, piet.vanderpaelt@mil.be, Department of Mathematics, Royal Military Academy, Brussels, Belgium; Ben Lauwens, ben.lauwens@mil.be, Department of Mathematics, Royal Military Academy, Brussels, Belgium; Beat Signer, bsigner@vub.be, Web & Information Systems Engineering Lab, Vrije Universiteit Brussel, Brussels, Belgium.

SIMULTECH 2023, July 12–14, 2023, Rome, Italy

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

© 2023 Copyright held by the owner/author(s).