Geophysical Research Abstracts
Vol. 20, EGU2018-7758, 2018
EGU General Assembly 2018

Models of Models!? Using Domain Specific Languages in Environmental Modelling

Will Simm (1), Faiza Samreen (1), Richard Bassett (2), Gordon Blair (1), and Paul Young (2)

(1) Ensemble Group, School of Computing and Communications, Lancaster University, UK (w.simm@lancaster.ac.uk), (2) Lancaster Environment Centre, Lancaster University, UK r.bassett@lancaster.ac.uk

It is a fundamental practice in Software Engineering to create a model of a system. As with environmental models, the software model is an abstraction designed to facilitate understanding of a system, and can take various forms from whiteboard sketches to precise encoded models that support code generation. Model Driven Engineering is a discipline of computer science whereby these precise models are created of software systems that facilitate interaction with, but also provide abstraction from underlying complexity.

As researchers in software engineering, we have observed current practices in environmental modelling, including looking in detail at the Weather Research and Forecasting model (WRF), and have observed significant levels of complexity. Users of WRF require a broad skillset including systems administration, deep knowledge of Unix command line tools, and intimate understanding of filesystems. In addition, there a large number of often repeated tasks such as data transfer, data preparation, model configuration and output data manipulation.

In this session, we present our work on the use of a model driven engineering technique, the development of a "Domain Specific Language" (DSL) that encodes domain knowledge and allows abstraction from the complexity of the computer system, model configuration and data manipulation – leaving this to be controlled by the underlying software model of the system, defined by the DSL. By defining an experiment in terms of the science to be studied rather requiring the user configures the computer system, will allow the user more time to spend on their experiment.

The advantages of this approach to the modelling community are numerous: i) the complexity of deployment and configuration is abstracted away, dealt with by the software model using containerisation tools such as Docker and code generation; Ii) the model can become platform independent, opening up the possibility of exploiting architectures such as elastic cloud, taking advantage of on-demand compute and new data storage and analytics techniques; iii) the democratisation of the model to those without high performance computing access or computing skills; and iv) allowing the model to be deployed in new configurations and new ensembles controlled by DSL.