

Extending ScalaTion, a Domain-Specific Language for Modeling & Simulation, for Simulation Optimization

*Michael E. Cotterell, John A. Miller, Jun Han and Tom Horton,
Department of Computer Science, University of Georgia, Athens, GA, United States,
{mepcott@, jam@cs., jun@cs., thorton@}uga.edu*

The ability to effectively specify a simulation optimization can often be impacted by the choice of the language or tool selected. Over the years OptQuest and SimRunner have been examples of the simplification of this effort. There is room for improvement in the connection between simulation models and related optimization problems. By combining the power of the Scala programming language and small Domain-Specific Languages (DSLs) designed to model optimization problems, ScalaTion is extended to enable domain modelers to easily describe their simulation models and optimization objectives.

Modeling & Simulation (M&S) has had a long history of using both General-purpose Programming Languages (GPLs) and Simulation Programming Languages (SPLs). Traditionally, SPLs may be viewed as embedded DSLs (EDSLs), although M&S is broader than many domains of study. Thus, SPLs require many of the features of a GPL, extensive custom language support and longer development cycles. Scala is a GPL suitable for building an EDSL for M&S because it includes the following features: object-oriented, functional, support for Unicode, and adequate performance.

Built on Scala, the ScalaTion project is working to develop an EDSL for M&S. It highlights the five modeling paradigms or world-views: event, activity, process, state and dynamics. A library is provided that eases the creation and execution of simulation models, contains useful functions for numerical statistical analysis, and streamlines language innovation in this domain with Unicode notation. One of its goals is to make the code concise, readable, and in a form familiar to domain experts. In some cases the code looks very similar to textbook formulas. To enhance readability, a version of ScalaTion is provided that heavily utilizes Unicode. Here a few examples of supported statistical functions.

```
def Ⓔ (x: VectorR)           = Σ(x) / x.length           // mean
def σ2 (x: VectorR)         = Ⓔ(x↑2) - Ⓔ(x)↑2           // variance
def σ (x: VectorR)           = σ2(x)↓2                 // std. dev.
def cov (x: VectorR, y: VectorR) = Ⓔ(x*y) - Ⓔ(x) * Ⓔ(y)       // covariance
def ρ (x: VectorR, y: VectorR) = cov(x, y) / (σ(x) * σ(y)) // correlation
```

Simulation Optimization has attracted considerable research interest from the M&S community, however, most of the Simulation Optimization routines are problem-specific and not designed for reuse. Several optimization problems have been modeled by using ScalaTion. First we model these optimization problems using the optimization DSL and then demonstrate how to select an appropriate optimization routine, execute the optimization process, and perform statistical analysis on the output.