

Contribution ID: 508

Type: **Poster Presentation**

Hybrid agent-based library

Monday, 9 July 2018 19:45 (15 minutes)

The Hybrid Agent-based Library (HAL) is a Java Library made of simple, efficient, generic components which can be used to model complex spatial systems. HAL's components can broadly be classified into: on and off lattice agent containers, finite difference diffusion fields, a GUI building system, and additional tools and utilities for computation and collecting data. These components were designed to operate independently, but are standardized to make them easy to interface with one another.

HAL is a useful asset for researchers who wish to build efficient 2D and 3D hybrid models in Java, while not starting entirely from scratch. It is available on GitHub at <https://github.com/torococo/HAL> under the MIT License. HAL requires at least Java 8 or later to run, and the java jdk version 1.8 or later to compile the source code.

The spatial components in HAL all use a consistent indexing scheme. Setting values in a diffusible field, for example, is syntactically similar to setting visualization pixel values, or finding the agents that occupy a position. The consistent syntax helps substantially with HAL's learning curve. It also helps with understanding the source code of models that use components that the reader is not familiar with.

There are several pre-existing general hybrid agent-based modelling frameworks, the most popular being Chaste, Repast, Mason, and Netlogo. Each of these facilitates modelling under a different centralized control structure: In Chaste centralized control done by a Simulator object, in Repast this component is called an Engine, in Mason it is called the Schedule object, and in Netlogo it is called the Go loop.

HAL shares many characteristics with these frameworks, but primarily differentiates itself with its simplicity and decentralized design. There is no controller or scheduler, so the modeler designs the logical flow and the scheduling of interactions between components of the model explicitly. This allows the modeller to define the execution of the model using simple programming constructs such as "for loops", and sheds the difficulty of learning to manipulate a more complex control architecture.

Primary author: BRAVO, Rafael (Integrated Mathematical Oncology Department, Moffitt Cancer Center)

Presenter: BRAVO, Rafael (Integrated Mathematical Oncology Department, Moffitt Cancer Center)

Session Classification: Poster Session

Track Classification: Techniques for Mathematical Biology