

14.2.3 Packages

SEAMCAT has different modules. These are used to group certain classes of the code and handle dependencies between them. This is similar to the layers of abstraction where the engine only knows general system types but not specifics. Modules can be used to enforce such dependencies.

There are two main modules: model and application.

- Model holds all the abstract definitions that composes a System – Transmitter, Receiver, Antenna, Propagation model, Distribution, etc.;
- Application implements the SEAMCAT UI, simulations, Propagation Models, and basically ties everything together. We will discuss this structure in more detail later.

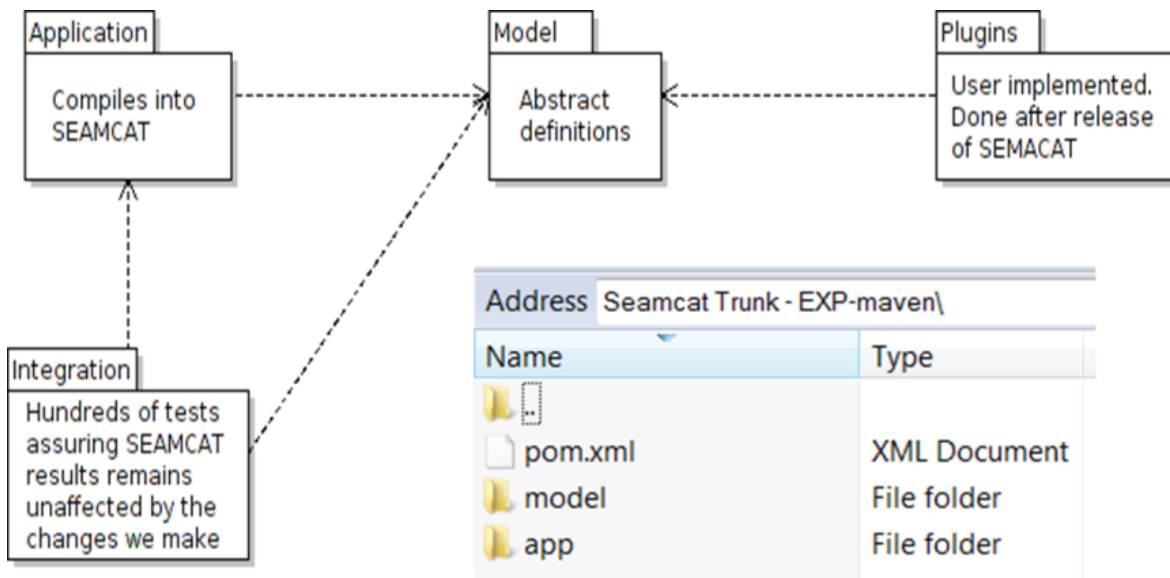


Figure 304: Modules of SEAMCAT

Inside a module we organise the software files in packages. It is similar to a folder and like a folder it contains the individual source code files. The names of the packages is helpful when navigating the source code.

The packages of the model module:

- The names can be helpful when navigating the source code;
- Cellular and generic contains the definitions for the cellular and generic systems;
- Plugin holds the plugin definition for the mentioned types (Eventprocessing, Coverageradius, propagation etc...);
- Distribution, functions, and mathematics holds the distributions and other helper functions;

- Simulation and types holds the simulation results and general type definitions (Antenna, Receiver, RadioSystem, etc).

The packages of Application is more messy than the model. So here it is harder to know exactly where to go. The packages to know here is: scenario and simulation:

- Scenario contains the implementation of some of the types defined in Model;
- Similarly with simulation: here are the specific simulations e.g. CDMADownlinkVictimSimulation.

It is the very nature of code to change, but with a good design and understanding of the abstractions it should be easy to identify part that will change the least. Most of the dynamic parts of the code can be examined with runtime inspection by software development tools. Mechanism vs Policy is a software principle based on the fact that mechanism change infrequently whereas policies change all the time.

Mechanism is here the engine of SEAMCAT and policies are the plugins. The parts that changes behaviour are left open for developers.

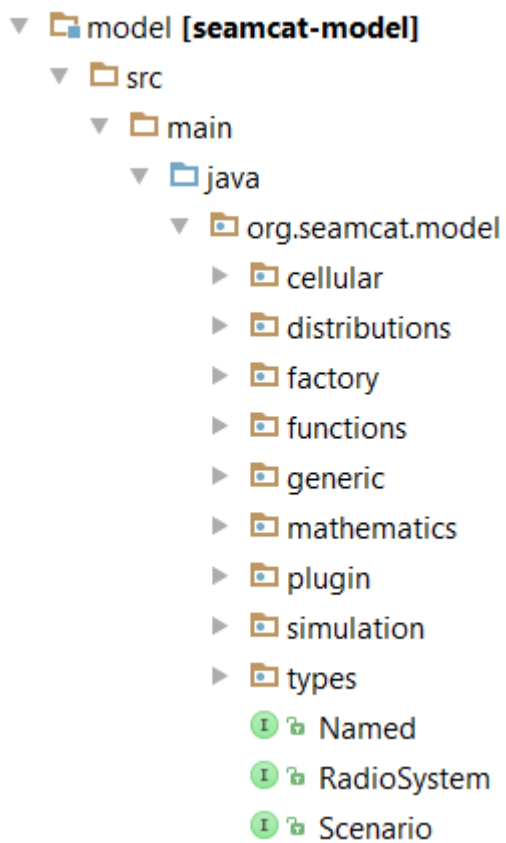


Figure 305: Packages of Model

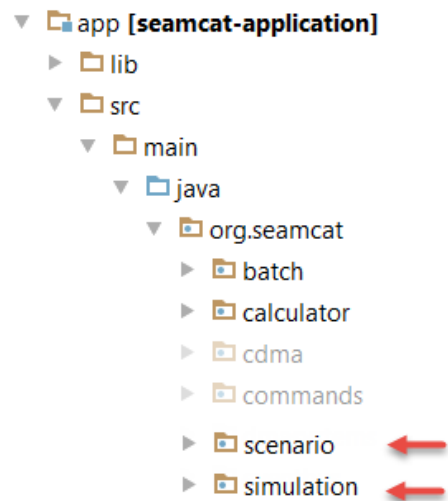


Figure 306: Packages of Application

Revision #1

Created 2026-07-14 06:37:02 UTC by ECO TECH

Updated 2026-07-14 06:38:07 UTC by ECO TECH