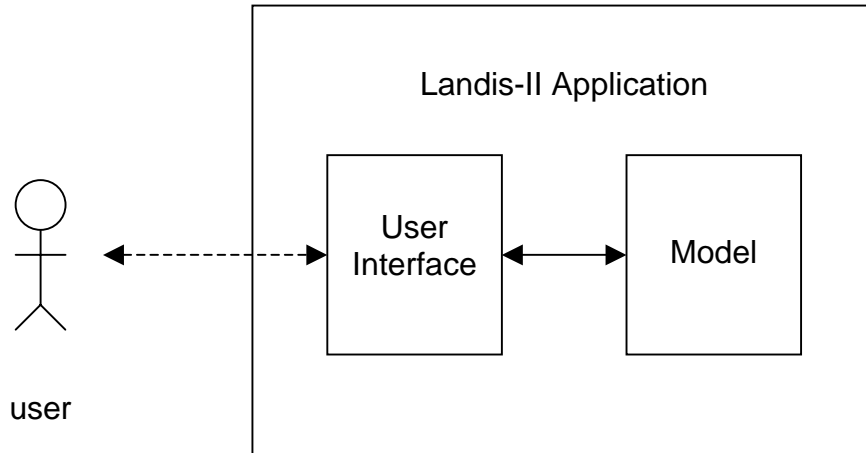


Landis-II – Application Architecture

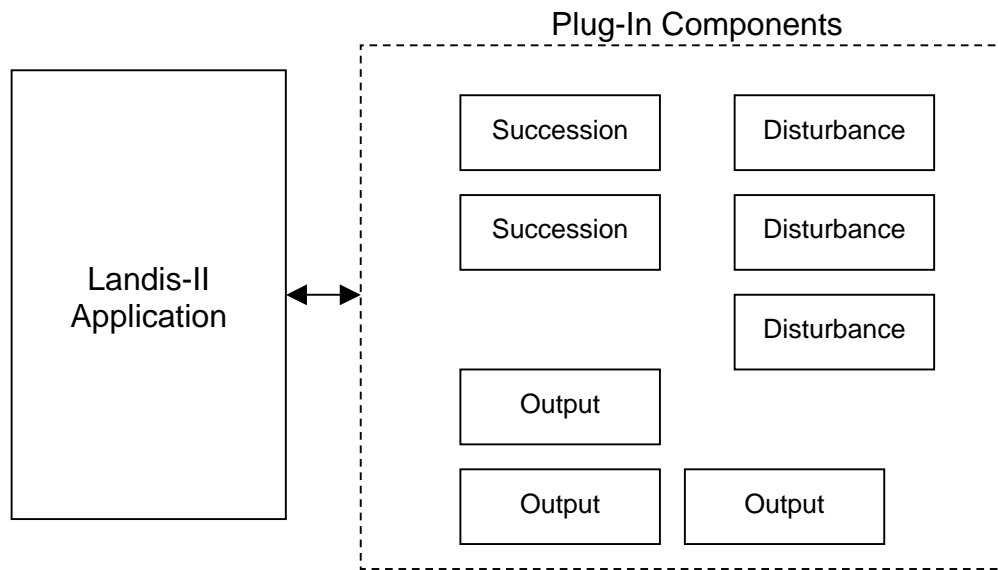
- At the highest level, the application consists of two modules: the user interface (the front end) and the model itself (the back-end).



- There are two forms of the user interface: text (console-based) and graphical.
- The user provides a model scenario as input to the application.

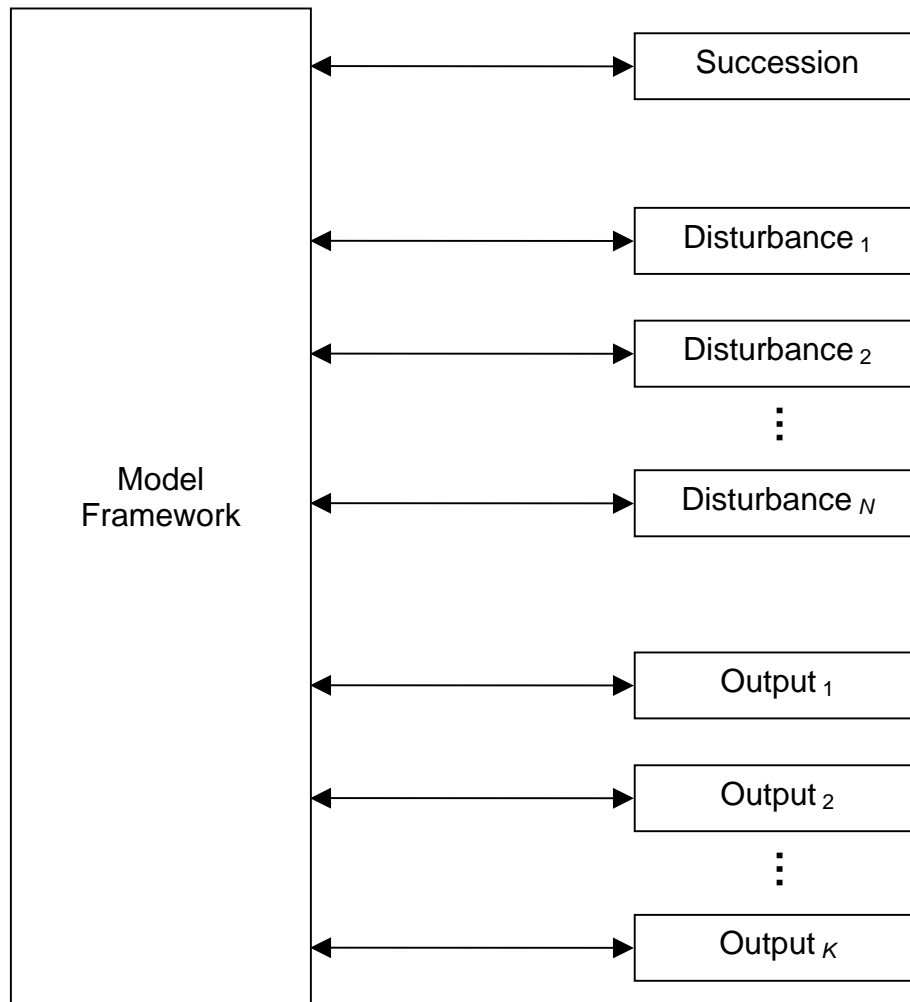
Plug-ins

- A plug-in is an additional component that the user can install to enhance or extend the application's functionality.
- 3rd-parties can develop plug-ins (3rd-party as in not the Landis-II team or the user).
- There are 3 types of plug-in components: succession, disturbance and output.



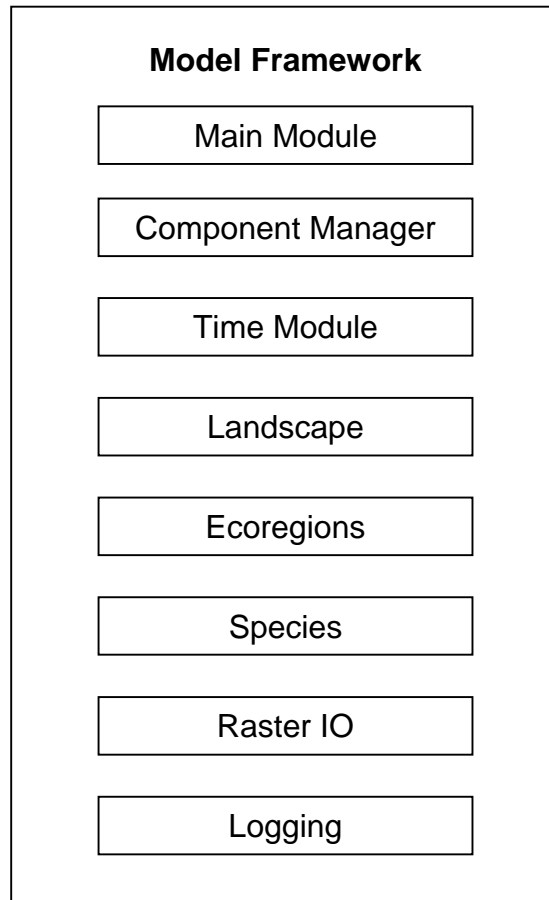
Model Architecture

- The model is a framework which the components plug into.



- A model scenario must specify only one succession component.
- Disturbance components are optional in a model scenario. N is the number of disturbance components used in a scenario; it is $=$ or > 0 .
- A model scenario must specify at least one out component. Additional output components are also optional in a model scenario. K is the number of output components used in a scenario; it is > 0 .

Model Framework



Main Module – initializes the other modules in the framework.

Component Manager – loads (and initializes?) the plug-in components used by a scenario.

Time Module – keeps track of the timestep during a simulation, and ensures that plug-in components execute in the proper order.

Landscape – contains data for sites on the landscape.

Ecoregions – contains parameters for ecoregions

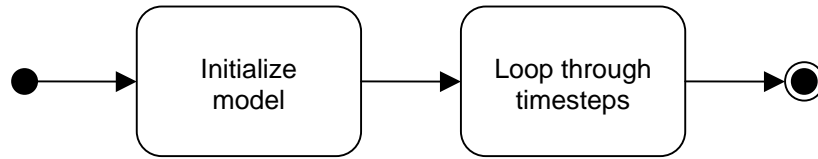
Species – contains parameters for tree species

Raster IO – reads data from and writes data to raster files (images).

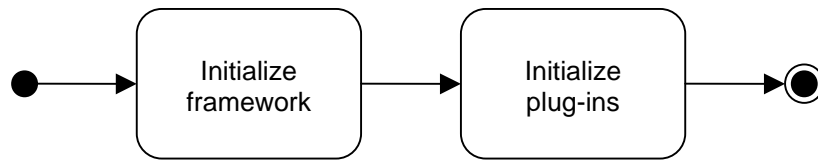
Logging – logs the progress of a simulation and any unusual events that occur.

- A plug-in component may interact with any of the modules in the framework except for the main module.

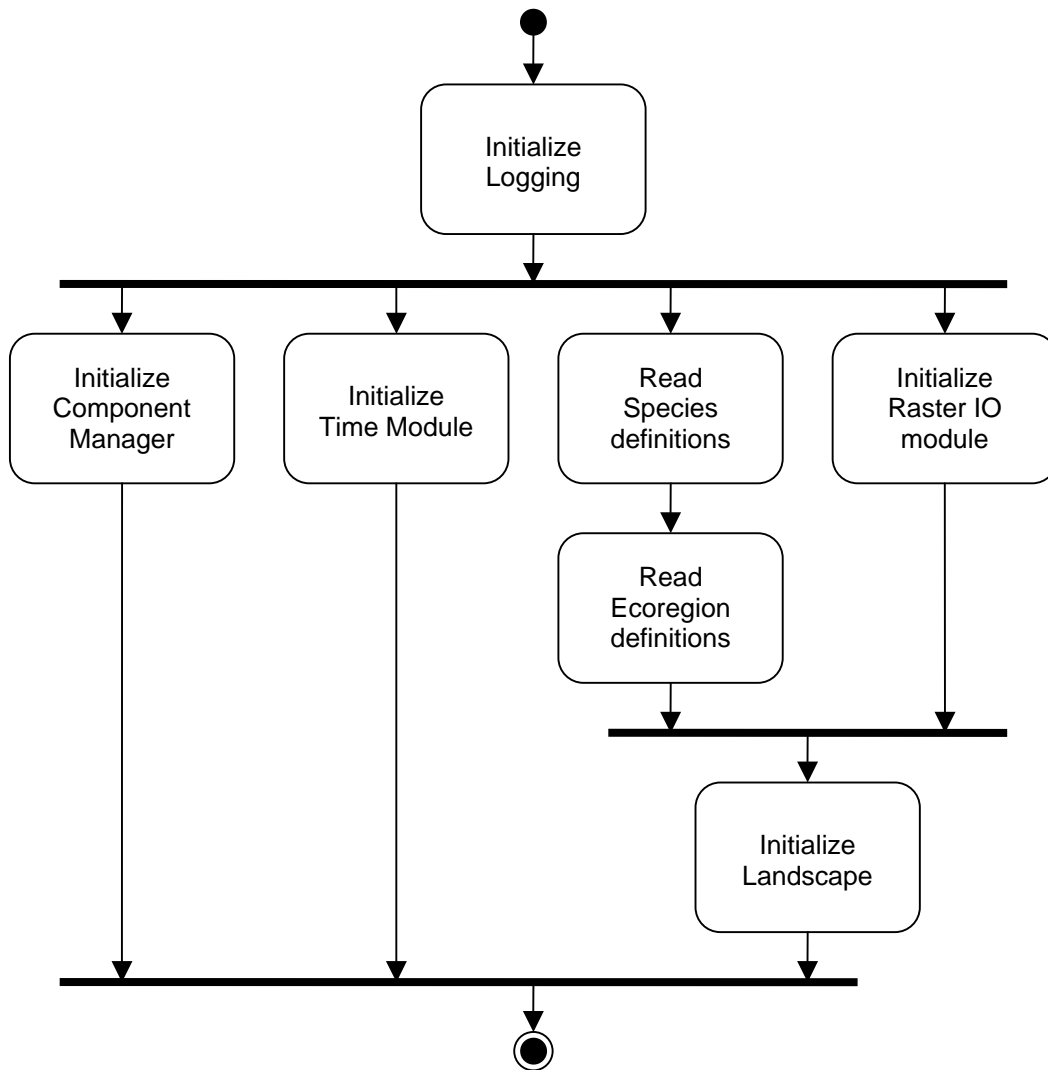
Execute Model Scenario



Initialize Model

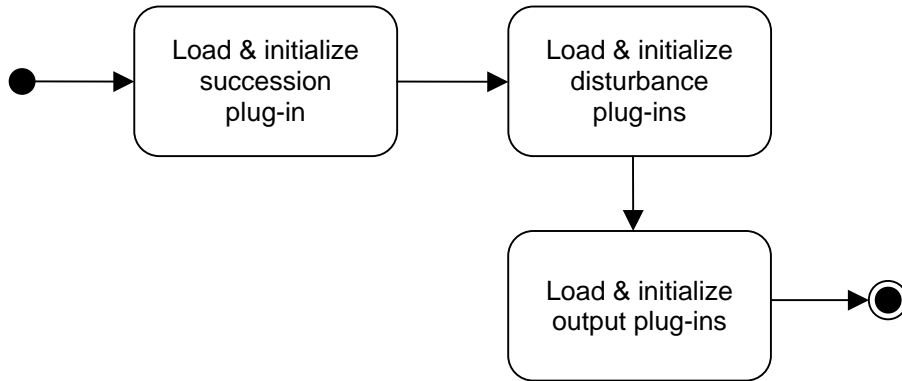


Initialize Framework

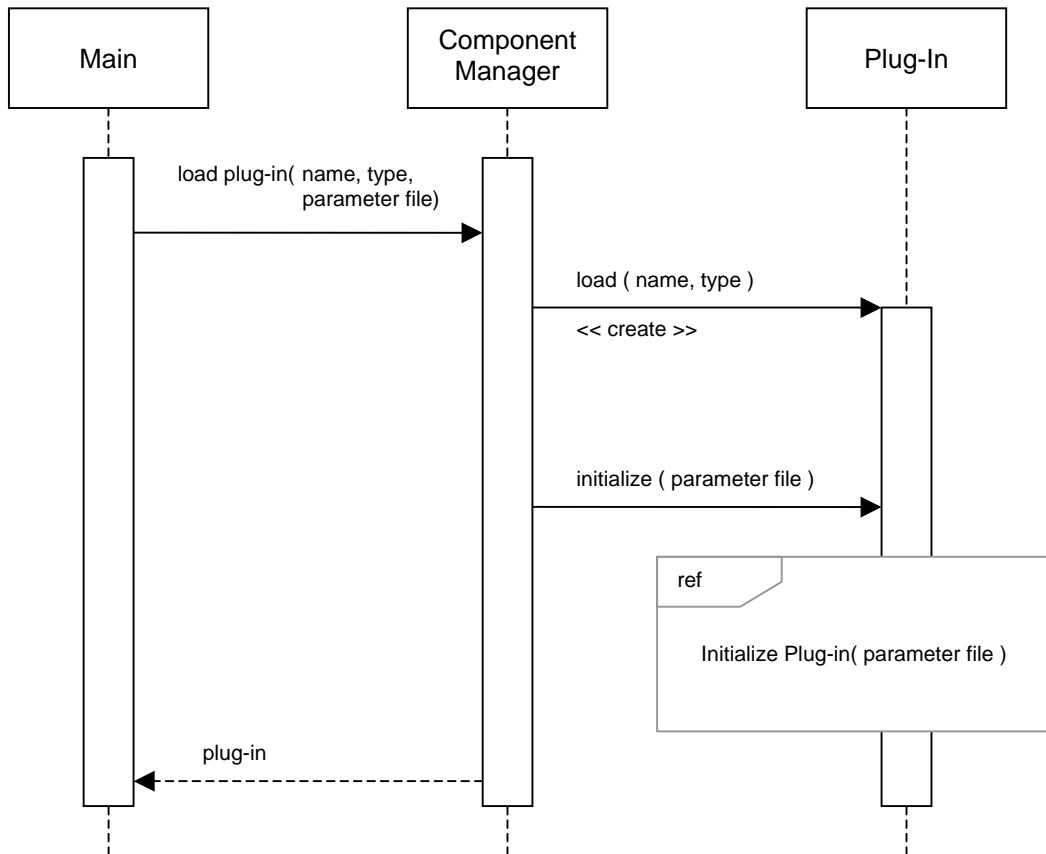


- The Main module directs this initialization process.
- Logging module is initialized first so that the initialization of the other modules can be logged.
- Species definitions are read before ecoregion definitions because ecoregions have species-specific parameters.
- Landscape initialization has 2 phases: 1) creating the internal data structure for active sites, and 2) creating and initializing the site variable “ecoregion”. Both phases require the ecoregions input map. Reading this map requires the ecoregions definitions and the raster IO module.

Initialize Plug-ins



Load and Initialize Plug-In



Initialize Plug-in

