**UPGM/AgES-W Integration Overview**

The original UPGM/AgES-W integration compiled a modified version of UPGM into a single dynamic link library (i.e., a .dll file which is a collection of compiled FORTRAN files) that could be run daily for UPGM. However, since AgES-W has multiple HRUs in a run, global variables in the library would be shared by all HRUs (which would produce incorrect results). There were two ideas on how to solve this problem. Initially, Robert and Olaf worked on a version where an executable (.exe) file was built instead of a .dll library. This executable would be run daily, and each HRU that ran the executable would have its own memory space. This version of UPGM and AgES-W corrected errors that existed from multiple HRUs using the same memory; however, it introduced another problem. For each day the executable was run, information from the previous day was lost. There are many variables in UPGM which require information from the previous day or are initialized on the first day of simulation. Therefore, the AgES-W team elected to use a second approach to the original problem. This approach involved making copies of the original .dll library and each HRU would have its own individual library copy. Therefore, the library could be loaded into memory, persist through each day, and each HRU would have its own memory space. Robert and Olaf worked on this approach as well.

Unfortunately, the “individual library” approach also was not without problems. For example, if there are a large amount of HRUs the memory space occupied by the library is “untested” and may be a bottleneck. Another problem was there were so many arguments being passed to UPGM that the maximum number of arguments (to pass back to the AgES-W Java code) was reached. A third issue was a result of modifications to the original UPGM source code, i.e., there were so many changes to the UPGM standalone version that it was difficult to determine which changes from the standalone needed to be integrated back into AgES-W. Therefore, it was difficult to finalize a basic set of connection and communication protocols between AgES-W and UPGM. In order to try and solve these problems, a new version of the UPGM integration was built starting in April-May, 2015. The new version did not require UPGM source code modifications and thus permitted easy comparison and updating of the standalone and AgES-W versions of UPGM. Note that the reason that this was not originally done was because the goal was to have a single library for all HRUs (where we would attempt to save and load the entire library variable state for each HRU and switch between them during run-time). However, since we determined that a copy of the .dll library was needed for each HRU, there is no reason to save and load the entire variable state every day. This resulted in a significant reduction in the number of parameters that are passed between AgES-W and UPGM (and subsequently stored in the .dll library). The new (current) approach created two “wrapper” modules (written in FORTRAN) that are specific to the AgES-W version of UPGM. The first wrapper module is called JupgmInit.f90; this module is executed at the beginning of every crop planting and essentially is responsible for initializing crop parameters. The second wrapper module is called Jupgm.f90. Jupgm.f90 is executed every day between planting and harvesting and performs the core UPGM plant growth calculations. This module is also responsible for passing output from UPGM back to AgES-W.

Once the final connection and communication protocols were built for AgES-W and UPGM and the wrapper modules were in place, the final step in the integration process was to link UPGM to the rest of the AgES-W modules via the UPGM input/output (I/O) variables. After carefully analyzing the PotentialCropGrowth.java component (from the SWAT plant growth model implementation), a final list of I/O variables for UPGM was created. The output variables are the ones in the I/O list that UPGM is passing back to AgES-W. UPGM-required input variables are read in from a new UPGM-specific crop database. This database has undergone multiple updates and now provides all the required input to UPGM.