**Specification of the Subject of Performance**

**Name of the service:** Development of a program for the simulations of impacts of various biotic disturbance agents on the forest with interface to an existing ecosystem model

**Requested activities – description, quantification (e.g. number of samples, forest area), time schedule, etc.)**

Development of specific program for the simulation of the impacts of a broad range of biotic disturbances on the forest based on the most recent scientific understanding of ecology of these agents. The biotic disturbances will include fungi, insects (wood and bark borers and defoliators), and small and large mammals (i.e. five agents at minimum). The program will include an interface that will allow for its full integration with an existing ecosystem model.

The program will be coupled with the forest landscape and disturbance model iLand (<http://iland.boku.ac.at/>). The model iLand is freely available and licenced under the GPL open source licence. iLand is written in C++ and relies on the open source toolkit Qt (<http://qt.io>). Furthermore, iLand uses a JavaScript engine internally, that allows script-based access to many internal functions of the model. For example, forest management is defined as JavaScript code which gives the model user a high degree of control over details of the management implementation.

The integration of the new program and iLand should allow biotic agents to affect individual trees in the ecosystem model (e.g., kill or defoliate trees) and to retrieve information at a specific location about the current state of the vegetation (e.g., availability of host species, tree dimensions) or other environmental conditions (e.g., climate data). Achieving such an integration without sacrificing computational performace requires a tight technical integration of the new model into iLand. The development of the new software fully integrated and compatible with C++/Qt is therefore mandatory and advanced knowledge about the technical foundations of iLand is requried.

Moreover, the biotic disturbance program needs to be highly flexible with regard to the definition of the behavior of the biotic disturbance agents. Ideally, this flexibility is achieved by facilitating a scripting-based approach that leverages the built-in JavaScript engine of iLand.

The software will need to be designed with the following key characteristics:

* Processes: The biotic disturbance program needs to include the processes of introduction, dispersal, colonization, population dynamics, and impact (note that not all processes are applicable for each agent)
* Spatial resolution: the biotic disturbance model runs on a grid with a variable spatial resolution that should range from 10m to km.
* Temporal resolution: The principal resolution should be annual, although some processes (e.g., multivoltinism in bark beetles) require higher resolution
* Access to vegetation state: the model needs to collect data on the state of the current vegetation on individual cells. This information include species, number of trees, tree dimensions, available biomass for target pools (foliage, stem, roots), and information about the regeneration layer
* Modification of trees: the model needs to affect trees in the ecosystem models (impact of biotic disturbances) by either removing biomass from target biomass pools (e.g. defoliation) or by killing trees. For biomass removals (foliage, roots) a biological response of the affected trees (e.g., reduced tree growth) need to be designed and implemented
* Outputs: the new model needs to provide grid-based and aggregated outputs for all agent processes
* JavaScript API or equivalent solution: The software should provide an API (or equivalent solution) that is used for the script-based definition of the agent behaviour. The model should provide the basic functions (e.g., for data retrieval and the impact on trees) and should allow further customization. For example, the user should be able to implement alternative types of agent impact, or to change agent specific distribution kernels on the level of JavaScript or equivalent solution.

Source code and detailed technical documentation of the program and its interface, and description of the scientific foundations of the implementation (e.g. parameters for individual biotic agents, source literature, etc.) will be supplied.

**Required methodology:**

As described above, the implementation should be C++/Qt and should provide a scripting API for increased flexibility in various scientific applications. The simulated data will be stored in a database.