Silva Fennica vol. 50 no. 1 article id 1393, availabe at http://dx.doi.org/10.14214/sf.1393

Supplementary file

Technical Basics

Name: Wood Quality Toolbox
Developer: Chair of Forest Growth and Yield Science
Contact Address: Hans-Carl-von-Carlowitz-Platz 2, D-85354 Freising, Germany
Year first available: 2015
Software required: R statistical environment (R Core Team, 2013).
Software recommended: RStudio.
Availability: Upon request, introductory support recommended.
Program size: 400 KB.

As explained in the referencing publication, the simulation model covers the major relevant linkages reaching from stand management and site influence via individual tree competition and growth to crown shape and branch dimension and further down to knottiness, wood density and strength of the individual board.

During simulation, the developing stand is a container of variable tree objects. The stand state of each finalised year is stored as an age related collection of invariable quasi frozen tree objects to constitute a growth history. Thinning and mortality remove trees from the current year container and store them into a global container of all felled trees. All variable data of tree dimension and structure are held within the tree object and further objects inside that represent nested levels of tree architecture, such as stem, crown, whorl, log, branch, and board. Currently simulation is organised into (1) growth, thinning, pruning, (2) construction of log objects and (3) construction of inner log and board structure. Each tree object is passed through a chain of computation objects starting with growth and thinning and ending with sawing of logs and boards. To the greatest extent the tree object is modified, extended and diversified by functionality of the modules that handle it, whereas its own interface is kept lean and general. The state of each computation object remains constant once initialized with parameter values, while any state change is stored to the passing tree objects. As each module represents an elementary and common task of simulation, such as competition intensity or potential growth, the functionality of the model is readily modified by adding or replacement of plug-in computation objects. The system also includes a module for strength grading in accordance to European standards EN 338, EN 14081 that had not been applied within the scope of the referencing study.

Fig. AP 1 to Fig. AP 5 illustrate the simulation process using the implemented perception of stand, individual tree, log and board. Prediction starts with the stand structure at setup inventory. Reconstruction is based on the stand state at setup inventory and a reported planting pattern (Fig. AP 1).

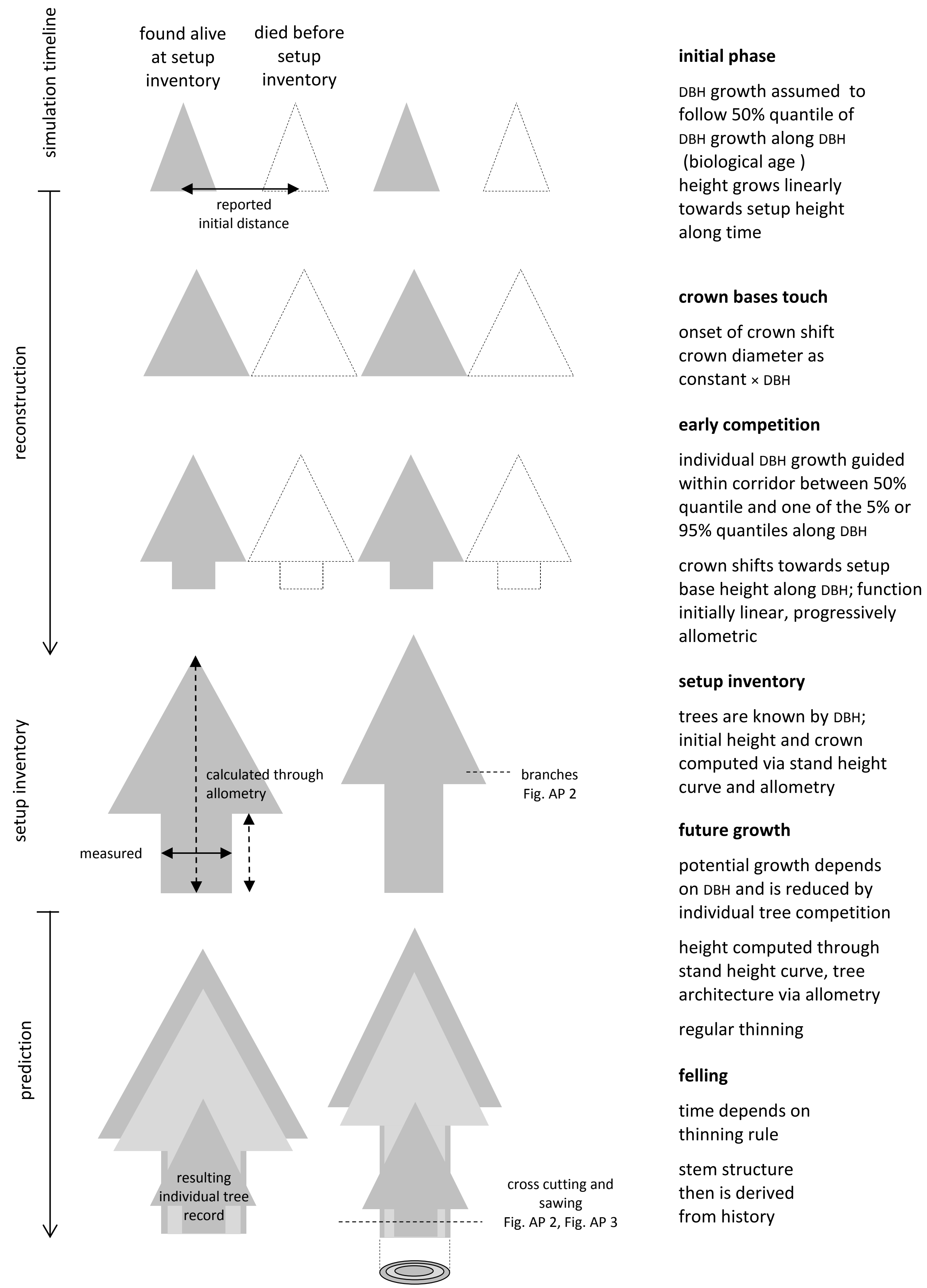

Fig. AP 1. Principles of annual growth reconstruction and prediction.

Branch size is constructed from the hull of the crown. The resulting branch length is translated into branch diameter based on data from field measurement. During growth simulation, branch properties are stored within whorl objects. At that stage there is no requirement for a branch object of explicit inner geometry: Branch objects are constructed as needed from the growth record of the individual tree in a separate step (Fig. AP 2).

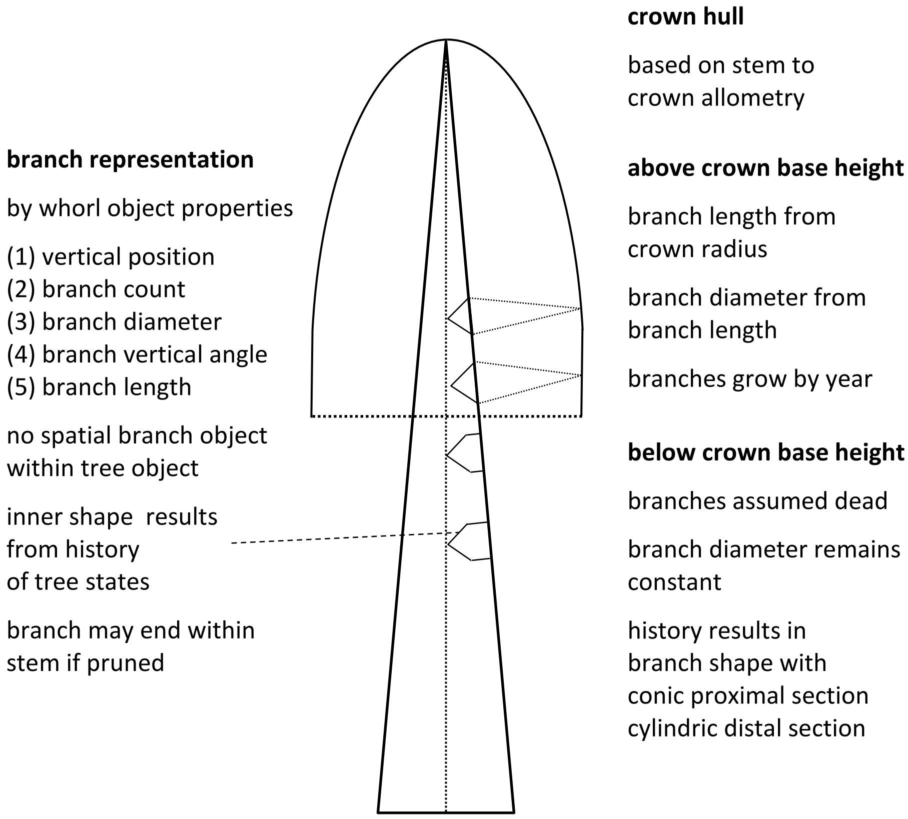

Fig. AP 2. Principle of branch growth reconstruction and prediction.

After reconstruction and prediction, each individual tree object is supplemented with a collection of newly constructed log objects based on the tree history: Log structure is defined by the annual ring radii at median vertical position within the log and the branch properties registered in any whorl object that is enclosed by the vertical log limits (Fig. AP 3). Based on the log objects and the cross sectional sawing pattern that applies to the log radius at top position, the structure of each individual board is computed as explained by Fig. AP 4.

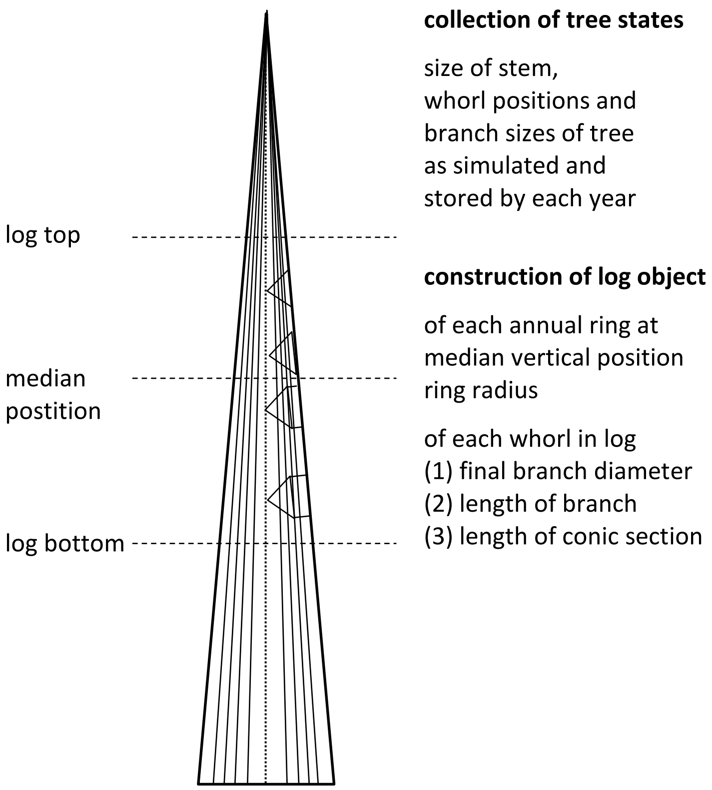

Fig. AP 3. Principle of tree history to log structure translation.

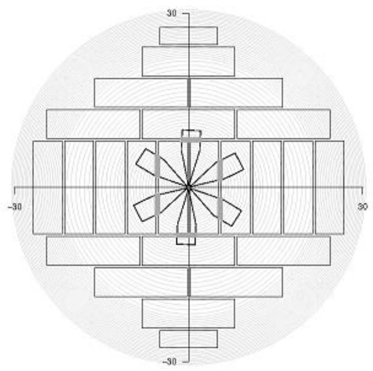

Fig. AP 4. Principle of log to board structure translation.

Board strength is calculated by individual board, based on stored tree objects with associated log objects and board collections. The process may be run at a different time and with any algorithm selected for computation of strength.

Fig. AP 5 finally illustrates the plugin architecture using a snippet of growth computation (whorl and branch geometry): Each current year tree of class Tree is handled by an object of class TreeHandBranchGeometry that is a TreeHandler. Any TreeHandler inherits a list of Growth objects and its function *runallon* applies method *runon* of each list member to the tree that is processed (diagram left top). TreeHandBranchGeometry holds two specialisations of Growth in its list: WhorlDiameter for computation of whorl diameter and BranchGeometry for branch size and vertical angle. Each Tree object is passed to all TreeHandler objects. TreeHandler objects exists for all main tasks of the system, such as growth, stem structure and thinning. Growth algorithms may be readily modified by creating new Growth objects through extension of the base class and inserting them into the appropriate tree handler.

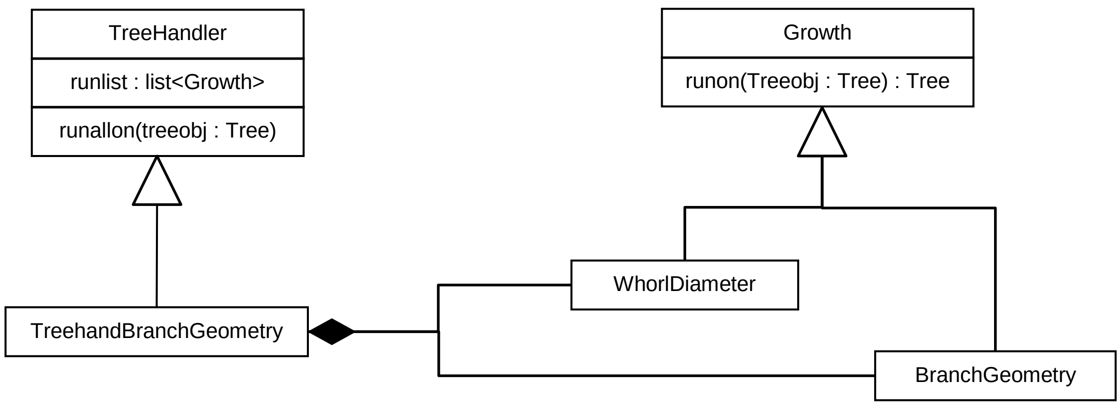

Fig. AP 5. Principle of the plugin architecture as a class diagram (UML).