

Software Development for Forest Growth Models and Management: CORKFITS Web-Based Growth Simulator

Surovy, P., Vones, P. & Ribeiro, N.A.

Keywords: Forest dynamics, growth models, simulation software

Abstract: New methods of forest management and the study of their impact on sustainability are strongly dependent on realistic mathematical modelling. The complexity of the models however, makes the use of computational power, and thus the incorporation of knowledge from computer science and research, indispensable. In this paper we wish to demonstrate the development of a simulator for the growth and production of cork oak woodlands – montados. The software is divided into three sub-modules, sharing a common core, with functions and mathematical operations. The desktop client allows for repeated operations for more intense calculations, and statistical operations for modelling purposes. The web version is intended to be used by final users in forest practice. It permits simulation of inventory data based on individual tree measurements, and inventory data based on plot description with a reduced amount of detail (number of trees per ha, diameter structure, etc.) The last module allows the incorporation of the cork model into other software by means of SOAP protocol, via web services. It conforms to the WS-I Basic Profile 1.1, to ensure interoperability among the largest number of clients. This module allows other developers to use the cork oak growth-model in their software, and the developers from other areas of expertise (management optimisation, decision support...) have the opportunity to test their techniques on real stands, with the most recently-updated model versions.

Received August 6, 2010; Accepted November 30, 2010

1. Introduction

The management of forest stands has for a long time been considered adequately handled, by simulation techniques only (Mitchell, 1975). The simulation models used for estimation of growth and production can be divided into several levels: from cell level through organs, individual trees, and up to the stand or landscape level (Pretzsch, 2009). Every level of models however, usually attempts to take into account so many input parameters, and produce so many outputs, that using the model without computer support is nearly impossible.

The need for computers has been recognised for a long time. Modern software tools began to be introduced into forestry management, mostly in the United States, but in the last decade, also in Europe (Pretzsch *et al.*, 2002a). Wykoff *et al.* (1982) describe the model PROGNOSIS; its structure and behaviour, options, input requirements, interpretation of output and applications for planning. Wensel and Koehler (1985) describe software CACTOS, a simulator based on an individual tree growth-model for Northern Californian coniferous forests. The software uses the individual tree distance-independent model and is supposed to handle the forest inventory data without the need to measure tree coordinates. It was noted in this paper that the publication of the software helps to evaluate model accuracy by using independent data sets, which clarifies the necessity of making the software accessible to public usage on the one hand, but on the other, also points out the necessity for systematic updates of the software with regard to the model. Jaegger and de Reffye (1992) proposed simulation software (AMAP) for architectural (functional – structural) models of trees. The simulation of growth is divided into two steps: 1 - topological part where bud activity was simulated and 2 - geometric part where the spatial characteristic of this activity was defined.

Recently, a significant increase in individual tree models was noted,

together with their software implementation. Among these are Pretzsch *et al.* (2002a) – SILVA, Sterba and Moser (1995) – Prognaus, Nagel and Schmidt (2003) – BWINPro and Fabrika (2005) – SIBYLA. Many others can be found on various registers and catalogues available on the internet (<http://models.etiennethomassen.com/>; [http://www.iefc.net/\(Modeles\)](http://www.iefc.net/(Modeles)); <http://formis.emu.ee/formod/>)

Pretzsch *et al.* (2002b) described several rules for the development of software for forest-growth simulation. Four principal questions should be addressed when developing software.

1 – Ease of use of the software. The conventional GUI (Graphic User Interface) is these days a common standard for software interface. Text command interface also exists (R – statistical software) due to the very low resource requirements, but is generally considered more suitable for specialists than for the general public. Usually the command line software is vulnerable to typos (typing errors during typing the commands), while GUI eliminates this danger. On the other hand, many modern applications are accessible through graphical web interface, which leads to the second question from the article.

2 – Flexibility of use. Flexibility is explained by the authors as the use of programming language that is platform-neutral. It is important to distinguish between the programming language and the programme (software) itself. In fact, most of the programming languages are themselves platform-independent, but when the software starts to use GUI, it calls up routines from the operating system and in this way it becomes platform-dependent (whatever this platform may be: operating system, virtual machine (runtime environment)). Therefore it would be more accurate to refer to flexibility as to software which is largely platform-neutral. This requirement is once again relatively easily fulfilled (as already mentioned in question 1), by web application. The GUI of internet web page allows maintaining the interface as ‘easy to

use' and typo-resistant, while keeping it fully platform-independent. Compared with the use of virtual machines (or runtime environments - like JAVA, .NET), web application doesn't even require the installation of this machine on the computer. The only requirement is a web browser, which is a common component of most of the operating systems.

The third question to be addressed by development of the software, according to Pretzsch *et al.* (2002b) is: 3 - The possibility to integrate the growth simulation into information flow in forestry practices. This can be fulfilled by the correct interfaces for user-input data and defined data formats for outputs. The authors define two possible uses in forest practices: interactive use for professional training and scenario calculations, and batch-mode via external files control for large-scale scenario calculations. In the latter part it is also important to notice that batch-style processing permits optimisation of management and the search for optimal management techniques. In the case of web applications, these can be accessed via the web service protocol (<http://www.ws-i.org/Profiles/BasicProfile-1.1.html>). This means that the core application is stored on the server, from where it can be accessed, or by users via the internet page, or by other software via the web service. The commands files are supposed to define the processing steps and data sets can also be sent this way. This approach allows not only accessing of the models by other software to communicate the results, but also opens up new possibilities for optimisation purposes by other developers.

The last recommendation for software development from Pretzsch *et al.* (2002b) is: 4 - The documentation which is regularly updated in the help files in desktop and also in the web version.

2. Mathematical model

The CORKFITS software is based on the distance-dependent individual tree growth-model (Ribeiro *et al.*, 2006). The potential growth in diameter and height is calculated based on the stand quality (soil). The potential production is derived from stem circumference and debark height, defined by the user, or derived from legislation (3x the circumference at 1.3 metre height). The potential production is later decreased by competition status derived from the competition index proposed by Hegyi (Hegyi, 1974).

3. Software architecture

The structure of the software and its architectural composition is displayed in Figure 1. The main functions are stored in core libraries which are later linked, as needed, to the individual applications. Other applications link to the core libraries and use those links which are needed for the particular module.

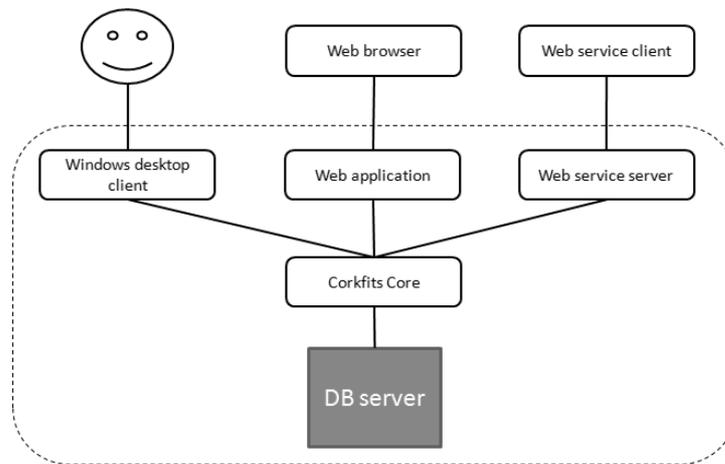


Figure 1. Structure and architectural composition of CORKFITS software

3.1 Corkfits core

This basic module contains all shared and common functionalities of the simulator and consists of three parts: A - object model and database access, B - algorithms, C - visualisation. Module A contains functionalities for database access, configuration of communication and utilities, objects for individual trees and stand, and objects for database responses. Module B – algorithms, contains calculation procedures, the individual tree model functions, competition functions and all the mathematical functions necessary for the simulations of individual tree growth. It also contains the stand-summary functions; especially the average function for cork thickness, stem structure (amount in individual thickness classes), the quality index of cork, etc.

Module C – visualisation, handles some additional functionality connected to visualisation of the results. In the cork oak woodlands one of the important management objectives is the crown cover (Ribeiro *et al.*, 2004). Too-high crown cover results in a limited amount of light in the understorey and so inhibits the pasture production. Too-low crown cover increases the risk of erosion. The crown cover itself is not produced as an output from individual tree growth-models and it is difficult to calculate the crown overlaps when using the ellipsoids, as is the case in the model used in CORKFITS. The procedure for calculating the crown cover is conducted using bitmap functionalities, where at first all the crowns are drawn in horizontal projection with green pixels. Then the green pixels are counted and compared with the total stand-area. Subsequently the two-dimensional crown map is plotted using only line graphics, as shown in Figure 3 (right part).

3.2 Windows desktop client

The flow chart from desktop client is shown in Figure 2. The user, in the beginning, defines the stand with which he is going to work.

This stand can be measured together with horizontal structure and all necessary variables. If data are missing (or only the simplified inventory method was used), before continuing it is then necessary to generate the missing parameters (called generator functionality from Core).

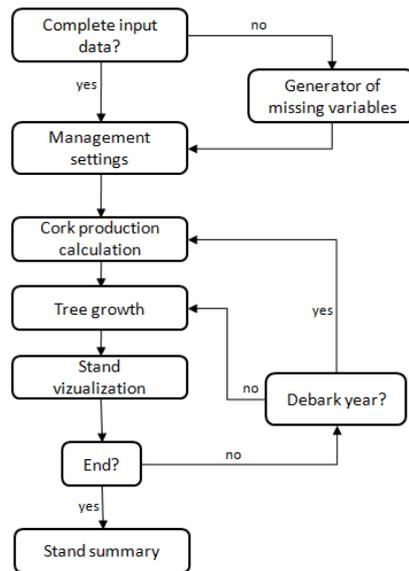


Figure 2. Flow chart from CORKFITS – desktop client

After the introduction of input data the management of the stand has to be defined. This includes settings for bush removal (disking/cutting), pasture (cattle/sheep/none) and economics of debarking (productivity of work, payments, regeneration costs and worker payments). All of these variables are stored and later used when calculating stand economy (costs and profits). After these settings, the user is asked to define simulation interval and simulation is started. In the first year the trees are debarked. Then the increment in height and diameter is calculated, and subsequently, crown expansion and mortality. The simulation con-

tinues until the final year is reached.

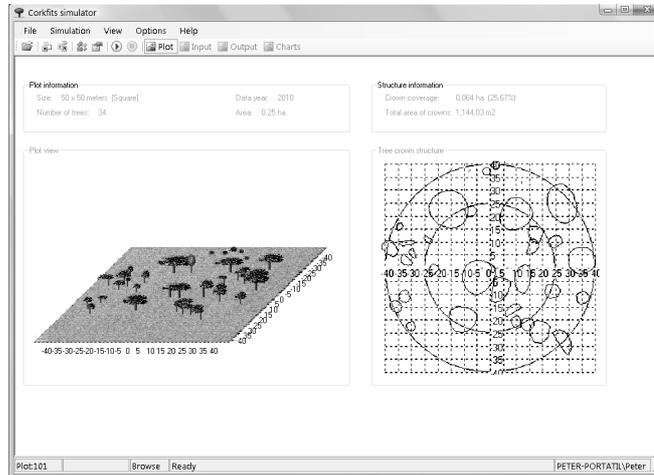


Figure 3. Screenshot from CORKFITS - desktop client

3.3 Web application

The flow chart of web application architecture is shown in Figure 4. The difference between desktop application and web application is that there cannot be any ‘window’ use, only web forms, and the variables must be transferred from one form to another, either by HTTP messaging, or by storing them in the database. The forms are displayed to the user sequentially. In the first one the user is asked to introduce data for the plot and the missing variables are generated. Next, management settings are processed and finally the simulation settings (with equal information as in desktop client). At the end the settings are in the database and the user receives a code (ticket) with which the particular simulation can be extracted from the database at a later stage. The screenshot from web application can be seen in Figure 5.

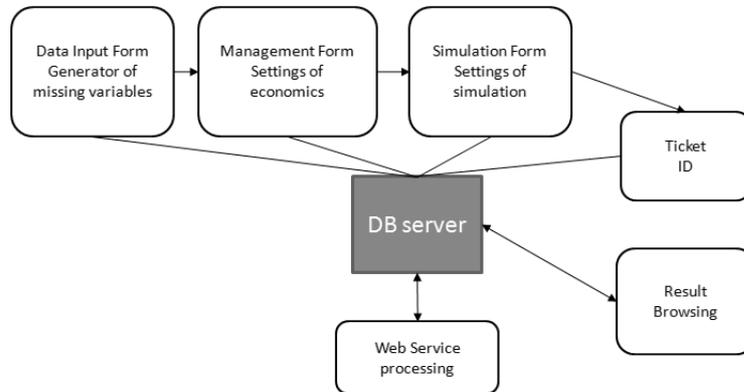


Figure 4. Flow chart from CORKFITS – web application

Input the number of trees per hectare:

class1 (circumference 30 - 70 cm) :

class2 (circumference 70 - 100 cm) :

class3 (circumference 100 - 130 cm) :

class4 (circumference 130 - 160 cm) :

class5 (circumference 160 - 180 cm) :

class6 (circumference 180 ++ cm) :

Regularity of structure :

cover : 26.15%

Figure 5. Screenshot from initial form of CORKFITS–web application

3.4 Web service server and client

The flow chart of web service application is displayed in Figure 6. The simulation is started in the same way as in the case of web application, by introduction of initial data. The difference is that there is no Graphical User Interface, so every command must be sent by the client software.

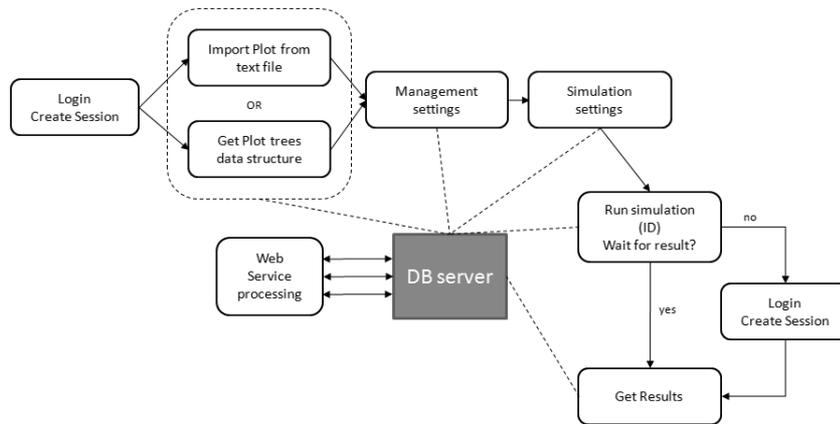


Figure 6. Flow chart from CORKFITS– web service

The simulation results are stored in a database and can be later retrieved as in the case of web application, using ID of the simulation, but for optimisation purposes there is an option to wait for the result, in order to be able to set new management settings only, based on the present result. Figure 7 shows an example of a client capable of connecting to this service. This client was written in .NET but any programming environment and language (running on any OS) can be used to connect to the web service.

The only requirement for the programming environment is that it supports the WSDL (<http://www.w3.org/TR/wsdl>). The list of commands necessary for communication is imported when the request is

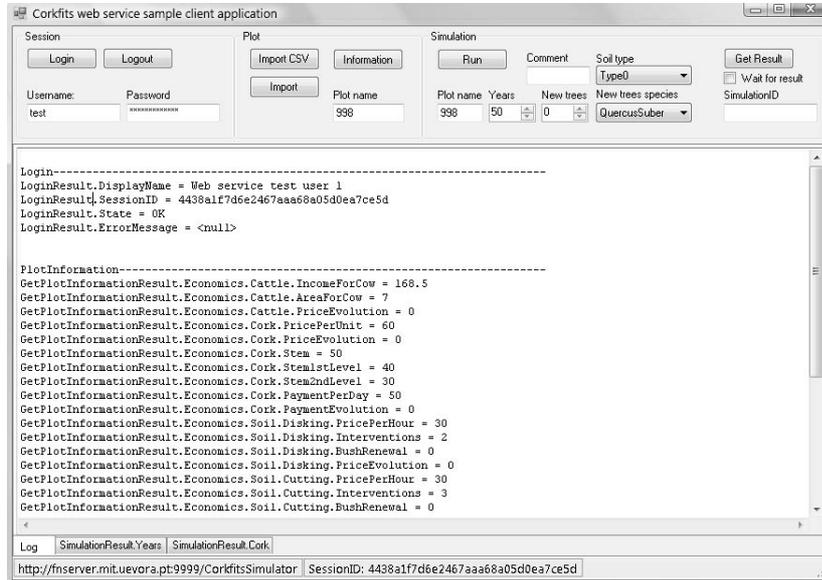


Figure 7. Screenshot from CORKFITS example–web service client

sent via HTTP to the web service.

4. Discussion and future work

In this article we present the software for simulation of the cork oak woodlands. The software is based on .NET technology, allowing several functionalities to be joined in central assemblies and later incorporated into different levels and applications. There are, at the moment, three main applications which have been developed: desktop application, web application and web service.

The desktop application is intended to be used mostly by model developers and professionals with higher levels of knowledge of the problems of tree models and production estimation. The main advantages of this solution are: speed, richness of the options for data input and

interactivity (the possibility to return certain numbers of years, change management options and run alternative simulation). The disadvantages are: the desktop version (if used by more users) has to be regularly updated because of the updates in software and database structure and the necessity to have local installation of run-time environment (.NET) and database servers.

The web application is mostly focused on forest practitioners and researchers to assess the evolution and production of cork oak woodlands in order to visualise the development of their stands or test management scenarios and visualise the impact of management decisions. The principal advantage of this solution is that the application logic is stored centrally in a server, and any update to the software logic or database structure is immediately available to all the users. Another very important advantage is that in this way (as suggested in Pretzsch *et al.*, 2002b) the application itself is completely platform-independent and it can be run on any computer system, or accessed by PDA and other devices capable of HTTP communication, without any additional component-installation. The disadvantage is that the communication has to be optimised for low data-flow, and minimisation of graphic outputs is required to improve the speed of form loading.

The web service is designed for advanced computer users who want to incorporate the cork model into their existing software, or for users who want to use the model for optimisation purposes. During optimisation the model has to be recalled many times, so that every time the driving variable is changed, outputs are evaluated or saved. Well-designed web service technology allows the service to receive sequential command files, process them and return the result in a set of results, or individually, in real time. The communication doesn't usually require any graphic exchange, so it is relatively light in data transfer. The main advantage of this solution is that the developer of the opti-

mising algorithm can, at any time, access the most recent version of the software whenever something changes in it. The optimisation can be run without the need to reinstall the components, and the same is valid for the data which are stored in the main server and can be accessed at any time when new changes are made. A second important advantage is that this way the software is available to the software developers in any possible environment for programming, and in any programming language capable of working with WSDL (Web Services Description Language). The disadvantage is that the communication is dependent on existing internet connection and despite the fact that it can be restarted at any time, when the connection goes down, in long-term calculations, the instability in internet connection may cause some additional delays.

In future, however, with the ever-increasing quality and technological levels of computer research and technology, we believe that the internet approach for application development will become attractive, not only for developers and specialists, but also for the broader public.

Acknowledgement

The authors would like to thank Tohoku University Ecosystem Adaptability of Global COE and the Institute of Statistical Mathematics for the financial support to attend and present this research at the international symposium FORMATH Tachikawa in March, 2010. The first author would like to acknowledge FCT (Fundação de Ciência e Tecnologia) for funding the research, through grant SFRH / BPD / 29953 / 2006.

References

- Fabrika, M. (2005) *Simulátor biodynamiky lesa SIBYLA, koncepcia, konštrukcia a programové riešenie*, Habilitačná práca, Technická

univerzita vo Zvolene, 238

- Hegy, F. (1974) A simulation model for managing jack-pine stands, In: Fries, J. [Ed.] *Growth Models for Tree and Stand Simulation*, Royal College of Forestry, Stockholm, Sweden, pp 74–90.
- Jaeger, M. and de Reffye, Ph. (1992) Basic concepts of computer simulation of plant growth, *J. Biosci.*, 17(3): 275–291.
- Mitchell, K.J. (1975) Dynamics and simulated yield of Douglas-fir, *For. Sci. Monogr.* 17: 39.
- Nagel, J. and Schmidt, M. (2003) The single tree model BWINPro, a key component for silvicultural decision support, In: Vacik H. *et al.* [Eds.] *Decision support for multiple purpose forestry, A transdisciplinary conference on the development and application of decision support tools for forest management*, April 23–25, 2003, University of Natural Resources and Applied Life Sciences, Vienna, Austria, CD-Rom Proceedings, p 1–9.
- Pretzsch, H., Biber, P. and ursky, J. (2002a) The single tree-based stand simulator SILVA: construction, application and evaluation, *Forest Ecol. Manag.* 162: 3–21.
- Pretzsch, H., Biber, P., ursky, J., Von Gadow, K., Hasenauer, H., Kandler, G., Kenk, G., Kublin, E., Nagel, J., Pukkala, T., Skovsgaard, J.P., Sodtke, R. and Sterba, H. (2002b) Recommendations for standardized documentation and further development of forest growth simulators, *Forstw. Cbl.* 121: 138–151.
- Pretzsch H. (2009) *Forest Dynamics, Growth and Yield: From Measurement to Model*, Springer-Verlag Berlin, Heidelberg, 664pp.
- Ribeiro, N.A., Dias, S., Surovy, P., Goncalves, A.C., Ferreira, A.G. and Oliveira, A.C. (2004) The importance of crown cover on the sustainability of cork oak stands, A simulation approach, *Adv. Geocol.* 37: 275–286.
- Ribeiro, N.A., Surovy, P. and Oliveira, A.C. (2006) Modelling Cork

Oak production in Portugal. In: Hasenauer, H. (Ed.) *Sustainable Forest Management, Growth Models for Europe*, pp 285–313, Springer-Verlag Berlin Heidelberg.

Sterba, H. and Moser, M. (1995) Prognaus—ein Waldwachstumssimulator für Rein- und Mischbestände, *Österreichische For.* 5: 1–2.

Wensel, L.C. and Koehler, J.R. (1985) *A tree growth projection system for Northern California coniferous forests*. Northern California Forest Yield Cooperative, Department of Forestry and Resource Management, University of California, Berkeley, CA.

Wykoff, W.R., Crookston, N.L. and Stage, A.R. (1982) *User's Guide to the Stand Prognosis Model*, INT-133, USDA Forest Service, Ogden, UT.