

forstarchiv 87, 11-23
(2016)

DOI 10.4432/0300-
4112-87-11

© DLV GmbH

ISSN 0300-4112

Korrespondenzadresse:
jthiele@gwdg.de

Eingegangen:
05.10.2015

Angenommen:
16.12.2015

Design and implementation of a web-based decision support system for climate change impact assessment on forests

JAN C. THIELE¹ and ROBERT S. NUSKE²

¹ Department of Ecoinformatics, Biometrics and Forest Growth, Georg-August University of Göttingen, Büsgenweg 4, 37077 Göttingen, Germany

² Northwest German Forest Research Station, Grätzelstraße 2, 37079 Göttingen, Germany

Abstract

Climate change has altered and further will change the environmental conditions for many sectors. Whereas annual cropping systems can be adapted yearly, decisions in forest management usually have long-lasting effects. Depending on the region and tree species rotation periods range between 80 and 180 years in Central Europe. Therefore, today's tree species selection should also take into account the impacts of climate change in the future.

Although many attempts have been made to understand single aspects of climate change impacts on forests, the available knowledge has to be conflated into an integrated assessment to support decision making. A comprehensive system comprising different impact models and an economic assessment suitable for Central European forests and driven by high-resolution temporal- and spatial data is currently missing.

We present a conceptual design and a reference implementation of a Decision Support System (DSS) tailored to assess climate change impacts on German forests. To provide high ease-of-use, the system was implemented as a web application and offers information for single stands on-demand as well as interactive maps and preprocessed assessments on a coarser level for entire Germany. To create such a complex, integrated system from legacy models written in different programming languages the interfaces had to be developed carefully. The key of this DSS is its building blocks: established models describing different climate change impacts. Since the DSS is a very modular system, it is easy to replace submodels and to adapt it to other study areas, forest systems or research questions if suitable parameterized models and input data are available. The presented technical solution is adaptable to other systems integrating existing models and the source code is available.

Key words: climate change adaption, risk assessment, forest management, software architecture, DSS implementation, ecological modelling, model coupling

Introduction

Climate change has altered and further will change the environmental conditions for many sectors (IPCC 2013, Munich Re 2011). Whereas annual cropping systems can be adapted yearly to those changes, decisions in forest management will have long-lasting effects. Depending on the region and tree species, rotation periods in Central Europe mostly range between 80 and 180 years (Lower Saxony Ministry of Food, Agriculture and Consumer Protection 2014). Therefore, today's tree species selection should take into account the climate change impacts in the future (Lindner et al. 2014, Yousefpour and Hanewinkel 2015).

Impacts of the projected climate change on forests are manifold. In Germany, for example, rising temperatures, changing precipitation patterns and increasing frequency of extreme events are projected (Spekat et al. 2007, Becker et al. 2008, Jacob et al. 2008). These are likely to change growth conditions of forests having positive as well as negative effects (Bolte et al. 2009a, Lindner et al. 2010). However, the projected changes will differ temporally and spatially (Enke et al. 2005, Spekat et al. 2007). Thus, choosing a tree species, rotation time, and forest treatment solely based on past experiences may become inadequate in the light of new combinations and dynamics of environmental factors (Kirilenko and Sedjo 2007, Jansen et al. 2008).

Several methods have been developed to evaluate the impact of climate change on Central European forests. For example, Kölling (2007) adapted the method of (bio-)climatic envelope modelling (e. g. Box 1981, Huntley et al. 1995) to tree species in Germany using the factors yearly mean temperature and annual precipitation. A similar approach was pursued by Asche (2009, 2008) who modeled climate change impacts on German forests by empirical ecograms. Climate change was in this case described by a constant temperature increase and relative changes of precipitation to project length of vegetation period, water balance, and nutrient availability. Since climate change does not happen from one moment to the next but is a non-monotone process, it can be important to use high temporal resolution climate data time series at least for some impacts. For example, a single extreme drought year can be limiting to some tree species (Allen 2010), which cannot be detected by a single point in time or longtime average considerations only.

Several models describing single impacts have been developed. For example, Blennow et al. (2010) assessed wind damages of Swedish forests and Panferov et al. (2009) of a forest stand in Germany, Carvalho et al. (2011) projected forest fire danger in Portugal, Murdock et al. (2013) examined the risk of insects outbreaks in a changing climate in British Columbia, and Berc et al. (2013) for Czech Republic, Gustafson and Sturtevant (2013) modeled drought stress mortality in USA, and Coops et al. (2010) evaluated climate change

impact on productivity in British Columbia as well as Albert and Schmidt (2010) in Lower Saxony, Germany.

Although all these attempts are important to understand single aspects of climate change impacting forests, at the end this knowledge has to be combined into an integrated assessment to support decision making. To the best of our knowledge, a comprehensive system comprising different impact models and an economic assessment suitable for Central European forests driven by high-resolution temporal- and spatial data is currently missing. Merely conceptual considerations have been presented several years ago by Lindner et al. (2002). Decision Support Systems (DSS) are important tools able to fill this gap (e. g. Wenkel et al. 2013). Following the definition of Sprague (1980), DSS are usually employed for complex decision making problems of upper level managers where models are combined with data. They are flexible and adaptable, but at the same time the access to the system for non-computer-expert users is easy. Therefore, DSS are ideal platforms for transferring knowledge from science into practice and have been already applied to forest management problems (e. g. Reynolds 2005, Packalen et al. 2013, Vacik et al. 2013). DSS can integrate a wide range of data and models into a single system. They are, thus, well suited to link existing models of different aspects of climate change impact driven by high-resolution climate projection data (Lindner et al. 2002).

In this article we present the design and a reference implementation of a DSS tailored to assess climate change impacts on German forests. The idea of linking established models representing different aspects to an integrated DSS can be adapted to other forest systems and other study areas if suitable parameterized models and input data are available. The focus of this article is neither on the selection of submodels nor the simulation results. Such descriptions can be found in Jansen et al. (2008), Thiele et al. (2009) and Thiele et al. (2010). Instead, we describe an approach to integrate existing models written in different programming languages to an integrated system. Therefore, the structure of the article is adapted to this task and starts with a presentation of the scope, development and concept of the DSS including the master model that links and coordinates the different submodels and the required data. This is followed by a description of the reference implementation of the DSS. Finally we close with discussion and outlook.

Scope, development and concept of DSS

The Decision Support System “forest and climate change” (German: “Wald und Klimawandel”) was developed in the context of a joint research project funded by the German Federal Ministry for Education and Research. The funding measure imposed strict requirements regarding climate projections (see implementation section below) and integration of practitioners. The resulting tool had to be useable for forest managers and consultants. Thus, the DSS should be easy to install, intuitive to use and must answer the right questions in a way easy to understand. Since the topic of climate change impact assessment is extremely complex the presentation of results needs to be as simple as possible. A high level of transparency regarding data, models, and uncertainties of the system should be achieved so that user do not consider it a “black box”. However, to interpret the results in their full extent expert knowledge of forests will still be required.

To comply with these requirements we installed a “stakeholder process” with representatives of forest practitioners from state and private forests as well as chambers and nature conservation organizations. In biannual meetings we gave approximately ten stakeholders the possibility to test development versions on provided notebooks. Three dummy versions (without real results) and two prototypes have been presented. During the process we discussed what informa-

tion such a DSS should provide and which system design would be most usable. The stakeholders commented in oral discussions as well as on questionnaires on layout and navigation of the web application, data input, report (layout and content) as well as background information. This approach resulted in an iterative feedback-driven development of the graphical user interfaces as well as the backend system. The process was quite time consuming but necessary to meet the needs of the DSS users.

Since the simulation is driven by temporal high resolution climate data ranging from 1971 to 2100 and covers entire Germany the amount of data is that large that it is inconvenient to offer a download of the DSS as a desktop application. Furthermore, the software requirements regarding installation and setup of databases, compilers and interpreters are so complex that it would be very challenging for average users to get the system running on their computers. Therefore, it was decided to offer the system as a web-based DSS, where the user requests are processed on a server having all necessary software and data readily available. Since the DSS deals with climate projections ranging over 130 years covering entire Germany and a number of other spatial data sets, the database containing the input data is not small (approx. 28.5 gigabyte storage volume; climate data already comprise more than 98 Mio. records with 22 fields). Additionally, this architecture avoids license conflicts since we were not allowed to share some of the input data unprocessed.

Due to the heterogeneity of the stakeholder group, opinions about purpose of the system and the presentation of the results were quite diverse. Forest practitioners were highly interested in evaluating actual forest stands regarding tree species selection and rotation periods whereas lobbyists were more interested in getting a larger picture of the projected changes on a regional level as well as information about climate change and impacts on forests in general. However, forest practitioners asked as well for an overview of the changing climate and the different risks for forests. They were particularly interested in the location of hot spots and how much they are affected in comparison to others. A summarizing fact sheet containing objectives of the DSS, target audience, application scenarios and derived functional and technical requirements is given in Table 1. Because of the trade-offs between flexibility/complexity and processing time, we decided to serve the diverging demands by splitting the system into different components.

The dynamic component allows the user on-demand simulations based on user inputs regarding geographic location, detailed stand characteristics, soil type and topography. Because the number of possible input combinations is infinite, preprocessing the results was not feasible. Simulation results could not be presented in direct response, as we experienced simulation times of several minutes and users of websites usually do not tolerate response times longer than about 30 seconds (Nah 2004). Therefore, the user will initiate a simulation online and receive the results in form of a PDF attachment via email when ready. Thus, the web application needs a registration and authentication process with validated email addresses and a job queue.

We tried to achieve a balance between the trade-offs flexibility/complexity on one hand and response time on the other hand, by providing a second component. It offers preprocessed results generated in advance with the dynamic simulation system. The results can be requested from a database through the web application and presented immediately. Thus, only a limited number of input combinations can be offered. In contrast to the dynamic component, the climate input data are not downscaled and results are provided at the original resolution of the climate data (20 km pixel size). To cover entire Germany more than 1200 cells are needed. At 1 km resolution it would be even about 415.000 cells. Furthermore, the user can select from up to ten different soil types and five tree species but the stand characteristics and topography are kept constant. The combination of climate data cells with five tree species and at maximum ten soil

Table 1. Fact sheet of DSS scope and requirements defined in the stakeholder process.

<p>1. Objectives Provide information for developing adaptation strategies for forest management valid for entire Germany</p>
<p>2. Target group Decision-makers in forestry as well as political actors in the forest sector</p>
<p>3. Use cases</p> <p>a) Decision-maker in forestry (target group 1) searches for information about tree species selection considering climate change for a specific area/forest stand</p> <p>b) Political actor (target group 2) looks for Germany-wide information about the changing climate and its impact and potential hazards for forests</p> <p>c) User of the web application (target group 1 or 2) query for background information on climate scenarios and employed models</p>
<p>4. Functional Requirements</p> <p><i>Regarding use case a)</i></p> <ul style="list-style-type: none"> • easy data input (e. g. default values, drop-down lists) • defining location via input of coordinates or interactive map • offer a set of relevant soil types for selected coordinates • support soil type selection by short descriptions/images • comparative presentation of different tree species • presentation of results as table • presentation of results as charts • include legends and explanations • ample links to background information <p><i>Regarding use case b)</i></p> <ul style="list-style-type: none"> • interactive map application • selection of various thematic maps • base maps for orientation • map legends • links to background information <p><i>Regarding use case c)</i></p> <p>hierarchical presentation of information about</p> <ul style="list-style-type: none"> • climate scenarios • core concept • submodels
<p>5. Technical requirements</p> <p>The use cases described above are best realized by different components within a common web application. The following components can be derived from the three functional requirements:</p> <p><i>Evaluation of tree species</i></p> <ul style="list-style-type: none"> • WebMap component • dynamically filled menus (database connectivity) • geodatabase with input data and preprocessed scenarios • master model for database query and submodel linkage • self-registration and user management module • job queue for execution of on-demand simulations • graph library for chart output • report generation library <p><i>Interactive maps</i></p> <ul style="list-style-type: none"> • WebMap component • map server • integration of map services <p><i>Background information</i></p> <ul style="list-style-type: none"> • content management system or static web pages <p>The different components have to be integrated into one web application, which requires a webserver. For fast development, high code quality and easy maintenance the application should be based on proven software frameworks and libraries instead of individual developments from scratch</p>

types summarizes to 34.870 preprocessed simulation results to be stored. In contrast to the dynamic component the forest stands are static, i.e. they do not grow, to isolate the influence of the changing climate from effects of different stand characteristics. Therefore, the results have to be interpreted as indicators to assess the magnitude and direction of changes on a regional level.

The input data and the simulation results of the preprocessed component of the DSS are also presented in the form of interactive overview maps (WebMap component) within the web application. This proved to be a very accessible way to compare changes in different regions and to provide an impression of the projected change in general (Dransch et al. 2010). Since the number of results and hence map layers is already large, we provide information only for the most frequent soil type. Less common soil types seemed to add local variation but did not change the overall pattern.

The background information system is the last component of the web application. It contains detailed information about the input data, especially the climate projections, the employed models and their parametrizations and the overall structure of the DSS.

Resulting from the stakeholder process the core of the DSS is a simulation system describing the different impacts of climate change on forests. To serve the needs of the stakeholders the simulation system has to deliver information on:

- site index,
- risk of drought induced mortality,
- risk of wind damage,
- risk of pest hazards, and
- economic assessment of changes.

Impacts by fire events, fungi, viruses, and snow-breaks have been considered to be less important for Germany compared to the ones selected by the stakeholders and are, therefore, not included in the DSS. This is in good accordance with the results of a survey of the forest administrations of the federal states of Germany (Bolte et al. 2009b). The simulation system is the backend of the user-specific on-demand component and also used with reduced complexity for preprocessing the data for the other components, i. e. the overview maps and the preprocessed component. The simulation system consists of input data and various submodels coordinated by a master model. The master model links climate-sensitive submodels simulating forest growth potential, soil water, wind as well as insect phenology and aggregates the projected growth and risks into an economic assessment. These submodels depend on further models delivering the required input data resulting in the following model collection:

- a climate data downscaling component,
- a vegetation period model,
- a rooting depth model,
- a site index model,
- a stand generator,
- a leaf and stem area index model,
- a soil water simulation model with integrated drought stress mortality risk model,
- a three-dimensional wind simulation model with integrated wind damage risk model,
- a pest simulation model for spruce and pine with integrated damage risk model,
- a stand growth simulation model, and
- an economic assessment.

Before we describe the master model, i. e. the schedule of sub-model linkage, we present some conceptional fundamentals. To keep input data requirements and simulation complexity manageable, we used a point-based approach, meaning we run the simulation for a point location specified by the user. All input data, i. e. downscaled climate, soil, topography and nitrogen deposition, is mapped to the

spatial resolution of the input data with the coarsest resolution, in our case (see below) the downscaled climate data with 1 km grid cells. Additionally, the user can change some of the input values if better information is available. The user can choose from a set of typical soil types at the selected point location and provide relief information. This approach ignores the land cover of the point's neighborhood. This is partly due to the fact that most models do not take the neighborhood into account, like the soil water model, and most users are not able to provide the required information. However, the wind model considers the surrounding relief and the stand growth model simulates a representative forest stand of a quarter hectare centered at the point location including competition on tree level. Thus, the spatial scale of the DSS is determined by the spatial scale of input data and the master model itself is scale-independent.

The models are connected by data exchange in a coordinating time interval of 10 years. The coordinating time interval determines the update frequency of the submodels, which is not necessarily identical with their internal time step. For example, the risk models get updated forest stand information every 10 years but the stand growth model runs internally with a 5 year time step and the soil water and the wind models run with a daily time step. In some cases data collections are exchanged at the coordinating time interval providing information at a higher temporal resolution, e. g. the vegetation period model offers an array with yearly start and end dates and the soil water model provides an array of soil moisture indices at daily intervals for the wind damage model. Therefore, the coordinating time interval defines the maximum feedback interval.

To keep the complexity and processing time at bay, we refrained from introducing feedbacks between the damage models and the stand growth. The stand growth model describes a growth potential and risks are defined as loss of part of the stand in percent of the area. Thus, damage does not occur at single tree level and does not alter the stand structure. This simplification must be kept in mind when interpreting the simulation results. It also means that the predisposition of forest stands to pest infestation caused by wind throw events is not represented in the simulation. Since pest control management can reduce these risks considerably, this simplification seems justified.

Beside the coordinating time interval of 10 years, there is an output time interval of 30 years. This is according to the usage instructions of the climate scenario data (DKRZ 2007) ensuring results of the climate model are interpreted as climate trends and not as single events at specific years.

Output variables of the DSS are:

- site index (tree height at age 100),
- risk of wind damage,
- risk of pest hazards,
- risk of drought induced mortality,
- wood production value, and
- risk costs.

For the output time interval the results of the coordinating time interval are averaged for the site index using an arithmetic mean. For the risks the aggregation is calculated as follows:

$$Risk_{30} = 1 - (1 - Risk_{1-10}) * (1 - Risk_{11-20}) * (1 - Risk_{21-30}) \quad (1)$$

where $(1 - Risk)$ gives the conditional survival probability of the respective coordinating time interval (for details, see Staupendahl 2011). The 10-year risk values are aggregated for all risk types to 30-year drop out probabilities. Afterwards, the 30-year risks of all risk types are aggregated to a total risk value in the same manner:

$$Risk_{total} = 1 - (1 - Risk_{biotic}) * (1 - Risk_{drought}) * (1 - Risk_{wind}) \quad (2)$$

The total conditional probability that a stand, that survived until the beginning of a 30-year period still exists at its end, is one minus the total risk. The economic assessment is based on the aggregated site indices and the total drop-out probabilities ($Risk_{30}$) and delivers wood production value and risk costs of 30-year periods.

The driving forces of the simulation are the daily climate data available for the time periods 1971–2000 and 2010–2100. In the second period the simulation is carried out once for each of the climate scenarios (A1B and B1). The topography and the soil characteristics are assumed to be constant over the considered periods, whereas the climate and the nitrogen deposition are available as time series derived from scenarios of possible future development.

The master model is characterized by two nested loops representing the 30-year output and the 10 year coordinating time period, as described above. Figure 1 contains a schema of the master model including input data, submodels and process flow. A simulation run is started based on the user provided information (starting time, point location, soil type, tree species and optional elevation) together with the corresponding soil parameters and the topography of the neighborhood. The simulation always runs till the year 2100, so the number of iterations depends on the user selected starting time. To provide a reference for the climate projections the DSS offers, additionally to the starting times 2011, 2041 and 2071, the starting time 1971. Since there is a 10-year gap between the reference period and the future periods the stand growth model continues in this case for ten years outside of the loops.

The 10-year coordinating time interval starts with loading and preprocessing daily climate data and yearly nitrogen deposition in yearly resolution for the current time interval. Climate data of the 3×3 neighborhood of the user selected location provide the spatial support in the downscaling module following the recommendations of the usage guidelines of climate projections (DKRZ 2007). The downscaled climate data are then used to calculate the start and the end of the vegetation period for every year within the coordinating time interval. The vegetation period model also provides climatic water balance, mean yearly temperature, annual precipitation as well as mean temperature sum within the vegetation period. The rooting depth, soil horizon update routine and root density/distribution model based on the forest stand and soil characteristics run parallel. On the basis of the results of the vegetation period model, the soil characteristics and the nitrogen depositions the site-index model estimates the growth capacity. The site index is the main input for the forest stand growth and stand generator. In the first iteration a forest stand is created by the stand generator using yield tables. This virtual forest stand is then grown throughout the rest of the simulation. The generated stand provides the necessary information for the leaf and stem area index. Having stand, soil and climate data available, the soil water model including a drought stress mortality risk module can be run. It returns the mortality risk as well as soil water indices. The soil water information is an important input for the wind and the biotic risk models. The wind model needs additional information about location, topography, stand characteristics including rooting depths, and daily climate. The model describing the biotic damage to spruce and pine requires location, stand age, soil water as well as climate data. Updating the site index and growing the virtual forest stand in preparation for the next iteration is the last step in the 10-year loop.

After three iterations of the inner 10-year loop, risk and site indices are aggregated and the economic assessment is carried out. If the end of the simulation period is not yet reached the 10-year loop is entered again.

Please refer to the listing of the in- and output variables of the submodels in Appendix A to gain a deeper understanding of the interdependencies of the submodels (https://sourceforge.net/projects/dss-wuk/files/Appendix_Forstarchiv/).

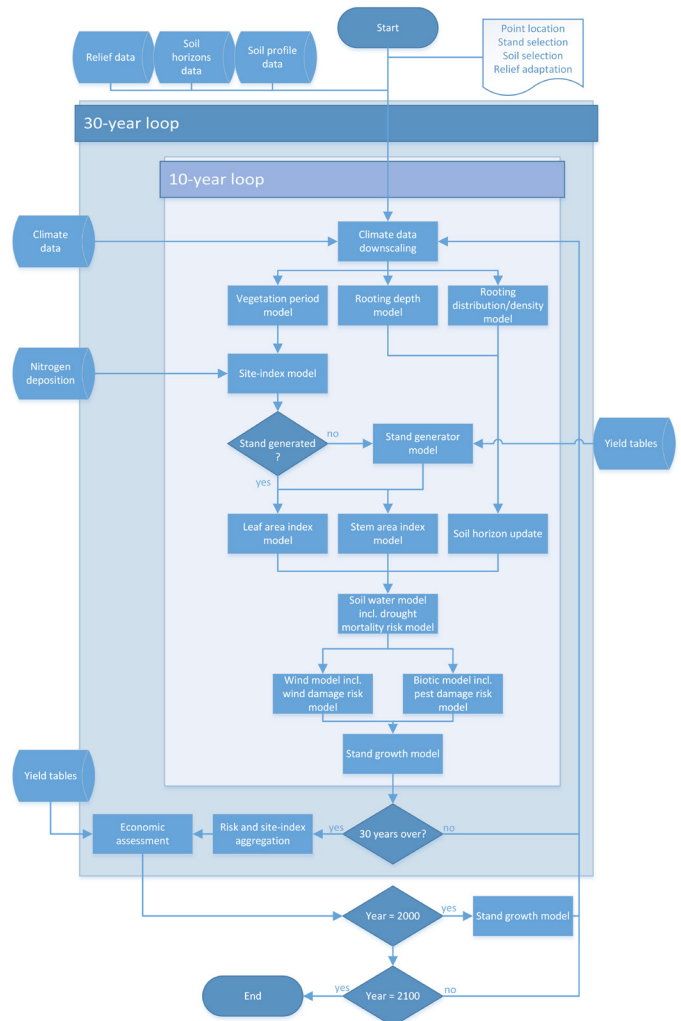


Figure 1. Schema of the master model including input data, submodels and process flow.

Implementation

Building on the requirements resulting from the stakeholder process and the concept of the simulation present as a composition of input data and submodels in the master model we now outline the implementation of the reference system.

Data and Database

Since the DSS shall provide valid results for entire Germany, datasets covering the whole country were needed for the reference implementation. This proved to be a severe restriction resulting in coarser resolution of some input data because data of high spatial resolution is only available for parts of Germany. However, as the master model itself is scale independent, it is possible to exploit high resolution data when applied locally as shown in Thiele et al. (2010).

For the reference implementation following input data sources were used:

- Relief* information is derived from digital elevation models based on the Shuttle Radar Topography Mission (SRTM) dataset Ver-

sion 4 edited by the CIGAR Consortium for Spatial Information in 3' resolution, i. e. approx. 90 meters (Jarvis et al. 2008) and aggregated to the required target resolution of 1 and 20 km, respectively.

- b) *Soil* data consisted of 124 soil types as defined by the Forest Soil Map (Wald-BÜK 1:1 Mio., Richter et al. 2007). The user can select from maximum 10 typical soil types at each location. The soil types provide information about available soil water capacity, effective cation exchange capacity, and groundwater depth. More detailed information such as bulk density, stone content, organic matter, available field capacity is provided for the separate horizons.
- c) Annual deposition rates of *nitrogen* are preprocessed with the MAKEDEP model (Alveteg et al. 1998) and stored at a spatial resolution of 1 km.
- d) *Forest stand* information is generated based on a combination of user input, climate sensitive site index model, yield tables (Schober 1995, with adaptations by Wollborn and Böckmann 1998), and the stand generator included in TreeGrOSS (Döbeler et al. 2002, Nagel 2011). For the economic model additional yield tables generated with TreeGrOSS are preprocessed and stored in the database.
- e) As *climate* forcing data the climate scenario 20C3M for the period of 1971–2000 (reference) and SRES A1B and B1 (IPCC 2000) for the period of 2011–2100 modeled by coupled AOGCM ECHAM-5-MPIOM (run 1). The data were regionalized with the Climate-Local-Model (CLM) (Hollweg et al. 2008) and interpolated to the spatial resolution of 0.2° (Datastream 3; Lautenschlager et al. 2009a, 2009b) in daily time step. The climatic data were then bias corrected using the observation data of the German Weather Service, implementing the delta-change and linear transfer functions (Mudelsee et al. 2010). The vapor pressure was adjusted to bias-corrected temperature.

The selection of a relational database management system (RDBMS) with spatial extension was based on a simplified utility-analysis (see Table 2) at the time of implementation (2008). The most common RDBMS offering support for spatial data were compared regarding different criteria. The solution with the highest utility value was selected. To facilitate reusability and code transparency we focused on free and open source software (FOSS) with one exception due the very limited offer of FOSS licensed solutions at that point.

Based on the utility-analysis we selected the RDBMS PostgreSQL (The PostgreSQL Global Development Group 2015) with the spatial extension PostGIS (Refractions Research 2015). PostgreSQL is a free and open-source object-relation database management system.

PostGIS follows the Simple Features for SQL specification from the Open Geospatial Consortium (OGC 2010) and adds geographic objects and functions to PostgreSQL.

Since this DSS is concerned with the impact of climate change on forest stands, climate data are important in the data model. The table containing the spatial location of the CLM cells and the relief information needed for downscaling is central in the data model (see Figure 2 for visualization on table level; see Appendix B, https://sourceforge.net/projects/dss-wuk/files/Appendix_Forstarchiv/, for model visualization on field level). The tables with the climate projections as well as the tables concerning the job queue link to it. If a user asks for a simulation run of the dynamic component, the request is stored in a table which is checked every 15 minutes by a cron job that also starts the next simulation. A list of finished and failed jobs is kept for later inspection. The nitrogen deposition is stored at the resolution of the SRTM elevation model. Hence, both tables are connected via a table containing the spatial location of the SRTM cell centers. The ten most common soil types at each CLM cell are stored to be presented to the user in a selection list. The soil types are connected to the horizons that constitute the respective profiles. All yield tables of the considered tree species are stored in one table and accompanied by a table of initial stand characteristics. Another set of tables caters the preprocessed component of the DSS (see Figure 3 for visualization on table level; see Appendix B, https://sourceforge.net/projects/dss-wuk/files/Appendix_Forstarchiv/, for model visualization on field level). They contain information about the spatial extent of the preprocessed simulation runs, the considered soil types and their properties and finally the simulation results.

Software

The implementation of the web-based DSS relies on a web framework for common activities such as database access, session management, input validation, templating. Using standard components especially for security aspects keeps the source code compact and frees resources for the development of the DSS. To select the most suitable framework we assessed three widely used web application frameworks all associated with different programming languages using a simplified utility-analysis (see Table 3). We selected Django (Django Software Foundation 2015a), a web-application framework for the programming language Python (Python Software Foundation 2015). More precisely we used Django together with additional spatial functions for building GIS Web applications distributed as GeoDjango (Django Software Foundation 2015b). Selecting this framework we also choose Python, a general-purpose, high-level programming language

Table 2. Utility-analysis of most common Geodatabase solutions (utility values per category from 0 to 4, where 4 is the best value; equally weighted).

	Oracle Spatial	MySQL + Spatial-Extension	PostgreSQL + PostGIS-Extension
Version	11g	5.0.51	8.3.1
License	proprietary	GPL	PostgreSQL + GPL
ACID conformity	4	2	4
Referential integrity	4	4	4
max. DB size	no/4	no/4	no/4
Save geometries	3	2	4
Geodata processing	3	1	4
OGC SFS	4	1	4
Utility value sum	22	14	24

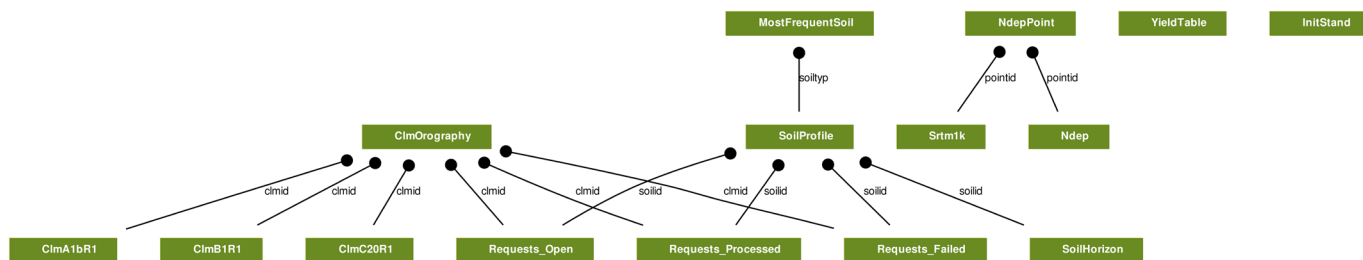


Figure 2. Data model of input data on table level for the dynamic simulation component. The connecting lines visualize relations between fields of different tables (see Appendix B, https://sourceforge.net/projects/dss-wuk/files/Appendix_Forstarchiv/, for model visualization on field level).

designed for high readability, fast development and small codes. Django follows the model-view-controller architectural pattern and is, therefore, especially suitable for database-driven applications coming with an object-relational mapper. Many common features needed for web-applications are already on-board or are available through extensions. In particular we used the user authentication, which is shipped with Django and added the registration extension (Cutler 2015). Users have to register themselves and provide a valid email address because the dynamic component sends simulation results as PDF via email. To generate the PDF files containing text, tables and figures we employed ReportLab (ReportLab Inc. 2015) a free and open-source engine for creating data-driven PDF documents written in Python. To dynamically generate the HTML pages we used the Django templating language and the Cascading Style Sheets (CSS) framework YAML (Yet Another Multicolumn Layout, Jesse 2013) for layout and styling.

WebMaps were needed at two different places in the web application: 1) location selection via a mouse-click within a zoomable map, 2) presentation of the main parameters and preprocessed results as interactive maps. Non trivial WebMap applications consist usually of two parts, a client-side application presenting the maps in the browser and a map server providing map layers in form of images. We selected the client-side OpenLayers JavaScript library (OpenLayers Contributors 2015) after an evaluation of common libraries (see

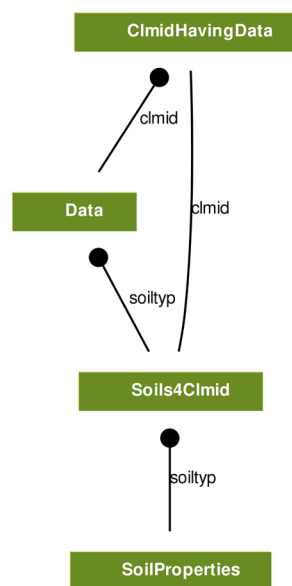


Figure 3. Data model of the preprocessed component on table level. The connecting lines visualize relations between fields of different tables (see Appendix B, https://sourceforge.net/projects/dss-wuk/files/Appendix_Forstarchiv/, for model visualization on field level).

Table 3. Utility-analysis of different web-application frameworks (utility values per category from 0 to 4, where 4 is the best value; equally weighted).

	(Geo-) Django	Ruby on Rails	Struts 2
Version	0.96	2.0.2	2.0.11
License	BSD	MIT	ASL
Language	Python	Ruby	Java
Object-relational mapper	4	4	4
Security framework	4	3	3
Template system	4	4	4
Form validation	4	4	4
Session management	4	4	4
Caching system	4	4	3
Model-View-Controller paradigm	4	4	4
Native GIS support	4	1	1
Performance	4	3	2
Development speed	4	3	2
Utility value sum	40	34	31

Table 4). For the WebMap in the preprocessed component we implemented the user interface (UI) using GeoExt (GeoExt Community 2010). GeoExt is a JavaScript library connecting OpenLayers with additional UI-functionalities of ExtJS (Sencha Inc. 2015a, 2015b), for building complex WebMap application on the web. It offers a clearly arranged and very accessible layer tree providing structure for the numerous map layers.

For providing the maps of input and preprocessed data to OpenLayers we also needed a map server creating and delivering the requested maps through the Web Mapping Service (WMS) standard (OGC 2015). We selected the UMN MapServer (see Table 5) and defined the required mapfiles containing the definition of the data source and styling of the various map layers.

Beside the presentation of selected maps the full range of preprocessed results can be requested via the web application and are presented directly in the browser. The results are loaded for a user selected location, soil type, and tree species. The user can switch between a table and a chart view. The tables are constructed via the Django templating language and the charts are requested from the Google Charts API via the GChart Python Wrapper for Django (Quick 2009).

A major part of the DSS is the dynamic component driven by the master model. Although the master model itself is not a web application it nevertheless uses the web framework because the connection to the database is conveniently encapsulated with the object-relational mapper and comes with needed spatial functions. The master model is responsible for the flow of control and data as well as managing the state of the models. The last point is essential because some of the submodels must keep their state from one iteration to the next.

The soil water model, for example, must keep the state of soil water between two 10-year periods and continues with an updated forest stand and new climate data in the next iteration. Furthermore, the output of one submodel serves as input of other submodels resulting in an intensive exchange of a large amount of data. Since exchanging data via input and output files is very slow and error-prone, especially when multiple simulations run in parallel on one computer, we decided to make all models directly accessible at run-time. Submodels implemented by us were written in Python facilitating easy data exchange and flow control. However, several submodels have been delivered in programming languages other than Python (cf. Table 6). Therefore, we needed to implement wrappers for the submodels to facilitate interchangeable data types and flow control across submodels, e. g. instantiation and function/method calling from Python.

Two submodels had to be rewritten since the provided code base could not be incorporated into the master model. The soil water model was originally implemented in Visual Basic, a language native to Microsoft Windows. The model makes extensive use of Windows specific libraries and thus cannot be executed under Linux by the development and runtime environment Mono (Mono Project 2015). Furthermore, the implementation does not adhere to the object-oriented paradigm making it difficult to store and clone complex soil water states. Since the soil water model runs internally with a daily time step, it is performance critical for the entire master model. Therefore, we decided to re-implement the model in C++ (Standard C++ Foundation 2015) following the object-oriented paradigm.

The second submodel rewritten was the economic assessment, as it was delivered in form of an Excel spreadsheet. Since the model is only executed in the outer 30-year loop, performance is not crucial

Table 4. Utility-analysis of different web mapper frameworks (utility values per category from 0 to 4, where 4 is the best value; equally weighted).

	OpenLayers	WMS Mapper	Mapbender	Mapbuilder
Version	2.5	0.03	2.4.5	1.0.1
License	BSD	AFL	GPL	LGPL
Zoom bar	4	0	0	0
Panning	4	0	4	2
Tiling	4	4	0	0
Layer overview	3	0	4	4
Proprietary layer	3	0	0	0
Client side	4	4	0	0
Extensible	4	2	3	2
Memory usage	4	4	1	1
Utility value sum	30	14	12	9

Table 5. Utility-analysis of different map server (utility values per category from 0 to 4, where 4 is the best value; equally weighted).

	UMN MapServer	GeoServer	deegree
Version	5.0.2	1.6	2.1
License	MIT	GPL	LGPL
Performance	4	2	2
Legend	4	4	4
WMS standard	4	4	4
Server demands	4	1	1
Utility value sum	16	11	11

Table 6. Listing of submodels with programming languages other than Python and employed wrapper libraries.

Submodel	Programming Language	Wrapper library
Climate data downscaling	Fortran	F2PY (The Scipy community 2014)
Site-index model	R	RPy2 (Gautier 2014)
Soil water model	C++	SWIG (SWIG2015)
Wind model	Fortran	F2PY (The Scipy community 2014)
Biotic model	R	RPy2 (Gautier 2014)
Stand growth model	Java	JCC (Apache Software Foundation 2012)

and we re-implemented the Excel calculations in Python easing data exchange and control flow. A transformation of the Excel spreadsheet into an OpenOffice spreadsheet, which could be connected to Python through the pyUNO bridge (Apache OpenOffice 2010) would have been an alternative but would have come with enormous dependencies.

All submodels are split into separate modules making it easy to replace submodel implementations. This is especially true for submodels implemented in Python. Submodels written in other programming languages come with an overhead. The necessary wrappers are described in the following.

The climate data downscaling model and the wind model are implemented in Fortran. We used the command line tool F2PY (The Scipy community 2014) to compile the Fortran source code into a dynamic library containing Python wrappers (cf. Figure 4). The dynamic library is then loaded from Python and Fortran functions are called directly. The different data types are also converted from Fortran to Python and vice versa by the wrapper code in the dynamic library.

Integration of models provided as R scripts and R objects (site-index model and the biotic risk models) was achieved with the RPy2 package (Gautier 2014). RPy2 provides an interface to R from Python. It is not necessary to create individual wrappers since the RPy2 package allows to call R functions from Python and transforms data structures between the two languages (cf. Figure 5).

We used JCC to make the stand growth model written in Java callable from Python. JCC is a C++ code generator for calling Java from C++/Python (Apache Software Foundation 2012). JCC first generates a C++ wrapper for public Java methods and then a Python wrapper on top. Before JCC can do its work the Java source code has to be compiled with a Java compiler to a Java class file or a Java Archive (JAR). All of this is finally bundled into a Python-Egg (cf. Figure 6). JCC requires a Java Runtime Environment to operate and provides functions to start a Java Virtual Machine from Python.

The soil water model, for performance reasons written in C++ (see above), was made available to Python using the Simplified Wrapper and Interface Generator (SWIG 2015), a tool to connect C and C++ code with several high-level programming languages. Data types, imports, functions and variables were declared in an interface file, which is then compiled using SWIG, resulting in a C++ wrapper and a Python module. In a second step the original C++ source code is compiled together with the wrapper using a C++ compiler such as GCC (Free Software Foundation Inc. 2015) and the resulting object files are packaged into a shared library. The Python module provides access to the shared library. The workflow is similar to the F2PY tool but needs more manual steps (cf. Figure 7).

Having discussed all submodels and wrappers we now turn to the parts of the master model implemented as class ModelEngine. The constructor of the class ModelEngine takes the user input (as described above) as arguments. Compulsory arguments are longitude and latitude, the soil type, the tree species, age and yield class as well

as the start year. Optionally the user may provide more precise information regarding elevation, aspect and slope at the specified location. Furthermore, an indicator argument handles soil water model errors and reduces the step size of the integration interval if necessary. Moreover, static input data such as relief and soil data are loaded from the database and all information is stored as class members. Finally the constructor initializes or instantiates the submodels and stores the objects as class members as well.

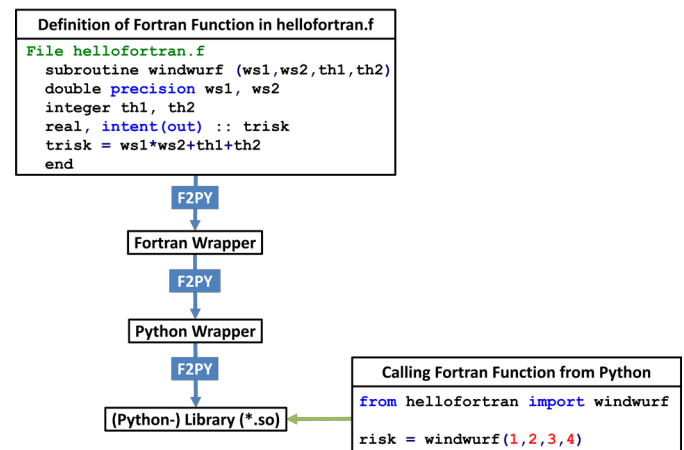


Figure 4. Simplified example of wrapping Fortran code to gain access from Python using F2PY. Green arrow indicates access from Python at run-time. Blue arrows indicate transformations from original Fortran code into a Python library generated by F2PY in advance.

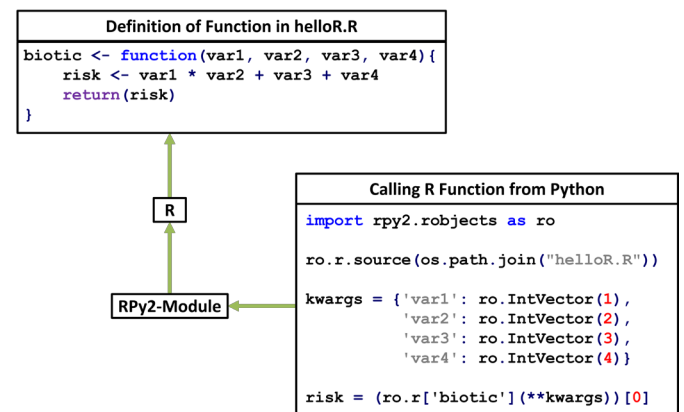


Figure 5. Simplified example showing how to call an R function from Python using RPy2. Green arrows indicate access at run-time.

The starting point for the simulation is the method taking care of the outer 30-year loop. It takes two arguments defining start and end of the simulation time span. After initializing an empty dictionary to store the results of the 30-year output intervals the inner 10-year loop is run three times. The method coordinating the 10-year loop calls sequentially the methods dealing with the different submodels and some helper methods, e. g. for data handling (see Table 7). The output of the three 10-year time intervals is aggregated and stored in the results dictionary. The economic assessment submodel runs on the aggregated values.

In case the simulation started with the year 1971 the gap between the reference period and the climate projections is closed by running the stand growth model for 10 years with constant site-index. If the simulation enters the climate projections (start year 2011) the simulation is carried out for two climate scenarios (A1B and B1) successively as described above.

In previous sections we gave a comprehensive overview of the architecture of the master model and our reference implementation, but the most truthful and detailed description is the source code itself. It might as well serve as starting point for similar projects in the future. Thus, the source code of the application including the Django-based web application is available at <https://sourceforge.net/projects/dss-wuk>.

Discussion and Outlook

The construction of the DSS “forest and climate change” started with dummies and prototypes and a stakeholder process, leading to a conceptional model and ended with a successful implementation of a reference system. The stakeholder process provided the information needs of forest practitioners regarding the impact of climate change on forest. The process resulted in a requirements specification for the DSS. Involving stakeholders in the DSS development is very important but nevertheless often neglected leading to systems not accepted by the target audience (Lynch et al. 2000, Lynch and Gregor 2004).

We, thus, designed the conceptional model and the technical concept on the basis of the collected stakeholder input considering necessary model complexity and simulation time. By carefully heeding stakeholder feedback the DSS received some attention by the target audience (e. g. Haufe 2011) and was promoted by stakeholders (e. g. Hillmann and Zimmeck 2011a, 2011b, 2011c). Nevertheless, the usefulness of the system was impaired by inconsistencies in the climate projections (e. g. average temperature below minimum temperature in some cases), which are communicated on the web page of the reference implementation. We tried to correct for some known biases (see e. g. Lindau and Simmer 2012), but we have not been able to remedy deficiencies completely. Moreover, the regionalized climate data might not have been adequate for usage at this level. Furthermore, the knowledge needed to parameterize the employed risk models might still be weak. Nevertheless, research on effects of climate change on forests is still ongoing and the creation of such a coupled modeling system helped to point out where further research is needed.

From a technical point of view, we have shown that an integrated system can be built from established models. By splitting the system into different components (dynamic, preprocessed and background information) with different targets concerning reaction time and spatial resolution the application can cater different needs. The whole application is built on well-known frameworks to keep the source base small and therefore easy to maintain. Since the application is highly modularized, submodules can be replaced quickly by other implementations. This is a very important architectural feature as knowledge in climate change impact still increases rapidly and the development of more adequate simulation models proceeds apace. We presented a way to interface established models implemented in different programming languages to use them as building blocks of an integrated DSS. The need of such interfacing mechanisms will increase as climate change impact assessments need to incorporate different impact models into one integrated simulation system.

We inserted our simulation system into a web-application for easy use by forest practitioners. However, the employed framework and our master model can also be run from the command line or called

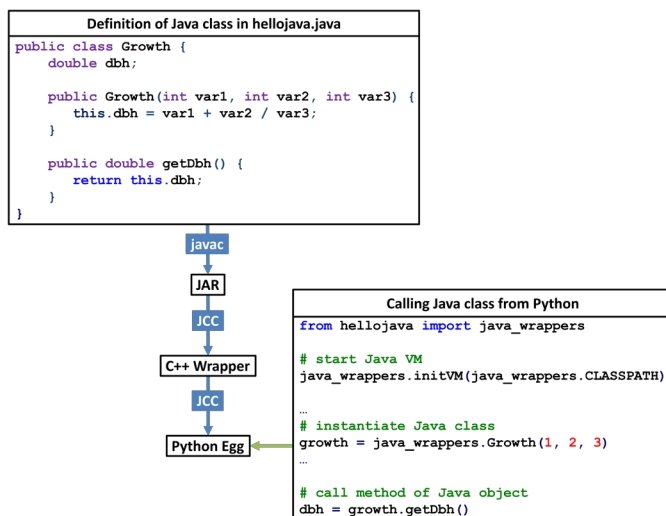


Figure 6. Simplified example showing how a Java class can be called from Python using JCC. The green arrow indicates access from Python at run-time. Blue arrows indicate transformations from Java code to a Python-Egg (i. e. a single-file importable Python project) generated in advance.

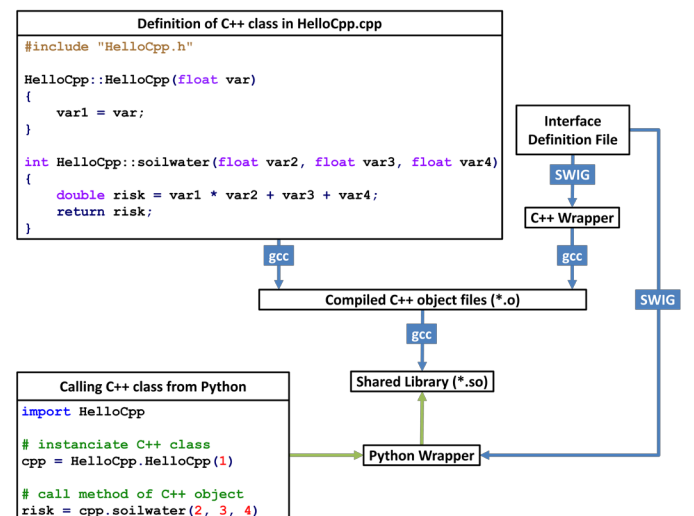


Figure 7. Simplified example of wrapping C++ code to gain access from Python using SWIG. Green arrows indicate access from Python at run-time. Blue arrows indicate transformations from C++ code into a Python library.

Table 7. List of method and function calls in inner 10-year loop in the order of execution.

No.	Function/Method	Module	Description
1	calculate_days_and_dates	mastermodel	Calculate start and end dates
2	get_clm_climate_data	input	Load and prepare climate data from database
3	regionalize	submodels.downscaling.downscaling	Run downscaling submodel (Interface to Fortran)
4	moistureCorrection	submodels.downscaling.downscaling	Run bias correction of moisture/humidity in CLM data (Interface to Fortran)
5	calc_veg_period	submodels.phenoallometric.vegperiod	Call the vegetation period submodel
6	calc_nutrient_number	submodels.soilparameters.nutrients	Calculate a soil nutrient class for site-index
7	calculate_rootdepth	submodels.phenoallometric.rootdepth	Calculate rooting depth
8	update_soilhorizons	submodels.soilparameters.changesoil	Update soil profile considering rooting depth
9	set_soil_to_brook	submodels.soilwater.handler	Set soil characteristics within soil water submodel (Interface to C++)
10	change_DRAIN	submodels.soilwater.handler	Change DRAIN parameter of soil water submodel (Interface to C++)
11	calc_rootdensity	submodels.phenoallometric.rootdensity	Call the root density calculation
12	get_nitrogen_deposition	input	Load and prepare the nitrogen deposition data from database
13	get_site_index	submodels.siteindex.handler	Call the site-index submodel (Interface to R)
14	get_yieldtable_values_from_yieldclass	input	If stand was not created so far, load and prepare initial stand characteristics from database
15	generate_stand	submodels.standgrowth.handler	If stand was not created so far, call the stand generator submodel (Interface to Java)
16	calc_maxlai	submodels.phenoallometric.lai	Call the leaf area index submodel
17	calc_sai	submodels.phenoallometric.lai	Call the stem area index submodel
18	set_standparameters_to_brook	submodels.soilwater.handler	Send stand characteristic to soil water submodel (Interface to C++)
19	reduce_climatic_arrays	mastermodel	Reduce time series of climate data to 10 years
20	set_climate_to_brook	submodels.soilwater.handler	Send climate data to soil water submodel (Interface to C++)
21	run_brook	submodels.soilwater.handler	Run the soil water submodel (Interface to C++)
22	run_scadis	submodels.wind.handler	Run the wind risk submodel (Interface to Fortran)
23	run_ips_typo_model	submodels.biotic.handler	If tree species spruce is simulated, call the <i>Ips typographus</i> risk submodel (Interface to R)
24	run_phy_cyanea_model	submodels.biotic.handler	If tree species pine is simulated, call the <i>Phaenops cyanea</i> risk submodel (Interface to R)
25	run_standgrowth	submodels.standgrowth.handler	Call the stand growth submodel (Interface to Java)
26	update_stand_information	mastermodel	Update the stand characteristics for the next iteration

from other code enabling simulation studies or integration into other systems. We simulated, for instance, the impact of climate change on spruce stands of an entire forest company (Thiele et al. 2010). Since the Desktop GIS QGIS comes with a Python API, it would be possible to integrate this DSS as a plugin.

The concept as well as most parts of the implementation are generic enough to be adaptable to other regions and spatial and temporal scales. Therefore, we offer our work to anybody interested to learn or build upon it.

Acknowledgment

We thank two anonymous reviewers for their valuable comments on an earlier version of the manuscript. Furthermore, we thank Prof. Dr. Joachim Saborowski for his trust in our work, his guidance and his communication skills as well as our colleagues of the DSS-WuK project for contributing sub-models. The development was supported by the German Federal Ministry of Research and Education (BMBF) as part of the project "Anpassungsstrategien für eine nachhaltige Waldbewirtschaftung unter sich wandelnden Klimabedingungen – Decision Support System Wald und Klimawandel (DSS-WuK)" (No. 01LS05117/01LS05118, Program klimazwei). We gratefully acknowledge this support.

References

- Albert M., Schmidt M. 2010. Climate-sensitive modelling of site-productivity relationships for Norway spruce (*Picea abies* (L.) Karst.) and common beech (*Fagus sylvatica* L.). *For. Ecol. Manage.* 259, 739–749
- Allen C.D. 2010. A global overview of drought and heat-induced tree mortality reveals emerging climate change risks for forests. *For. Ecol. Manage.* 259, 660–684
- Alveteg M., Walse C., Warfvinge P. 1998. Reconstructing historic atmospheric deposition and nutrient uptake from present day values using MAKEDEP. *Water Air Soil Pollut.* 104, 269–283
- Apache OpenOffice 2010. PyUNO bridge – Apache OpenOffice Wiki. https://wiki.openoffice.org/wiki/PyUNO_bridge (accessed 2015/09/12)
- Apache Software Foundation 2012. Apache Lucene – Welcome to JCC, Py-Lucene's code generator. <https://lucene.apache.org/pylucene/jcc/> (accessed 2015/09/12)
- Asche N. 2008. Waldtypenkarten für Nordrhein-Westfalen auf Basis der forstlichen Standortklassifikation unter besonderer Berücksichtigung des Klimawandels. *AFZ-DerWald* 63 (1), 25–27
- Asche N., 2009. Waldstandorte im Klimawandel. *Forst Holz* 64 (9), 14–17
- Becker P., Deutschländer T., Koßmann M., Namyslo J., Knierim A. 2008. Klimaszenarien und Klimafolgen. Informationen zur Raumplan. 6/7, 341–351
- Berec L., Doležal P., Hais M. 2013. Population dynamics of *Ips typographus* in the Bohemian Forest (Czech Republic): Validation of the phenology model PHENIPS and impacts of climate change. *For. Ecol. Manage.* 292, 1–9
- Blennow K., Andersson M., Sallnäs O., Olofsson E. 2010. Climate change and the probability of wind damage in two Swedish forests. *For. Ecol. Manage.* 259, 818–830
- Bolte A., Ammer C., Löf M., Madsen P., Nabuurs G.-J., Schall P., Spathelf P., Rock J. 2009a. Adaptive forest management in central Europe: Climate change impacts, strategies and integrative concept. *Scand. J. For. Res.* 24, 473–482
- Bolte A., Eisenhauer D.-R., Ehrhart H.-P., Groß J., Hanewinkel M., Kölling C., Profft I., Rohde M., Röhe P., Amereller K. 2009b. Klimawandel und Forstwirtschaft – Übereinstimmungen und Unterschiede bei der Einschätzung der Anpassungsnotwendigkeiten und Anpassungsstrategien der Bundesländer. *VTI Agric. For. Res.* 4, 269–278
- Box E. 1981. Macroclimate and plant forms: An introduction to predictive modeling in phytogeography, task for vegetation science. Springer, Dordrecht
- Carvalho A.C., Carvalho A., Martins H., Marques C., Rocha A., Borrego C., Viegas D.X., Miranda A.I. 2011. Fire weather risk assessment under climate change using a dynamical downscaling approach. *Environ. Model. Softw.* 26, 1123–1133
- Coops N.C., Hember R.A., Waring R.H. 2010. Assessing the impact of current and projected climates on Douglas-Fir productivity in British Columbia, Canada, using a process-based model (3-PG). *Can. J. For. Res.* 40, 511–524
- Cutler A. 2015. django-registration-redux 1.2: Python Package Index. <https://pypi.python.org/pypi/django-registration-redux/> (accessed 2015/09/12)
- Django Software Foundation. 2015a. The Web framework for perfectionists with deadlines | Django. <https://www.djangoproject.com/> (accessed 2015/09/12)
- Django Software Foundation. 2015b. GeoDjango. <http://geodjango.org/> (accessed 2015/09/28)
- DKRZ 2007. Nutzungsbedingungen für die Daten der Klimasimulationen mit dem Regionalmodell CLM. http://mms.dkrz.de/pdf/klimadaten/vortraege/extern/SGA-Files/Nutzungshinweise-CLM-DEZ2007_01.pdf (accessed 2015/12/26)
- Döbblers H., Albert M., Schmidt M., Nagel J. 2002. BWINPro – Programm zur Bestandesanalyse und Prognose. Handbuch zur Version 6.1. Niedersächsische Forstliche Versuchsanstalt, Abteilung Waldwachstum, Göttingen
- Dransch D., Rotzoll H., Poser K. 2010. The contribution of maps to the challenges of risk communication to the public. *Int. J. Digit. Earth* 3, 292–311
- Enke W., Deutschländer T., Schneider F., Küchler W., 2005. Results of five regional climate studies applying a weather pattern based downscaling method to ECHAM4 climate simulations. *Meteorologische Z.* 14, 247–257
- Free Software Foundation Inc. 2015. GCC, the GNU Compiler Collection – GNU Project – Free Software Foundation (FSF). <https://gcc.gnu.org/> (accessed 2015/04/10)
- Gautier L. 2014. rpy2. <http://rpy.sourceforge.net/> (accessed 2015/09/12)
- GeoExt Community 2010. JavaScript Toolkit for Rich Web Mapping Applications – GeoExt v1.1. <http://geoext.org/> (accessed 2015/09/12)
- Gustafson E.J., Sturtevant, B.R. 2013. Modeling forest mortality caused by drought stress: Implications for climate change. *Ecosystems* 16, 60–74
- Haufe F. 2011. Web als Berater bei der Baumartenwahl. <http://www.lwk-niedersachsen.de/index.cfm/portal/4/nav/0/article/16225.html> (accessed 2015/12/10)
- Hillmann M., Zimmeck W. 2011a. Web als Berater bei der Baumartenwahl. *Land Forst* 10, 60–61
- Hillmann M., Zimmeck W. 2011b. Klimafitte Wälder aufbauen – Web-basiertes Simulationsprogramm hilft bei der Baumartenwahl. *BWagrar* 16, 21–22
- Hillmann M., Zimmeck W. 2011c. Die Baumart im Internet wählen. Web-basiertes Computerprogramm hilft Klimarisiko abschätzen. *BLW* 22, 40
- Hollweg H., Böhm U., Fast I., Hennemuth B., Keuler K., Keup-Thiel E., Lautenschlager M., Legutke S., Radtke K., Rockel B., Schubert M., Will A., Woldt M., Ammer C. 2008. Ensemble simulations over Europe with the regional climate model CLM forced with IPCC AR4 global scenarios (M & D Technical Report 3). Hamburg
- Huntley B., Berry P., Cramer W., McDonald A. 1995. Modelling present and potential future ranges of some european higher plants using climate response surfaces. *J. Biogeogr.* 22, 967–1001
- IPCC (Intergovernmental Panel on Climate Change) 2000. Emissions scenarios: Summary for policymakers – A special report of IPCC Working Group III. <https://www.ipcc.ch/pdf/special-reports/spm/sres-en.pdf> (accessed 2015/12/26)
- IPCC (Intergovernmental Panel on Climate Change) 2013. IPCC Fifth Assessment Report (AR5), WGII: Impacts, adaption and vulnerability. <http://www.ipcc.ch/report/ar5/wg2/> (accessed 2015/12/26)
- Jacob D., Götte, H., Kotlarski S., Lorenz P., Sieck K. 2008. Klimaauwirkungen und Anpassung in Deutschland – Phase I: Erstellung regionaler Klimaszenarien für Deutschland (No. 204 41 138 UBA-FB 000969). <http://www.umweltbundesamt.de/sites/default/files/medien/publikation/long/3513.pdf> (accessed 2015/12/26)
- Jansen M., Döring C., Ahrends B., Bolte A., Czajkowski T., Panferov O., Albert M., Spellmann H., Nagel J., Lemme H., Habermann M., Staupendahl K., Möhring B., Böcher M., Storch S., Krott M., Nuske R., Thiele J., Nieschulze J., Saborowski J., Beese F. 2008. Anpassungsstrategien für eine nachhaltige Waldbewirtschaftung unter sich wandelnden Klimabedingungen – Entwicklung eines Entscheidungsunterstützungssystems „Wald und Klimawandel“ (DSS-WuK). *Forstarchiv* 79, 131–142
- Jarvis A., Reuter H., Nelson A., Guevara E. 2008. Hole-filled Seamless SRTM data V4, International Centre for Tropical Agriculture (CIAT). <http://srtm.csi.cgiar.org> (accessed 2015/12/26)
- Jesse D. 2013. YAML CSS Framework — for truly flexible, accessible and responsive websites. <http://www.yaml.de/> (accessed 2015/09/12)
- Kirilenko A.P., Sedjo R.A. 2007. Climate change impacts on forestry. *Proc. Natl. Acad. Sci.* 104, 19697–19702
- Kölling C., 2007. Klimahüllen für 27 Waldbaumarten. *AFZ-DerWald* 62 (23), 1242–1245
- Lautenschlager M., Keuler K., Wunram C., Keup-Thiel E., Schubert M., Will A., Rockel B., Boehm, U. 2009a. Climate simulation with CLM, climate of the 20th century run No.1, data stream 3: European Region MPI-M/MaD. World Data Center for Climate. http://cera-www.dkrz.de/WDCC/ui/Compact.jsp?acronym=CLM_C20_1_D3 (accessed 2015/12/26)
- Lautenschlager M., Keuler K., Wunram C., Keup-Thiel E., Schubert M., Will A., Rockel B., Boehm U. 2009b. Climate simulation with CLM, scenario A1B run No.1, data stream 3: European Region MPI-M/MaD. World Data Center for Climate. http://cera-www.dkrz.de/WDCC/ui/Compact.jsp?acronym=CLM_A1B_1_D3 (accessed 2015/12/26)
- Lindau R., Simmer C. 2012. On correcting precipitation as simulated by the regional climate model COSMO-CLM with daily rain gauge observations. *Meteorol. Atmospheric Phys.* 119, 31–42
- Lindner M., Sohngen B., Joyce L.A., Price D.T., Bernier P.Y., Karjalainen T. 2002. Integrated forestry assessments for climate change impacts. *For. Ecol. Manage.* 162, 117–136
- Lindner M., Maroschek, M., Netherer, S., Kremer, A., Barbati, A., Garcia-Gonzalo, J., Seidl, R., Delzon, S., Corona, P., Kolström, M., Lexer, M.J.,

- Marchetti, M. 2010. Climate change impacts, adaptive capacity, and vulnerability of European forest ecosystems. *For. Ecol. Manage.* 259, 698–709
- Lindner M., Fitzgerald J.B., Zimmermann N.E., Reyser C., Delzon S., van der Maaten E., Schelhaas M.-J., Lasch P., Eggers J., van der Maaten-Theunissen M., Suckow F., Psomas A., Poulter B., Hanewinkel M., 2014. Climate change and European forests: What do we know, what are the uncertainties, and what are the implications for forest management? *J. Environ. Manage.* 146, 69–83
- Lower Saxony Ministry of Food, Agriculture and Consumer Protection 2014. Waldbewertungsrichtlinien (WBR 2014) Tabelle 1.1 Zuordnung der Baumarten. https://www.landesforsten.de/fileadmin/doku/Produkte_u_Service/Waldbewertung/Tab_1_Zuordnung_D7-15.pdf (accessed 2015/12/26)
- Lynch T., Gregor S. 2004. User participation in decision support systems development: Influencing system outcomes. *Eur. J. Inf. Syst.* 13, 286–301
- Lynch T., Gregor S., Midmore D. 2000. Intelligent support systems in agriculture: how can we do better? *Aust. J. Exp. Agric.* 40, 609–620
- Mono Project 2015. Home | Mono. <http://www.mono-project.com/> (accessed 2015/09/12).
- Mudelsee M., Chirila D., Deutschländer T., Döring C., Haerter J., Hagemann S., Hoffmann H., Jacob D., Krahe P., Lohmann G., Moseley C., Nilson E., Panferov O., Rath T., Tinz B. 2010. Climate Model Bias Correction und die deutsche Anpassungsstrategie. *Mitteilungen DMG* 03/2010, 2–7
- Munich Re 2011. Topics Geo: Natural catastrophes 2010 – Analyses, assessments, positions. Munich. http://www.munichre.com/site/touch-publications/get/documents_E-595398459/mr/assetpool.shared/Documents/5_Touch/_Publications/302-06734_de.pdf (accessed 2015/12/27)
- Murdock T.Q., Taylor S.W., Flower A., Mehlenbacher A., Montenegro A., Zwiers F.W., Alfaro R., Spittlehouse D.L. 2013. Pest outbreak distribution and forest management impacts in a changing climate in British Columbia. *Environ. Sci. Policy* 26, 75–89
- Nagel J., 2011. Waldwachstumssimulation mit dem Softwarepaket Tree-GrOSS – Neuerungen, Erweiterungsmöglichkeiten und Qualitätsmanagement. In: Römisch K., Nothdurft A., Wunn U. (Hrsg.) Tagungsband Jahrestagung Sektion Forstliche Biometrie und Informatik im DVFFA und der AG Ökologie und Umwelt in der Internationalen biometrischen Gesellschaft. Die Grüne Reihe. Freiburg, 174–178
- Nah F.F.-H. 2004. A study on tolerable waiting time: How long are web users willing to wait? *Behav. Inf. Technol.* 23, 153–163
- OGC. 2010. Simple Feature Access – Part 2: SQL Option | OGC. <http://www.opengeospatial.org/standards/sfs> (accessed 2015/09/23)
- OGC. 2015. Web Map Service | OGC. <http://www.opengeospatial.org/standards/wms> (accessed 2015/09/12)
- OpenLayers Contributors 2015. OpenLayers 3 – Welcome. <http://openlayers.org/> (accessed 2015/09/12)
- Packalen T., Marques A., Rasinmäki J., Rosset C., Mounir F., Rodriguez L., Nobre S. 2013. Review: A brief overview of forest management decision support systems (FMDSS) listed in the FORSYS wiki. *For. Syst.* 22, 263–269
- Panferov O., Doering C., Rauch E., Sogachev A., Ahrends B. 2009. Feedbacks of windthrow for Norway spruce and Scots pine stands under changing climate. *Environ. Res. Lett.* 4, 045019
- Python Software Foundation 2015. Welcome to Python.org. <https://www.python.org/> (accessed 2015/09/12)
- Quick J. 2009. GChartWrapper 0.9 : Python Package Index. <https://pypi.python.org/pypi/GChartWrapper> (accessed 2015/09/12)
- Refractions Research 2015. PostGIS : Home. <http://postgis.refractions.net/> (accessed 2015/09/12)
- ReportLab Inc. 2015. ReportLab open-source PDF Toolkit – ReportLab.com. <http://www.reportlab.com/opensource/> (accessed 2015/09/12)
- Reynolds K.M. 2005. Integrated decision support for sustainable forest management in the United States: Fact or fiction? *Comput. Electron. Agric.* 49, 6–23
- Richter A., Adler G., Fahrak M., Eckelmann W. 2007. Erläuterungen zur nutzungsdifferenzierten Bodenübersichtskarte der Bundesrepublik Deutschland im Maßstab 1:1.000.000 (BÜK 1000 N, Version 2.3). Bundesanstalt für Geowissenschaften und Rohstoffe (Hrsg.), Hannover
- Schober R. 1995. Ertragstafeln wichtiger Baumarten. 4. Aufl. J.D. Sauerländer's, Frankfurt a. M.
- Sencha Inc. 2015a. Ext JS – MVC/MVVM JavaScript Framework for Cross-platform Web Apps | Sencha. <https://www.sencha.com/products/extjs/#overview> (accessed 2015/09/12)
- Sencha Inc. 2015b. GPL Licensing | Sencha. <https://www.sencha.com/legal/GPL/> (accessed 2015/09/12)
- Spekat A., Enke W., Krienkamp, F. 2007. Neuentwicklung von regional hoch aufgelösten Wetterlagen für Deutschland und Bereitstellung regionaler Klimaszenarios auf der Basis von globalen Klimasimulationen mit dem Regionalisierungsmodell WETTREG auf der Basis von globalen Klimasimulationen mit ECHAM5/MPI-OM T63L31 2010 bis 2100 für die SRESSzenarios B1, A1B und A2 (FKZ 204 41 138). Umweltbundesamt, Dessau
- Sprague R.H. 1980. A framework for the development of decision support systems. *MIS Q.* 4, 1–25
- Standard C++ Foundation 2015. Standard C++. <https://isocpp.org/> (accessed 2015/09/12)
- Staupendahl K. 2011. Modellierung der Überlebenswahrscheinlichkeit von Waldbeständen mithilfe der neu parametrisierten Weibull-Funktion. *Forstarchiv* 82, 10–19
- SWIG 2015. Simplified wrapper and interface generator. <http://www.swig.org/> (accessed 2015/09/12)
- The PostgreSQL Global Development Group 2015. PostgreSQL: The world's most advanced open source database. <http://www.postgresql.org/> (accessed 2015/09/12)
- The Scipy community 2014. F2PY Users Guide and Reference Manual – NumPy v1.10.dev-19c3cf1 Manual. <http://docs.scipy.org/doc/numpy-dev/f2py/> (accessed 2015/09/12)
- Thiele J., Nuske R., Ahrends B., Saborowski J., 2009. Entwicklung eines Entscheidungsunterstützungssystems für die Waldbewirtschaftung unter sich ändernden Klimabedingungen. In: Mahammadzadeh M., Biebler H., Bardt H. (Hrsg.), Klimaschutz und Anpassung an die Klimafolgen. Institut der deutschen Wirtschaft Köln Medien GmbH, Köln, 303–310
- Thiele J., Nuske R., Ahrends B., Jansen M., Saborowski J., Albert M., Nagel J., Junghans U., Czajkowski T., Panferov O., Staupendahl K., Spellmann H., Habermann M., Bolte A., Möhring B. 2010. Abschätzung der abiotischen und biotischen Risiken für die Fichte im Arnberger Wald unter veränderten Klimabedingungen – DSS-WuK Arnberger Wald (Abschlussbericht). Universität Göttingen, Göttingen (unpublished)
- Vacik H., Torresan C., Hujala T., Khadka C., Reynolds K. 2013. The role of knowledge management tools in supporting sustainable forest management. *For. Syst.* 22, 442–455
- Wenkel K.-O., Berg M., Mirschel W., Wieland R., Nendel C., Köstner B. 2013. LandCaRe DSS – an interactive decision support system for climate change impact assessment and the analysis of potential agricultural land use adaptation strategies. *J. Environ. Manage.* 127, S168–S183
- Wollborn P., Böckmann T. 1998. Ein praktikables Modell zur Strukturierung des Vorrates aus Ertragstafelschätzung. *Forst Holz* 53, 547–550
- Yousefpour R., Hanewinkel M. 2015. Forestry professionals' perceptions of climate change, impacts and adaptation strategies for forests in south-west Germany. *Clim. Change* 130, 273–286