Capsis: an open software framework and community for forest growth modelling

Dufour-Kowalski, Courbaud, Dreyfus, Céline Meredieu, Coligny

ORIGINAL PAPER

Capsis: an open software framework and community for forest growth modelling

Samuel Dufour-Kowalski · Benoît Courbaud ·
Philippe Dreyfus · Céline Meredieu ·
François de Coligny

Abstract
· *Context* Forest scientists build models to simulate stand growth and forests dynamics. Dedicated computer tools are often developed to implement these models in order to run silvicultural scenarios and explore simulation results.
· *Aims* Our objective was to encourage software reuse and simplify model implementation.
· *Methods* The scheme was to develop a framework and methodology allowing to simplify the implementation, integration, simulation and comparison of forest models by providing a set of common and standard tools.
· *Results* Capsis provides an open and modular software architecture based on various components, allowing to run forest growth simulations and display the results. The benefits of this framework are shown with the Samsara2 model, an individual-based and spatialised tree model. Capsis has been used successfully in many similar projects. In addition, the Capsis methodology defines how developers, modellers and end-users may interact.
· *Conclusion* The Capsis framework facilitates collaborative and shared software development. Moreover, it is a powerful way to support scientific animation in the frame of forest science.

Keywords Forestry · Silviculture · Software architecture · Modelling framework

S. Dufour-Kowalski · F. de Coligny (✉)
INRA, UMR931 AMAP,
Botany and Computational Plant Architecture,
TA A-51/PS2, Boulevard de la Lironde,
34398 Montpellier Cedex 5, France
e-mail: coligny@cirad.fr

B. Courbaud
CEMAGREF, Mountain Ecosystems Research Unit,
2 rue de la Papeterie,
38402 Saint Martin d'Hères, France

P. Dreyfus
INRA, UR629 URFM, Écologie des Forêts Méditerranéennes,
Site Agroparc, Domaine Saint Paul,
84914 Avignon Cedex 9, France

C. Meredieu
INRA, UMR1202 BIOGECO,
69 route d'Arcachon,
33612 Cestas Cedex, France

1 Introduction

Forest managers plan and organise silvicultural treatments to address particular management objectives in terms of yield, economics, diversity, protection or conservation. Simulation models of forest growth and dynamics can help foresters in their decision-making process by predicting forest stands natural evolution and the consequences of silvicultural treatments. Foresters have used yield tables and stand density management diagrams for long to estimate wood production. Computer simulations allow today to make virtual experiments to explore the effects of a wide range of management scenarios on diverse ecosystem services, diverse types of forests and environment conditions. To be fully exploited, forest dynamics models must be integrated in simulation tools and decision support systems (Muys et al. 2011). While computational power increases, forest growth models concurrently become more and more complex. Their conception and their implementation follow the same tendency. To implement a new

INRA ⁂ Springer

model, forest growth modellers have to face the following challenges: (1) tackle model development complexity, coding and efficiency, (2) make the models usable by other scientists or forest managers and (3) be able to compare various modelling strategies and examine model outputs and simulation results easily.

In environmental sciences, various approaches have been carried out to implement simulation models (Table 1). The most common approach in forest growth simulation consists in building a specific computer programme around a given model. Generally, it can support various parameter sets and can be adapted to various contexts by changing model inputs. A modeller has complete implementation control and can choose freely a data structure. However, such an approach is costly, requires extensive development and a good cooperation between modellers and computer scientists. It also needs investment in ergonomics, documentation and training. The resulting code is specific to the paradigm of one model. It may be difficult to reuse for the implementation of another model and this approach does not address development cost reduction. We can find in the literature a lot of examples of this implementation strategy: SILVA (Pretzsch et al. 2002), SORTIE/BC (Coates et al. 2003), Prognaus (Monserud and Sterba 1996), Sexi (Vincent and de Foresta 1998), Simcop (Ottorini 1991), FVS (Lacerte et al. 2006), etc.

To avoid these problems, modellers can use modelling frameworks or platforms. Usually, a framework is designed to address common modelling and engineering issues and intends to reduce the investment needed to develop a model. A framework provides the following features (Argent 2004): (1) a development environment making model integration, implementation and modification easier; (2) a set of domain-specific and reusable libraries and data structures; (3) some generic and reusable components for mathematics, visualisation or data manipulation; (4) a model execution system based on a simulation paradigm assisting with model execution and analysis processing; (5) a user interface to configure the model and run the simulations; and (6) a documentation and a test system to ensure software quality.

This kind of solution is widely used in environmental modelling but not particularly in forest growth dynamics modelling. "Choosing a framework (for a specific modelling problem) is an unclear process, driven by many factors not necessarily related to the particular advantages or disadvantages of a software package" (Argent et al. 2006). We indicate hereafter some of the frameworks used in environmental modelling and point their limitations, considering the development and the diffusion of forestry models. We must consider the following points: (1) not all forest models are based on the same assumptions (Porté and Bartelink 2002): models can be mono-specific at stand level, multi-species at tree level, distance independent or spatialised, etc. This heterogeneity requires a framework supporting different modelling and simulation paradigms; (2) model implementation is a difficult task for scientists. Modellers should be able to create easily new models or update previous ones, to share libraries and forestry objects and to reuse code in different models, leading to the creation of a modelling community; (3) since models are used by forest managers in some cases, ease of use and distribution aspects should be taken into account.

Scientific software like MATLAB (http://www.mathworks.com/) or R (http://www.r-project.org/) offer high-level languages and generic scientific libraries very efficient to build exploratory models with simple structures but lead to a specific programme for each problem. They are efficient for prototyping but they are not suitable for complex model implementations and make models difficult to distribute to end-users. Domain-specific languages like OCELET (Degenne et al. 2009) are also not suitable because their functionalities are dedicated especially to their domain and usually do not provide any user interface.

For non-programmers, frameworks like STELLA (http://www.iseesystems.com/), MODELMAKER (http://www.modelkinetix.com/) or Simile (Muetzelfeldt and Massheder 2003) provide graphic modelling environments dedicated to the simulation of dynamic systems based on compartment flows or individual-based modelling. However, visual programming is not suitable for complex model implementations. It results in many prototypes difficult to modify or to reuse in different contexts. These frameworks are not adapted for model distribution because of portability and licensing problems. They support only specific modelling paradigms and are not sufficient to cover all forestry modelling aspects. The same limitations appear with ECOSIM (Lorek and Sonnenschein 1999) which is specific to individual-based modelling or COINS (Roxburgh and Davies 2006) which is built on a grid and cells based space representation.

Many frameworks are dedicated to model linking. The OpenMI project (Moore and Tindall 2005) is an initiative to define a standard API for model linking and interaction in the hydrology domain. With the same principle, APSIM (Keating et al. 2003) and MODCOM (Hillyer et al. 2003) were designed to run simulations of agricultural systems. The TIME framework (Rahman et al. 2003) is another example of time dependent models integration framework. The properties and the capabilities of the models are described with a set of metadata, allowing the use of dynamic and generic processing tools like a model management system, an automatic input/output processor or a user interface generator. In addition some frameworks provide visual model interconnection. For example, VLE (Quesnel et al. 2009) allows to interconnect heterogeneous

Table 1 Modelling software/frameworks in order of citation

Software/ framework	Development interface[a]	Multi model integration[b]	Multi model linking[c]	End user interface[d]	Application domain[e]	Main modelling paradigm[f]
SILVA	Not extensible	No	No	Yes	Forestry	Distance-dependent tree model
SORTIE	C++/JAVA API	No	No	Yes	Forestry	Distance-dependent tree model
Prognaus	Not extensible	No	No	Yes	Forestry	Distance-independent tree model
Sexi	Not extensible	No	No	Yes	Forestry	Distance-dependent tree model
Simcop	Not extensible	No	No	No	Forestry	Distance-dependent tree model
FVS	SAS API	No	No	Yes	Forestry	Distance-independent tree model
MATLAB	Own language only	Yes	Yes	No	Numerical computing	Procedural
R	Own language only	Yes	Yes	No	Statistics	Procedural
OCELET	Own language only	Yes	No	No	Landscape modelling	Entity/relation/scenario
STELLA	Visual programming only	Yes	No	No	Not domain specific	Compartment flows
MODELMAKER	Visual programming only	Yes	No	No	Not domain specific	Compartment flows
Simile	Visual programming only	Yes	No	No	Agro-ecology	Compartment flows
ECOSIM	Own language and visual programming	Yes	No	No	Individual-based ecological modelling	Individual based
COINS	Delphi component API	Yes	No	Yes	Space-based ecological modelling	Grid cell/time scheduling
OpenMI	.NET and JAVA component API and visual programming	Yes	Yes	No	Not domain specific/ used in hydrology	Time scheduling
APSIM	.NET component API	Yes	Yes	No	Agriculture	Message passing system
MODCOM	.NET and COM/DCOM component API	Yes	Yes	No	Agro-ecology	Time scheduling
TIME	.NET component API	Yes	Yes	Yes	Not domain specific/ used in hydrology	Time scheduling
VLE	C++ component API and visual programming	Yes	Yes	No	Not domain specific	DEVS
OPENALEA	Python component API and visual programming	Yes	Yes	No	Plant modelling	Dataflow
SIMO	Python component API and conf XML	Yes	Yes	No	Forestry	Time scheduling
CAPSIS	JAVA component API	Yes	No	Yes	Forestry	Scenario

[a] Method to implement a new model or behaviour

[b] The modeller can create different models with the same tool. "No" includes the case when a single model is adaptable/parameterisable, meaning a variant (same processes) can be built/calibrated for various species/situations

[c] Aims at integrating and aggregating easily different autonomous models, possibly implemented in another software/language, to create a new one. Generally, such a framework provides an API to implement a model and a configuration tool to compose the simulation

[d] Integrated models can be configured, executed, explored in a dedicated user-friendly interface without any knownledge on the models implementation. End-user interfaces are different than model creation interface

[e] Domain for which the framework has been created. Usually, a domain-specific tool comes with dedicated software libraries to simplify the developement of models in the domain

[f] Main modelling principle. This can be an advantage because modellers do not have to care about some aspect of the model implementation, but it also can be a limitation because models which do not fit are hard to implement

models and simulate complex dynamical systems using a discrete event based formalism. In the domain of plant modelling, OpenAlea (Pradal et al. 2008) provides an environment making model development easy through the graphical assembly of different components using a data-flow paradigm. At the forest scale, the SIMO framework (Rasinmki et al. 2009) integrates forest growth models by using declarative programming. Simulation parameters are described in a file and launched in a dedicated runtime environment. All these frameworks focus on the combina-

tion of different component models sharing a similar paradigm. However, they do not provide a common development tool for models with different space and time representations and do not assess the need of sharing out domain-specific objects and libraries. Finally, they usually do not consider model distribution to final users.

In this paper, we present the Capsis project, an integration framework dedicated to forest growth and yield modelling. Capsis has been designed to address the modelling challenges specific to this domain. Thanks to its open architecture, its domain specificity and generic data structure, Capsis can host all kinds of forestry related models. These models share common and generic forestry objects like trees or stand descriptions that make model development and comparison easier. The framework provides an execution system to run models and to simulate forest management and silvicultural treatments. The user-friendly graphical interface makes the models easy to use by forest managers. A specificity of Capsis comes from its role in organising collaborations within a community of forest modellers. A set of rules have been defined to encourage collaborative development, model sharing and code reuse (see the *Capsis charter* below in Section 3). The originality of the project comes from this unique combination of open architecture and working community. In this paper, we first describe the Capsis software framework, its architecture and its various components. Then, we show how the Capsis community is organised and what are the benefits of such a community. Finally, we focus on one of the integrated models and discuss the advantages in using such a framework for forestry modelling projects.

2 Capsis framework and methodology

2.1 An open software architecture

Capsis is based on an open software architecture making model development and integration easier by providing modular and reusable components (Fig. 1). Since its version 4, Capsis has been built with the Java environment which provides an extended software development kit and ensures the framework to be portable on different computer operating systems without any effort. The framework has been designed to support a large panel of models from the forestry domain and tries to be non restrictive on a particular paradigm. The architecture is said open because it can be extended by adding new models and shared components. We present hereafter the main concepts used in Capsis.

The *Capsis Kernel* provides a model execution system. It doesn't contain any model but proposes fundamental classes like temporal and spatial data structures and
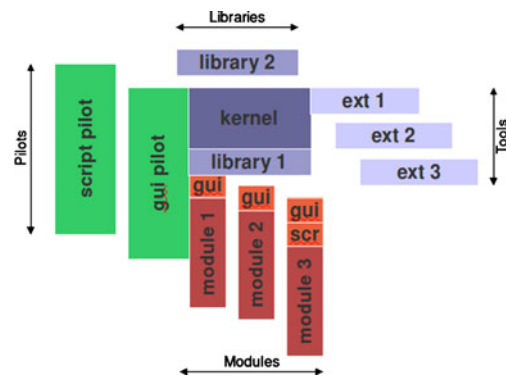

Fig. 1 The Capsis open software architecture. The kernel and libraries contain the simulation framework and the forestry domain-specific functions. The modules are pluggable packages containing the models. The pilots represent the various user interfaces (e.g. graphical user interface and script). The extensions (*ext*) are specific tools possibly shared between modules

functions to create and manage simulation projections. It also manages simulation execution. Each simulation is represented by a *Project* composed of different evolution *Steps*. Each step contains the state of a scene at a particular time (age or date). For forest models, the scene is generally a stand but can also be single tree, a set of trees, a set of stands covering a forest, a watershed or a landscape. To explore alternative silvicultural scenarios, a project can have multiple branches (Fig. 2). Each branch corresponds to a specific scenario and can be compared with the others.

The *Libraries* are just on top of the kernel. They propose common functions and objects for forestry modelling. This is the domain-specific part of the framework. These libraries provide forestry data structures like stands, trees and ground. Trees always have an age and a diameter at breast height but may or may not have additional characteristics such as location, crown dimension, etc. These template structures are reused by modellers when they create their own module. Capsis libraries also provide additional forestry specific objects and methods such as algorithms for tree location setting, genetics, economics, or biomechanics. These libraries are generic and can be reused in all models without restriction.

The *Modules* (Table 2) are the most important components in Capsis' architecture: they embed forestry models. Each model is implemented in a separate package called a module. Each module is autonomous and plugged into the framework. It can be removed or added without interfering with other modules. A module relies on the common data structures provided by the kernel and some libraries if necessary but is not limited to them. By using object-oriented programming concepts, especially inheritance (Sequeira et al. 1997), each module can provide its own forest data structure, objects and methods. The simulation logic of the model is also provided by the module itself. A module must provide specific entry points or methods that can be called by the kernel. These
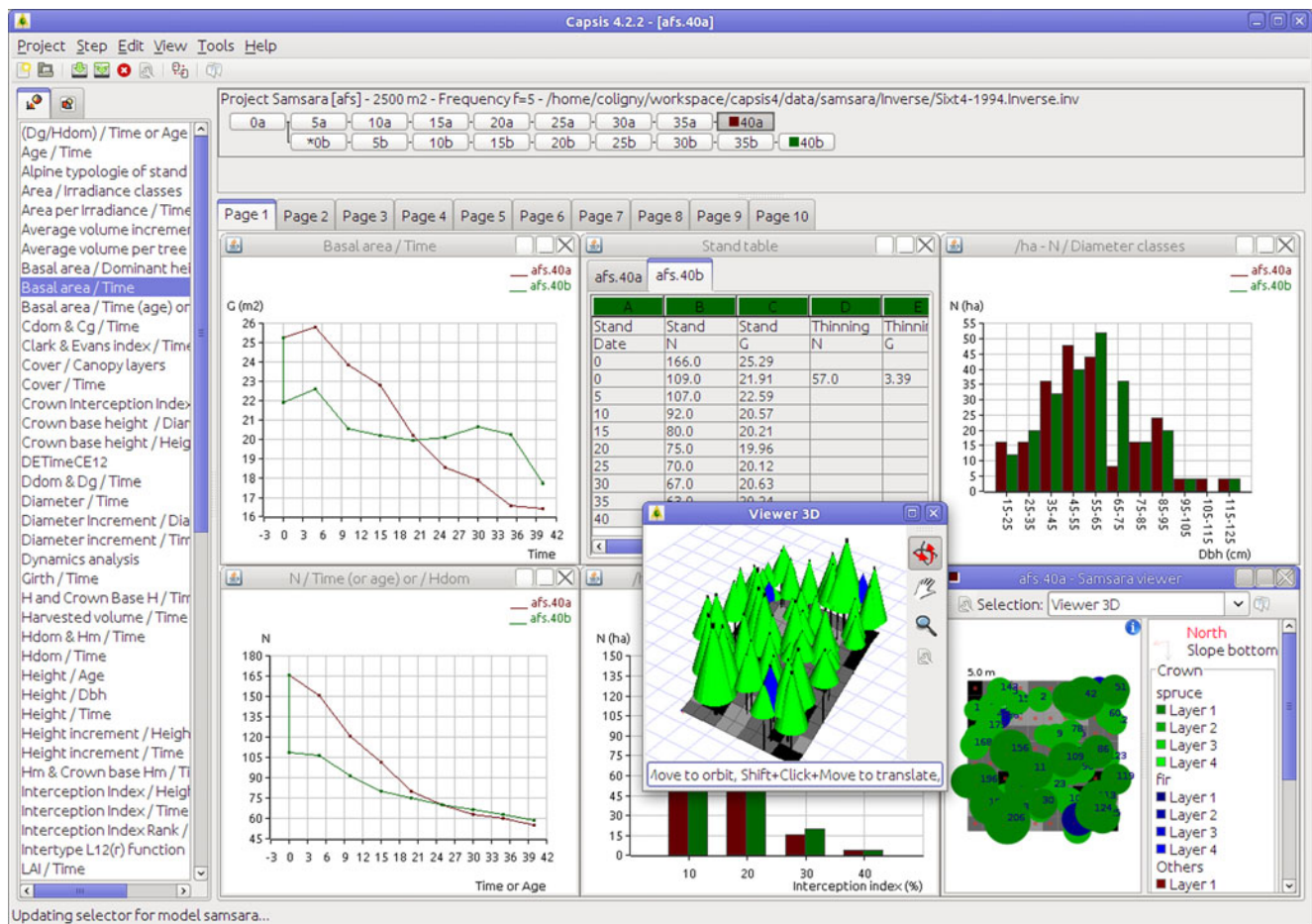
Fig. 2 Capsis main window. At the *top*, the project manager shows the simulations with various silvicultural scenarios. The simulations results are shown *below* by extensions like plots over time or 3D reconstructions

entry points define initialisation and evolution procedures. This approach is very flexible and allows implement a wide variety of models.

The *Extensions* give Capsis a large expandability and adaptability. An extension is a kind of plugin : a reusable and autonomous piece of software which can be added to the system dynamically (Papajorgji 2005). In Capsis, extensions are usually small and optional elements which can be used in different modules and contexts. Since extensions are dynamic, it is easy to add a new extension into the framework. Various families of extensions are available (Fig. 2): *interventions*, including mostly stand thinning tools but also possibly any other action changing the state of the stand such as fertilizing for example, *imports* from and *exports* to various file formats, *data charts* and *stand visualisations* in 2D or 3D, *data analysis tools* for economics or carbon storage evaluation, etc. When a modeller provides a new extension, for instance a thinning strategy, this extension can be made directly available for other modules. This mechanism is very useful to share tools between models and to reuse common utilities.

The *Pilots* are different user interfaces in the Capsis architecture. Multiple pilots target different kinds of users

with their different skills and needs such as forest managers, scientists or computer engineers. The *main pilot* provides a rich and user-friendly graphical interface (Fig. 2). It can be used to load a model, to run various scenarios and to display or export results in a unique interface. It is well adapted for simple experimentations and model exploration by users lacking computer programming skills. The framework also provides *script pilots* useful to run complex simulations, to do sensitivity analyses or to use the models outside the user interface. When using these script pilots, a user can define a simulation plan with a programming language like Java or Groovy (http://groovy. codehaus.org) or with a descriptive language like XML.

2.2 Forest growth and yield models integration

The Capsis framework has been designed specifically to integrate forestry and silviculture models with as little limitation as possible regarding model types. In addition to the tools and libraries described previously, the framework defines implementation conventions and design patterns that limit the work needed to write a model and promote

Table 2 Modules in Capsis: scope and main characteristics (see also Online resource 1)

Climatic zone	Stand composition[a]	Stand structure	Simulation levels[b]	Main processes[c]	Specific features[d]	Species[a]	Module name
Temperate	Pure	Regular	Average tree, stand	Gr (DI), Mo, Re, Di		*Cytisus scoparius*	Cytisus
			Average tree, stand	Gr (DI), Mo	Phy	*Abies alba, Fagus sylvatica*	Dynaclim
			Average tree, stand	Gr (DI), Mo	Phy, Ge	*A. alba, F. sylvatica*	PhysioDemoGenetics
			Average tree, stand	Gr (DI)	Th	*Pinus pinaster*	Lemoine
			Average tree, stand	Gr (DI)		Many species	Natura
			Average tree, stand	Gr (DI)		*Eucalypt* ssp., *Populus* ssp.	Regix
			Tree, stand	Gr (DI), Mo	Th	*A. alba*	Abial
			Tree, stand	Gr (DI), Mo		*Picea alba*	Afocelpa
			Tree, stand	Gr (DI), Mo		*P. pinaster*	Afocelpp
			Tree, stand	Gr (DI), Mo		*Cedrus atlantica*	CA1
			Tree, stand	Gr (DI), Mo		*Pseudotsuga menziensii, Larix* spp., *Picea abies*	Douglas
			Tree, stand	Gr (DI), Mo	Th	*Quercus petraea, F. sylvatica*	Fagacees
			Tree, stand	Gr (DI), Mo	Br	*Pinus halepensis*	Fiesta/NRG
			Tree, stand	Gr (DI), Mo		*Pinus nigra laricio*	Laricio
			Tree, stand	Gr (DI), Mo		*Abies balsamea*	Matapedia
			Tree, stand	Gr (M/D), Mo	Th	*P. pinaster*	ModisPinaster
			Tree, stand	Gr (DI)	Br	*Pinus radiata*	NZ1
			Tree, stand	Gr (DI), Mo	Br	*Pinus nigra nigra*	PNN/Pnn2
			Tree, stand	Gr (DI)		*P. pinaster*	PP3
			Tree, stand	Gr (DI), Mo		*Q. petraea*	QS1
			Tree, stand	Gr (DI), Mo		*Pinus sylestris*	Sylvestris
			Average tree, stand, forest	Gr (DI)	Th	2 virtual species	Mustard
			Tree, stand, forest	Gr (M/D)	Th	Many species	IFNCA
			Tree, stand, forest	Gr (DI), Mo, Re, Di	Ge	*C. atlantica*	Luberon
			Tree, stand, forest	Gr (DI), Mo		*Pinus contorta*	MPB
			Tree, stand, forest	Gr (DD), Mo, Re		Many species	Presage
			Tree, stand, forest	Gr (DI), Mo, Re	Ge	*Q. petraea*	Quercus
			Tree, stand, region	Gr (DI), Mo		*Pinus sylvestris, Quercus* sp.	Simmen
			Tree, stand, region	Gr (DI)	Th	*P. pinaster*	Sylvogene
		Irregular	Tree, stand	Mo, Re, Di	Ge	*Prunus mahaleb*	Prunus
			Tree, stand, forest	Re, Di		*C. atlantica*	Abccedrus
			Tree, stand, forest	Gr (DI), Mo, Re, Di		*A. alba*	Migration
			Tree, stand, forest	Gr (DD), Mo, Re		*P. abies*	Mountain
	Mixed	Regular	Average tree, stand	Gr (DI)		*P. abies, A. alba*	Melies
			Tree, stand	Gr (DI), Mo		*Q. petraea, F. sylvatica, P. sylvestris*	Fagacees–Sylvestris
			Tree, stand	Gr (DD)		*Q. petraea, F. sylvatica, P. sylvestris*	Oakpine1
			Tree, stand	Gr (DI)		*Q. petraea, F. sylvatica, P. sylvestris*	Oakpine2
			Tree, stand	Gr (M/D), Mo	Th	*Larix olgensis, Picea jezoensis, Abies nephrolepis*	LSFMGM
			Tree, stand, forest	Gr (DI), Mo		*Pinus banksiana, Picea glauca, Picea mariana*	JackPine
			Tree, stand, region	Re		Many species	Succes

Table 2 (continued)

Climatic zone	Stand composition[a]	Stand structure	Simulation levels[b]	Main processes[c]	Specific features[d]	Species[a]	Module name
		Irregular	Tree, stand	Gr (DD)		*Populus* spp., *Juglans nigra x regia, Prunus avium*, crops	Hi-sAFe
			Tree, stand	Re		Broadleaved species of north-eastern France	Regelight
			Tree, stand	Gr (DD), Mo, Re		*Quercus* sp., *P. sylvestris*	RReShar
			Tree, stand	Gr (DD), Mo, Re		*Acer saccharum, Betula alleghaniensis, Fagus grandifolia*, other broadleaves	Samare
			Tree, stand	Gr (DD), Mo, Re		*P. abies, A. alba*, other species	Samsara
			Tree, stand, forest	Gr (DI), Mo, Re, Di	Ge	*Sorbus torminalis, Q. petraea, F. sylvatica*	Alisier
			Tree, stand, forest	Gr (DI), Mo, Re, Di	Ge	*F. sylvatica*, virtual species	TranspopRege
			Tree, stand, forest/ landscape	Gr (DI), Mo, Re, Di	Ge, Th	*A. alba, F. sylvatica, P. nigra nigra, P. sylvestris, Pinus uncinata*	Ventoux/VentouG
			Tree, stand, forest	Gr (DI), Mo	Th	*P. halepensis, P. sylvestris, P. pinea, P. pinaster, P. nigra laricio, P. nigra nigra*	FireParadox
			Tree, stand, region	Gr (DI), Mo, Re	Th	Many species	Artemis
Subtropical	Pure	Regular	Average tree, stand	Gr (DI)	Th	*Pinus massoniana*	ISGM
Tropical	Pure	Irregular	Tree, stand	Gr (DI)		*Eucalyptus* spp.	Eucalypt
			Tree, stand	Gr (DD)		*Rhizophora* spp.	Mangrove
			Tree, stand	Gr (DD), Re		*Avicennia* spp., *Rhizophora* spp.	Paletuviers
			Tree, stand, forest	Gr (DI), Re	Ge	*Vitellaria paradoxa*	Karite
	Mixed	Irregular	Tree, stand	Gr (DD), Mo, Re		Many species	Selva
			Tree, stand	Gr (DD), Mo, Re	Br	Many species	Stretch

[a] "Pure" with a list of species means that the model either works with a pure stand of one of these species at a time or can simulate concurrently several pure stands of different species in the same forest

[b] Scale levels at which the model works or the module gives outputs (with more or less details depending on the model)

[c] Forest dynamics processes: growth (diameter and, possibly, height) (Gr), mortality (Mo), recruitment/regeneration (Re), dispersal (seeds/ seedlings) (Di). Details for growth submodel: distance-independent tree growth (Gr (DI)), distance-dependent tree growth (Gr (DD)), growth submodel relying on a transition matrix or a diameter distribution curve (Gr (M/D))

[d] Genetics (Ge), ecophysiological processes (Phy), branching model (Br), automated thinning (in addition to Capsis' interactive intervention tools) (Th)

clean design and modularity (Nguyen et al. 2004). As an illustration, the "model-view pattern" clearly separates the application data definition and management (the *model*) and the corresponding graphical user interface representation and user interaction handling (the *view*). It is widely used to ensure coherence in the code and encourage model reuse. However, model implementation is not limited to the provided structures and conventions. If a particular model does not fit with the default structures, it is still possible to define a specific behaviour using the object-oriented principles. To simplify integration, rapid development techniques are proposed to the modellers. New models are created easily by using a model template and by specifying

metadata like the name, authors and model type. The skeleton generated is then completed with specific code. Rahman et al. (2004) have shown that graphical user interfaces can be automatically generated by inspecting model parameters at runtime. This option is available in Capsis, thanks to the java language reflexivity. However, dedicated interfaces for specific needs or complex user interactions may also be provided by the modellers themselves.

Even if a modeller is released from a lot of technical issues, he (she) should provide the core functions of the model: an initialisation method to build the initial scene from model input parameters and data and an evolution

method to compute the scene evolution between two simulation steps. The other external actions are done by reusable *extensions* as defined previously. Silvicultural interventions, specific viewers or other tools are implemented separately using this extension mechanism. When a modeller adds a new and original thinning algorithm or a carbon storage estimation feature in the framework, it can be applied to any other model provided that the model has a compatible data structure and a set of functions expected by the tool. In most cases, Capsis can answer forestry modelling problems. However, there are some cases where the use of a model or a tool implemented in another modelling framework is desirable. Capsis is able to use various inter framework communication techniques as described by Argent and Rizzoli (2004). For instance, Capsis models use external agronomic models like STICS (Brisson et al. 2002; Table 2, Hi-sAFe) and wind risk models like ForestGALES (Gardiner and Quine 2000; Table 2, PP3) by exchanging files and running executable programmes. External networked data sources like Geographical Information Systems or remote databases can also be reached after adaptation of the given model to the GIS/database specifications. For instance, the FireParadox project (Rigolot et al. 2010; Table 2, FireParadox) uses a distant database to initialise its scenes.

Using a model A model can be used through different interfaces. Capsis provides a convenient and rich graphical user interface (Fig. 2). Outputs can be plotted in order to show evolution over time, summaries at given dates, or three dimensional views of the structure of a scene. This framework does not intend to replace software dedicated to data analysis but it provides simple tools to explore and compare simulation results. In any case, data can also be exported and analysed in external and specific tools. This graphical mode makes it possible for non-computer specialists and forest managers to run simulations easily. For a more specific usage, the user interface can be skipped and models are executed as scripts using another *pilot*. The possibility to use the models in different modes and with different interfaces provides a lot of flexibility.

3 Collaborative development

The first version of Capsis was developed in 1994 to embed a tree, distance-independent growth model for *Pinus nigra* (Table 2, PNN/Pnn2). As it appeared to be useful for silviculture simulation, the French National Forest Service (ONF) was immediately interested in having the software extended to other tree- or stand-level growth models existing at the time (for *Fagus sylvatica, Pinus pinaster, Picea abies,* etc.). This was done by adding Dynamic Link

Libraries to the software (in C++ at that time). Each DLL (equivalent to a module embedding a growth model) contained the equations for one model (Dreyfus and Bonnet 1995). Forest modellers were also interested in sharing development efforts and in transferring their models to forest managers and students. With support of ONF and FCBA (previously AFOCEL, technical centre for timber and paper production industries) and of the French ministry in charge of forests, forest modellers decided to invest together in this common software framework that should be able to integrate their models. After two versions the framework was redesigned to support a wider variety of forest models including spatially explicit models. In addition, with version 4 (in Java), a common methodology was adopted to simplify and encourage co-development.

A common organisation The Capsis methodology relies on co-development between few software developers and many forest modellers. A bottleneck often appears in a classical organisation when modellers completely rely on developers to write their models because of the limitation of development resources and because their responsibilities are not clearly defined. To promote co-development, the Capsis methodology defines the roles of all actors. Each actor works at his (her) level of competence with a great level of autonomy. The *developers* are computer engineers and are in charge of building the kernel, the common tools and the libraries (with the help of scientists for some of them) used in the models. They deal with technical problems and do not focus on modelling problems. They are also in charge of project animation, modellers training and overall support. For instance, they generally help modellers start a new project by co-designing the module structure.

The *modellers* integrate their models in the framework by themselves. They are usually forestry scientists and focus on modelling problems. They can have support from the developers at the beginning of their project. They work in autonomy by using simple development tools and languages and by following the framework structures and conventions. Modellers are responsible for their models, including validation and transfer. They can choose to distribute their models to end-users like forest managers or students.

Co-development and licensing In addition to the roles of all actors in the community, the Capsis charter defines key aspects of the project concerning co-development and licensing. A major reason of the success of Capsis is its source code sharing policy. When a modeller adds a module in the framework, he (she) becomes a member of the community. He (she) has access to the framework libraries and the common tools, but also to the previously integrated

modules and their source code. The same rules will be applied to his (her) own module code which will be open to other modellers. Every member of the community has full access to the models. However, this does not mean that all the modules are free. The Capsis charter explicitly reminds the intellectual property rights of all partners. Namely, model equations and parameters (i.e. the scientific knowledge) must not be disseminated outside Capsis, whether published or not, without the agreement of their scientific authors. The shared components in Capsis like the kernel, the common libraries and extensions are distributed under a LGPL (http://www.gnu.org/licenses/old-licenses/lgpl-2.1.html) license which is a non viral (meaning the modules may not be LGPL) free and open source license. All these shared components can be used, copied and distributed by everyone. Benefits of open software and co-development have been already shown (Raymond 1999). For scientific projects, a full access to the algorithms and their implementation improves understanding, validation, reproducibility and code continuity. Nevertheless, the Capsis framework does not impose a license for the modules and the embedded models. The authors decide the license for the diffusion of their model, free or commercial. They also choose the way they distribute their model outside the community and the other members of the community are not allowed to distribute them without the author agreement. These rules rely on mutual confidence and favour multiple partnerships. Co-development is possible at the modeller's level thanks to dedicated development and communication tools which are largely used. The use of a centralised code versioning system facilitates the exchange of code, all changes being stored on server. Development issues and requests are grouped on a web application dedicated to the project management. Unit and functional tests are executed continuously (by an automatic system at each version change) on a dedicated server facilitating model verification and improving software reliability (Duvall et al. 2007). The documentation is also written collaboratively on the Capsis wiki (http://www.inra.fr/capsis). Such tools have already been used largely by computer scientists for classic software development, and have shown in the Capsis project great benefits at the modellers' level too.

4 Discussion

4.1 Models and usage

Since 2000, Capsis has been used in more than 50 modelling projects (Table 2 and Online Resource 1), mainly from French research institutes but also through collaborations with modellers from Belgium, Canada, China, New Zealand and Portugal. These models are of various types including stand-level models (Lemoine 1991; Table 2, Lemoine; Tang et al. 1994; Table 2, ISGM), tree distance-independent growth and mortality models (Dreyfus et al. 2001; Table 2, PNN/Pnn2), individual tree distance-dependent models (Courbaud et al. 2001; Table 2, Mountain; Goreaud et al. 2006; Le Moguédec and Dhôte 2011; Table 2, Fagacees), spatially explicit or not. Semi-spatially explicit models mixing tree and cohort distance-independent models on a landscape (with polygonal compartments and cells grids) have also been integrated (Goreaud et al. 2005; Table 2, Ventoux). Growth models have been successfully integrated with other internal models like logging and wood quality models (Dreyfus and Bonnet 1997; de Coligny et al. 2005; Table 2, Fagacees, NZ1, PP3), or external models like tree architecture simulation models (Barczi et al. 2007; Table 2, Eucalypt, PP3) or forest wind risk models (Cucchi et al. 2005; Table 2, PP3). The framework has also been completed with several general purpose libraries like genetics with genotype description (Dreyfus et al. 2005; Table 2, VentouG; Wernsdörfer et al. 2009), spatial structure generation (Goreaud et al. 2006; Table 2, Oakpine1), light interception (Courbaud et al. 2003; Table 2, Mountain, Samsara, Regelight), tree biomechanics (Ancelin et al. 2004; Table 2, Mountain) or economics (Orazio et al. 2002; Table 2, PP3, Regix).

The Capsis framework and its community have promoted the development of several research projects requiring integrated models to explore various scientific questions related to forest growth and dynamics. The use of models is particularly useful in situations where observations and experiments are difficult. Capsis is also used to transfer knowledge to end-users (Meredieu et al. 2009). The Agro-ParisTech (http://www.agroparistech.fr/) forestry school uses Capsis in training sessions for teaching silvicultural management to graduate forestry students or in further education for forest managers. Some forest management organisations like the ONF have also used Capsis to build silvicultural guides (Chabaud and Nicolas 2009). *The silviculture handbook for the French northern Alps* (Gauquelin and Courbeaud 2006), has thus taken advantage of Capsis simulations. The Capsis framework is currently used in the preparation of a similar handbook for the French southern Alps for simulating silvicultural scenarios for stands of *P. nigra, P. sylvestris, P. uncinata, Abies alba, F. sylvatica*, as well as for mixed stands arising from colonisation by beech and silver fir in old pine forests stands (Dreyfus 2008; Table 2, Ventoux).

4.2 An example: the Samsara2 model

Description Samsara2 (Table 2: Samsara) is a spatially explicit, individual-based model of forest dynamics. A forest stand is a list of trees described by their identifier,

species code, x, y and z coordinates and diameter at breast height (dbh). The area is divided in square cells of size 5x5m. Seedlings on each cell are described by their species, location and height. Direct and diffuse incident lights are split into light rays. Ray tracing allows to calculate the global radiation intercepted by each tree during a vegetation season and the remaining irradiance under canopy on each ground cell (Courbaud et al. 2003). Allometries are used to calculate individual tree total height, crown base height and crown radius (Vieilledent et al. 2010b). Tree growth depends on the energy intercepted by the canopy (Vieilledent 2009). Mortality depends on tree dbh and growth (Vieilledent et al. 2010a). Recruitment depends on adult fecundity, dispersion and light on the ground (Vieilledent 2009). The model simulates multi-species forests and has been calibrated for *A. alba—P. abies* forests in the French Alps.

Usage A simulation is initialised by loading a file describing every tree characteristics in the initial forest stand, plot parameters and species parameters for demographic equations. After initialisation, different scenarii can be simulated for different durations and for different thinning strategies. Several thinning tools can be used to cut trees. For example, an interactive diameter distribution can be manipulated using small cursors to decrease the number of trees in diameter classes. Other cutting tools such as a stand map allowing interactive tree marking have been developed more specifically for this model. The simulation results can then be viewed with several generic stand-level and tree-level analysis tools (Fig. 3), for instance the stand dbh distribution or basal area computation along time. More specific views are also available such as crown projection maps and irradiance maps or the distribution of irradiance levels among cells or tree distribution among vertical strata. Either result variables can be saved directly in a file or full simulations can be saved and reopened later for further analysis.

Samsara2 and the previous versions named *Mountain* and *Samsara* have been used as exploration, teaching and scientific tools. For example, they helped to test silviculture scenarios to prepare the writing of a management guideline in collaboration with forest managers, to confront forestry students to the practical difficulty of maintaining mixed uneven-aged stands on a long run, or to analyse field tree markings with forest managers (Gauquelin et al. 2008). For such applications, the *interactive* mode has been used: simulations were built step by step, controlling stand evolutions before deciding the characteristics of each new cutting. In more scientific applications, numerous simulation replications could be made with the *script* mode. In these cases, silviculture strategies were completely determined right from the beginning. Such approaches made it possible to compare the impact of individual tree selection versus gap selection on forest protection efficiency against natural hazards (Cordonnier et al. 2006) or to analyse the impact of spatial stand structure on long term dynamics (Goreaud et al. 2006).

Implementation The development of the model took advantage of the framework. The model objects like the forest stand, trees and plot cells have been based on existing generic Capsis objects. A great advantage of this approach is that all the standard operations such as managing the relations between the stand, trees and cells, managing the simulation history, or integrating the user interface had already been prepared at the generic object level and needed only very few new development. The generic data analysis tools could be used without any modifications while some more specific tools like the irradiance map have been developed especially for the model. These new tools may then become available for other spatially explicit individual-based models integrated by the Capsis community. Samsara2 has also taken advantage of other developments done by the community like the Foreole wind loading library (Ancelin et al. 2004) and the Spatial distribution analysis library (Goreaud et al. 2006). This example shows the benefit of developing a forest model in Capsis. Only the specific requirements of the model has to be implemented and every new feature may then be shared with other modellers.

Benefits This example shows that there is a clear benefit of using Capsis for developing a forestry model. The growing community is organised around a common framework and every new implemented feature becomes available to everyone. For instance, a need in Samsara2 was to run structured simulation plans to do sensitivity analyses. This need was shared by several other modellers and this feature has been implemented for the entire community. The evolution of the framework is driven by both the computer scientists and the modellers themselves. This example also shows that the framework is not only dedicated to modellers but also to final users. Samsara2 was used as a teaching tool in education. Once integrated, the model can be used not only as a research tool but also as a user-friendly software. Finally, we showed the benefit to have a domain-specific framework, here applied to forestry, rather than a generic framework based on a particular modelling paradigm. The time, space, forest and tree representations used in Samsara2 are not the only ways to write models in Capsis. The framework proposes a large and growing number of forestry objects allowing scientists to address different problems with different assumptions and modelling approaches with the same framework.
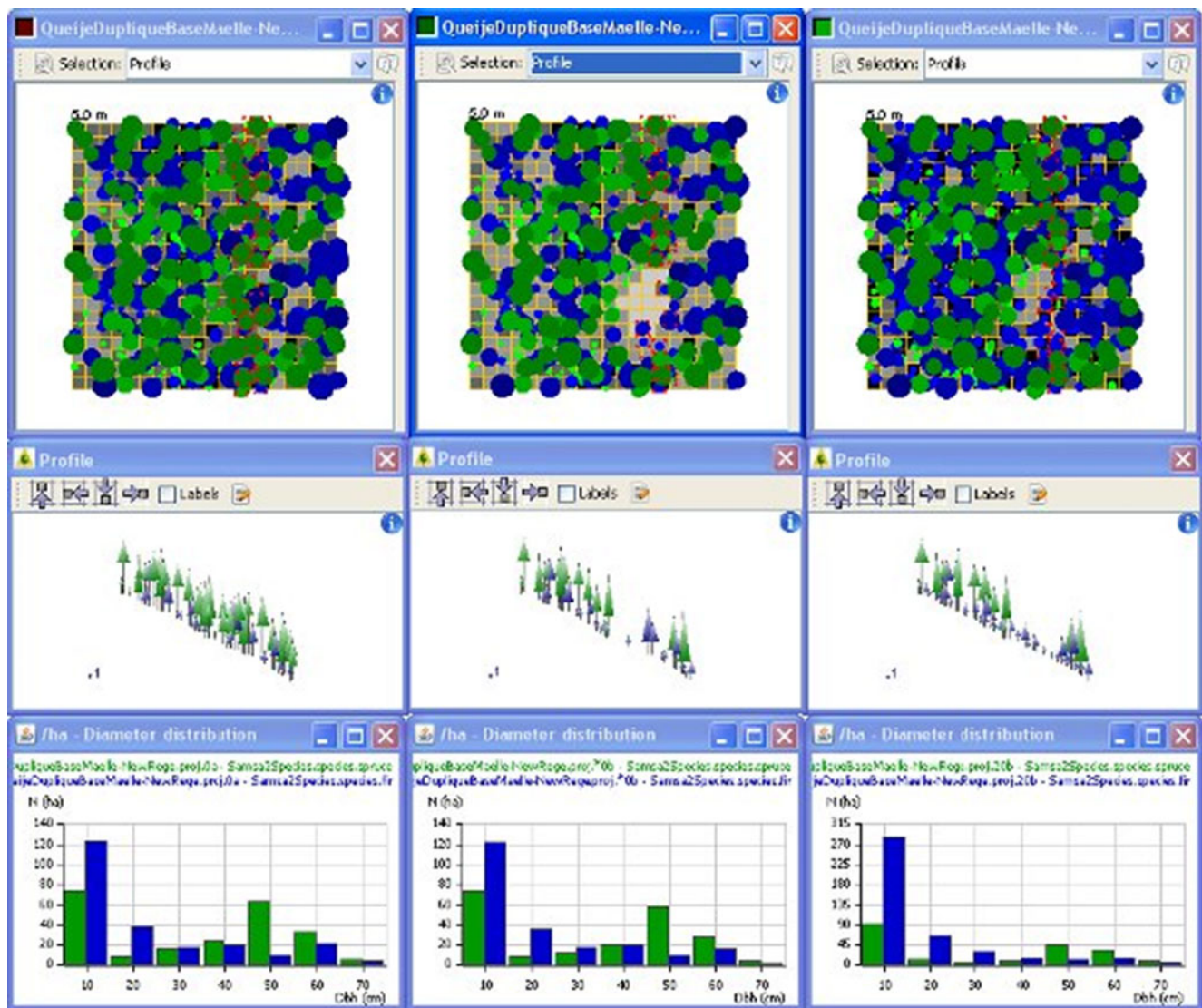
Fig. 3 Some visualisations of a simulation with Samsara2 on the Capsis result panel. A 1-ha observation plot in the forest of Queige (Savoie) is visualised before thinning, after thinning and 20 years after thinning. Plot maps show tree crown projections and irradiance levels under canopy. Spruce trees are represented in *green* and fir trees in *blue* with darker colours for higher canopy strata. *Grey* levels represent cell irradiance (*darker grey* for lower irradiance). *Small red dots* on the maps indicate transects, selected for vertical profiles presented below. The three histograms represent dbh distributions for spruce and fir. The cutting has consisted in a small gap and selective cutting of some of the biggest fir trees

5 Conclusions

The Capsis framework, its open architecture, the co-development principles and the community structure provide an environment to create forest models. The specificity of Capsis comes from the combination of a software framework adapted to forest models structure and an active community with clear role attributions. The open architecture is adapted to support future evolutions, new modelling approaches and ensure sustainability of the existing modules. The use of such a framework is a great improvement in forest model engineering and has replaced

with great benefit costly and time consuming ad hoc simulation software development. In 10 years, Capsis 4 has been used successfully in many modelling projects and has shown real benefits for its members. Co-development improves software reuse and simplifies models integration. In addition to a common technical base, the Capsis project promotes exchanges in the forestry modelling community. The great number of models integrated in Capsis shows that this shared approach addresses quite well the forest modellers needs. Capsis can also be used efficiently to transfer knowledge to students and to end-users, this result was illustrated by the writing of silvicultural handbooks

based on Capsis model simulations. The Capsis project is not static and the framework evolves continuously to answer new modelling challenges.

References

Ancelin P, Courbaud B, Fourcaud T (2004) Developing an individual tree-based mechanical model to predict wind damage within forest stands. For Ecol Manag 203:101–121

Argent RM (2004) An overview of model integration for environmental applications—components, frameworks and semantics. Environ Model Softw 19:219–234

Argent RM, Rizzoli AE (2004) Development of multi-framework model components. Complexity and integrated resources management. In: Pahl-Wostl C, Schmidt S, Rizzoli AE, Jakeman AJ (eds) Transactions of the 2nd biennial meeting of the international environmental modelling and software society: complexity and integrated resources management (1). International Environmental Modelling and Software Society, ISBN 88-900787-1-5, pp 365–370

Argent RM, Voinov A, Maxwell T, Cuddy SM, Rahman JM, Seaton S, Vertessy RA, Braddock RD (2006) Comparing modelling frameworks—a workshop approach. Environ Model Softw 21:895–910

Barczi J, Rey H, Caraglio Y, De Reffye P, Barthélémy D, Qiao XD, Fourcaud T (2007) AmapSim: a structural whole-plant simulator based on botanical knowledge and designed to host external functional models. Ann Bot London 101:1125–1138

Brisson N, Ruget F, Gate P, Lorgeou J, Nicoullaud B, Tayot X, Plenet D, Jeuffroy MH, Bouthier A, Ripoche D, Mary B, Justes E (2002) STICS: a generic model for simulating crops and their water and nitrogen balances. II. Model validation for wheat and maize. Agron Sustain Dev 22:69–92

Chabaud L, Nicolas L (2009) Guide des sylvicultures. Pineraies des plaines du centre et du Nord-Ouest. ONF, Paris

Coates KD, Canham CD, Beaudet M, Sachs DL, Messier C (2003) Use of a spatially explicit individual-tree model (SORTIE/BC) to explore the implications of patchiness in structurally complex forests. For Ecol Manag 186:297–310

Cordonnier T, Courbaud B, Franc A (2006) Permanence of stability properties and protection efficiency in mountain spruce stands. In: Natural disturbance-based silviculture—managing for complexity, IUFRO 1.05 Conference, Rouyn-Noranda, Québec, 14–18 May 2006, ISBN-10: 2-923064-17-8

Courbaud B, Goreaud F, Dreyfus P, Bonnet FR (2001) Evaluating thinning strategies using a tree distance dependent growth model: some examples based on the CAPSIS software "uneven-aged spruce forests" module. For Ecol Manag 145:15–28

Courbaud B, de Coligny F, Cordonnier T (2003) Simulating radiation distribution in a heterogeneous Norway spruce forest on a slope. Agric For Meteorol 116:1–18

Cucchi V, Meredieu C, Stokes A, de Coligny F, Suarez J, Gardiner BA (2005) Modelling the windthrow risk for simulated forest stands of Maritime pine (*Pinus pinaster* Ait.). For Ecol Manag 213:184–196

de Coligny F, Meredieu C, Labbé T, Vallet P, Dreyfus P (2005) Using Capsis for connection with wood quality. In: Proceedings of the 5th IUFRO workshop on connection between forest resources and wood quality: Modelling Approaches and Simulation Software. Waiheke Island, New Zealand, 20–27 November 2005

Degenne P, Lo Seen D, Parigot D, Forax R, Tran A, Ait Lahcen A, Cure O, Jeansoulin R (2009) Design of a domain specific language for modelling processes in landscapes. Ecol Model 220:3527–3535

Dreyfus P (2008) Dynamiques du Sapin, du Hêtre et des Pins dans l'arrière-pays méditerranéen: de la modélisation à l'aide à la gestion. Rev For Fr 60:233–249

Dreyfus P, Bonnet FR (1995) CAPSIS, logiciel de simulation de conduites sylvicoles. Rev For Fr 47:111–115

Dreyfus P, Bonnet FR (1997) Capsis (Computer-Aided Projection of Strategies In Silviculture): an interactive simulation and comparison tool for tree and stand growth, silvicultural treatments and timber assortment. In: Nepveu G (ed) Proceedings of the second IUFRO WP S5.01.04 workshop: connection between silviculture and wood quality through modelling approaches and simulation software, pp 194–202

Dreyfus P, Hamza N, Pignard G (2001) Construction de modèles de croissance pour les peuplements réguliers à partir des données dendrométriques de l'IFN. Rev For Fr 53:434–441

Dreyfus P, Pichot C, de Coligny F, Gourlet-Fleury S, Cornu G, Jésel S, Dessard H, Oddou-Muratorio S, Gerber S, Caron H, Latouche-Hallé C, Lefèvre F, Courbet F, Seynave I (2005) Couplage de modèles de flux de gènes et de modèles de dynamique forestière. Un dialogue pour la diversité génétique—Actes du 5ème colloque national BRG, Lyon, 3–5 Novembre 2004, Les Actes du BRG n°5, ISBN 2-908447-33-9, pp 231–250

Duvall P, Matyas S, Glover A (2007) Continuous integration: improving software quality and reducing risk. Free Software Foundation, Inc. 1999. Addison-Wesley, Reading

Gardiner BA, Quine CP (2000) Management of forests to reduce the risk of abiotic damage—a review with particular reference to the effects of strong winds. For Ecol Manag 135:261–277

Gauquelin X, Courbeaud B (2006) Guide de sylviculture des forêts de montagne—Alpes du Nord françaises. Cemagref—CRPF Rhône-Alpes—Office National des Forêts

Gauquelin X, Courbaud B, Fay J, Berger F, Mermin E (2008) Conduite de peuplements mélangés en forêt de montagne: exemple d'un transfert chercheurs-gestionnaires. Rev For Fr LX:207–214

Goreaud F, de Coligny F, Courbaud B, Dhôte JF, Dreyfus P, Pérot T (2005) La modélisation: un outil pour la gestion et l'aménagement en forêt. VertigO 6:12

Goreaud F, Alvarez I, Courbaud B, de Coligny F (2006) Long-term influence of the spatial structure of an initial state on the dynamics of a forest growth model: a simulation study using the Capsis platform. Simulation 82:475–495

Hillyer C, Bolte J, van Evert F, Lamaker A (2003) The ModCom modular simulation system. Eur J Agron 18:333–343

Keating BA, Carberry PS, Hammer GL, Probert ME, Robertson MJ, Holzworth D, Huth NI, Hargreaves JNG, Meinke H, Hochman Z, McLean G, Verburg K, Snow V, Dimes JP, Silburn M, Wang E, Brown S, Bristow KL, Asseng S, Chapman S, McCown RL, Freebairn DM, Smith CJ (2003) An overview of APSIM, a model designed for farming systems simulation. Eur J Agron 18:267–288

Lacerte V, Larocque GR, Woods M, Parton WJ, Penner M (2006) Calibration of the forest vegetation simulator (FVS) model for the main forest species of Ontario, Canada. Ecol Model 199:336–349

Le Moguédec G, Dhôte JF (2011) "Fagacées": a tree-centered growth and yield model for Sessile oak (Quercus petraea L.). Ann For Sci. doi:10.1007/s13595-011-0157-0

Lemoine B (1991) Growth and yield of maritime pine (Pinus pinaster Ait): the average dominant tree of the stand. Ann Sci For 48:593–611

Lorek H, Sonnenschein M (1999) Modelling and simulation software to support individual-based ecological modelling. Ecol Model 115:199–216

Meredieu C, Dreyfus P, Cucchi C, Saint-André L, Perret S, Deleuze C, Dhôte JF, de Coligny F (2009) Utilisation du logiciel Capsis pour la gestion forestière. Forêt-entreprise 186:32–36

Monserud RA, Sterba H (1996) A basal area increment model for individual trees growing in even- and uneven-aged forest stands in Austria. For Ecol Manag 80:57–80

Moore RV, Tindall CI (2005) An overview of the open modelling interface and environment (the OpenMI). Environ Sci Policy 8:279–286

Muetzelfeldt R, Massheder J (2003) The Simile visual modelling environment. Eur J Agron 18:345–358

Muys B, Hynynen J, Palahí M, Lexer MJ, Fabrika M, Pretzsch H, Gillet F, Briceño E, Nabuurs GJ, Kint V (2011) Simulation tools for decision support to adaptive forest management in Europe. Systems 19:96–99

Nguyen DZ, Ricken M, Wong S (2004) Design patterns for marine biology simulation. In: SIGCSE'04: Proceedings of the 35th SIGCSE technical symposium on computer science education. ACM, New York, ISBN: 1-58113-798-2, pp 467–471

Orazio C, Meredieu C, Saint-André L, de Coligny F (2002) Building bridges between modellers and end-users. A case study of Southern Europe. In: Colloque IUFRO, IEFC, ISA on incorporating forest growth models into decision-support tools for sustainable forest management, Lisbon, Portugal, 6–8 June 2002

Ottorini JM (1991) Growth and development of individual Douglas-fir in stands for applications to simulation in silviculture. Ann For Sci 48:651–666

Papajorgji P (2005) A plug and play approach for developing environmental models. Environ Model Softw 20:1353–1357

Porté A, Bartelink HH (2002) Modelling mixed forest growth: a review of models for forest management. Ecol Model 150:141–188

Pradal C, Dufour-Kowalski S, Boudon F, Fournier C, Godin C (2008) OpenAlea: a visual programming and component-based software platform for plant modeling. Funct Plant Biol 35:751–760

Pretzsch H, Biber P, Dursk J (2002) The single tree based stand simulator SILVA: construction, application and evaluation. For Ecol Manag 162:3–21

Quesnel G, Duboz R, Ramat E (2009) The virtual laboratory environment—an operational framework for multi-modelling, simulation and analysis of complex dynamical systems. Simul Model Pract Th 17:641–653

Rahman JM, Seaton SP, Perraud JM, Hotham H, Verrelli DI, Coleman JR (2003) It's TIME for a new environmental modelling framework. In: MODSIM 2004 International Congress on Modelling and Simulation. Modelling and Simulation Society of Australia and New Zealand Inc, Townsville, Australia, pp 1727–1732

Rahman JM, Seaton SP, Cuddy SM (2004) Making frameworks more useable: using model introspection and metadata to develop model processing tools. Environ Model Softw 19:275–284

Rasinmki J, Mkinen A, Kalliovirta J (2009) SIMO: an adaptable simulation framework for multiscale forest resource data. Comput Electron Agric 66:76–84

Raymond ES (1999) The Cathedral and the Bazaar. O'Reilly & Associates, Inc., Sebastopol, CA

Rigolot E, de Coligny F, Dreyfus P, Dupuy JL, Lecomte I, Pezzatti B, Vigy O, Pimont F (2010) Fuel Manager: a vegetation assessment and manipulation software for wildfire modelling. In: Viegas DX (Ed) ProcEEDINGS of the 6th international conference on forest fire research, University of Coimbra, Portugal. 12 pp

Roxburgh SH, Davies ID (2006) COINS: an integrative modelling shell for carbon accounting and general ecological analysis. Environ Model Softw 21:359–374

Sequeira RA, Olson RL, McKinion JM (1997) Implementing generic, object-oriented models in biology. Ecol Model 94:17–31

Tang S, Meng CH, Meng FR, Wang YH (1994) A growth and self-thinning model for pure even-age stands: theory and applications. For Ecol Manag 70:67–73

Vieilledent G (2009) Structurer l'incertitude et la variabilité dans les modèles de dynamique forestière. Application à la coexistence du Sapin et de l'Epicéa en forêt de montagne. Thèse, AgroParisTech-ENGREF, Nancy

Vieilledent G, Courbaud B, Kunstler G, Dhôte JF (2010a) Mortality of Silver fir and Norway spruce in the western Alps—a semi-parametric approach combining size-dependent and growth-dependent mortality. Ann For Sci 67:305

Vieilledent G, Courbaud B, Kunstler G, Dhôte JF, Clark JS (2010b) Individual variability in tree allometries determines light resource allocation in forest ecosystems—a hierarchical Bayes approach. Oecologia 163:759–773

Vincent G, de Foresta H (1998) A three dimensional dynamic model of Damar agroforest in Sumatra (Indonesia). In: Enriquez GL, Wasrin UR and Murdiyarso D (eds) Tropical forest dynamics. Biotrop Special Publication No. 60. pp 139–157

Wernsdörfer H, Caron H, Gerber S, Cornu G, Rossi V, Mortier F, Gourlet-Fleury S (2009) Relationships between demography and gene flow and their importance for the conservation of tree populations in tropical forests under selective felling regimes. Conserv Genet. doi:10.1007/s10592-009-9983-0