AMAPstudio: a Software Suite for Plants Architecture Modelling

Sébastien Griffon
Cirad, UMR AMAP
F-34000, Montpellier, France
Corresponding author: sgriffon@cirad.fr

François de Coligny
INRA, UMR AMAP
F-34000, Montpellier, France

***Abstract-AMAPstudio* is a user-friendly software suite designed for botanists and agronomists to edit, visualize, explore and simulate multi-scale plant architecture.**

It contains interactive tools to handle the topology (e.g. organs addition or deletion), the geometry (e.g. 3D selection, edition, rotation) and the dynamics (i.e. time line, scenarios) of plants at the individual or scene scale. *AMAPstudio* is based on the *Multi-scale Tree Graph* (MTG) data structure, which is commonly used to represent plant topology. Users can explore this data structure to test or to improve hypotheses on plant development. Specific data can be extracted with combinations of criteria and can be visualized in tables and graphs. Simple analysis functions can be launched or data can be exported to external tools, e.g. *R*, or any other statistical computing environment, for more specific analyses. *AMAPstudio* is also a framework in which modellers can integrate their own plant simulation models. Different scenarios can be computed for a growth model by interactively modifying model parameters or plant structure (e.g. by pruning) at particular time steps.

***AMAPstudio* is an open software built according to the flexible *Capsis* methodology. It is a free open-source software (LGPL) available on all *Java* compatible operating systems and it can be downloaded on http://amapstudio.cirad.fr.**

Keywords-plant architecture; modelling; 3D; simulation; framework; software

I. Introduction

Improving knowledges on plant architecture [1] help to answer questions in ecology, agronomy and forestry for instance on plant acclimation and adaptation to changing environments.

For this purpose, computational plants are becoming more and more popular and a lot of efforts are carried out to build structural and functional models at the organ level in order to finely simulate plant structure growth strategies [2]. In this context, scientists need software to help to model plant architecture, to simulate their growth and to analyse simulation output.

In the 80's, one of the first plant architecture growth simulators was designed by the *AMAP* laboratory [3]. The *AMAP* software suite contained a plant architecture growth engine and a landscape manager aimed at computer graphics. A large set of parameter values has been designed for various plant species (around 660 parameter files). It was followed by the development of the *AMAPsim* software that introduced physiological ages and the reference axis model [4] to improve botanical accuracy.

Almost at the same time, the *L-Studio/VLab* software was developed by Prusinkiewicz and collaborators [5] to simulate plant growth based on *L-systems*. With this formalism, the development of plants is computed according to a set of rewriting rules. Another software based on the *L-systems* is *GroIMP* designed by Kurth [6]. It is an open-source software that extends *L-Systems* principle to general graph rewriting with relational graph growth (RGG).

Later, the *OpenAlea* component-based framework was designed "*to facilitate the integration and interoperability of heterogeneous models and techniques from different scientific disciplines"* [7]. Its *VPlant* package is more dedicated to plant architecture modelling. The whole platform provides a visual programming interface to build and connect a wide range of tools dealing with plant models from the meristem to the scene scales, but also to deal with data analysis and image processing.

Other software like *Arbaro* [8] and *XFrog* [9] with poorer botanical knowledge or other approaches like self-organizing models [10] can be used to build plant structures. They are more dedicated to computer graphics, video games and image production.

The *AMAPstudio* project has been developed in the *AMAP* laboratory since the end of 2008 focusing on edition, visualization, exploration and simulation of plant architecture through a rich graphical user interface for common use or by scripting for expert users. *AMAPstudio* proposes a generic data structure encoding the plant topology and geometry. It can manage an individual plant with its organs, as well as a scene made of individual plants plus other objects.

AMAPstudio is also a framework to host individual-based plant growth models. Each model can indeed update the plant data structure whatever its simulation

formalism. It is based on the *Capsis* framework and methodology [11]. The *Capsis* scenarios management and its co-development approach were adapted.

The outline of the paper is as follows. In section II, we will introduce *AMAPstudio's* main options concerning topology and geometry management, section III will explain the features of the main software components. Two user applications will be shortly presented in section IV to illustrate the capabilities of the software suite, section V will finally discuss the various possible ways to join the project.

II. A CENTRAL PLANT DESCRIPTION

A. Plant topology

The *ArchiTree* is an extended implementation of the *Multi-scale Tree Graph* (MTG) formalism [12] to encode the plant architecture (topology and geometry). This formalism is able to consider the plant at different scales, for instance the axes scale, the growth units scale or the internodes scale according to the modelling hypotheses for the current study. Each node of the graph, or *ArchiNode*, is a plant entity with a type characterised by a decomposition level.

The *ArchiNodes* can be linked together with three types of topological graph edges to represent decomposition, succession and branching relationships (Fig. 1). These edges are oriented and typed according to their topology meaning (i.e. ancestor, successor, complex, component, bearer and branch). The *ArchiTree* offers high-level methods for assembling nodes (e.g. add, insert or remove at various scales), ensuring the graph consistency. *Iterators* and *Visitors* [13] are also provided to respectively sequentially traverse the graph in various ways (e.g. preorder, postorder) or to apply a function on each selected node.

The *ArchiNodes* can have many kinds of predefined or user custom attributes: numbers, texts, dates, colors, images with optional management of their units. Some of these properties can be used for the plant geometry building (see geometry section below).

In addition, because plant architecture may sometimes include similar subsets, the *ArchiTree* can handle *substructures* [14], i.e subtrees that can be referenced several times in the *ArchiTree* for memory and computing optimisation.

B. Plant geometry

Each *ArchiNode* has a specific geometrical property to handle its shape (a mesh) and transformation matrix. This 4x4 matrix is expressed from a global reference frame. These transformations can be automatically computed by a default geometry builder according to information saved in the node attributes (Fig. 1). These attributes can represent lengths, widths, heights, absolute coordinates, relative angles (e.g. insertion angles on bearer), local angles (e.g. Euler angles), absolute angles (e.g. azimuth) and other specific plant angles (e.g. phyllotaxy, bending, horizontal deviation).

Transformations are applied on meshes that represent branches and organs shapes. *AMAPstudio* provides standard normalized shapes for common organs (e.g. leaves, flowers), and a set of parametrized shapes (e.g. plane, sphere, tube).

For instance, a leaf mesh can be scaled, rotated and translated according to the transformation matrix, or a tube shape can be computed for a branch (i.e. a 2D circle contour is extruded along a 3D path defined by the branch component positions).

The geometry builder can selectively handle *ArchiTree* geometry at different scales and manage different geometry resolutions of the same plant architecture. To ensure coherence between scales, a set of bottom-up and top-down scale constraints are defined and can be extended. In a bottom-up approach, a list of *ArchiNode* geometries will be used to compute the bounding shape, while in a top-down approach, a macro scale geometry can be used to constrain a smaller scale geometry.

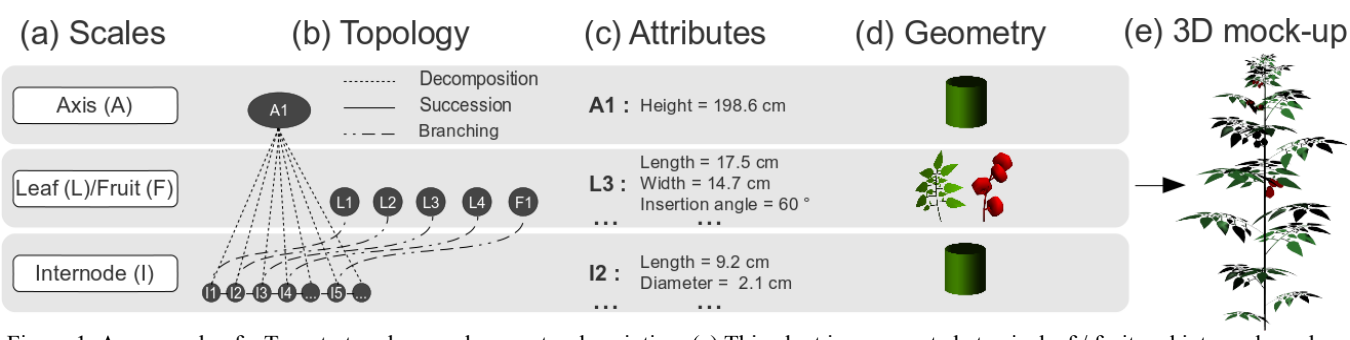

Figure 1. An example of a Tomato topology and geometry description. (a) This plant is represented at axis, leaf / fruit and internode scales, (b) the main axis is decomposed into successive internodes which can branch into leaves and fruits. (c)(d) Each organ is an ArchiNode and can have attributes and geometry informations. (e) The default geometry builder can compute a plant 3D mock-up.

When geometrical data are missing in the *ArchiTree* at some *ArchiNodes*, the geometry builder can propose alternative computing based on default values (e.g. for phyllotaxy angles) and default functions (e.g. linear interpolation for branch diameters). Depending on the species, an important configuration may be needed in some cases to give a realistic enough representation.

Moreover, for rendering purpose, each mesh contains appearance data defining material (i.e. ambient, diffuse, specular colors) and possibly texture information.

C. Description for several plants

A scene made of several plants is represented in *AMAPstudio* by a list of *ArchiTrees* with positions. These plants are located on a terrain with optional additional objects (e.g. polylines, polygons).

III. A SOFTWARE SUITE

A. Main objectives

AMAPstudio was designed for the plant architecture modelling community and its development was driven by several objectives: (i) a user friendly interface designed for modellers and end-users, (ii) no simulation paradigm imposed to the modeller for wider genericity, (iii) an integrative workspace to mutualise developments, (iv) interoperable with various plant modelling software and other toolkits (e.g. *OpenAlea/VPlant* [7], *Blender* [15], *Abaqus* [16]).

B. From the individual to the scene

The architectural approach may lead to big data structures in memory. For adult trees, the number of *ArchiNodes* can be huge and the multi-scale orientation increases again the need for memory. Indeed, an adult palm tree may contain 200000 *ArchiNodes* and several trees may be needed for some considerations (see section IV.B). To address this problem, *AMAPstudio* proposes two different tools to study one single plant or several plants.

Xplo is dedicated to the individual level. The recent computers performances make it possible to load a whole multi-scale plant description in memory even if it is large. *Xplo* provides a set of tools to edit the plant architecture in details. The user can add, remove, insert and prune virtual plant organs and edit their attributes. A 3D geometry mock-up is automatically updated while editing. The *ArchiTree* can be displayed in different views (see section III.C) to finely explore the plant structure. The rendering can be configured to display or hide some nodes, to change their shape and color.

Beside plant architecture edition, *Xplo* provides features for filtering, browsing, extracting and plotting data interactively in tables and graphs. The extraction tables have import and export facilities to be able to perform external statistical analysis in dedicated software (e.g. *Excel*, *R*). Furthermore, external features may also be called, in particular exporting to realistic rendering software.

All plant data, including topology, geometry and attributes, can be saved to an *Open Plant Format* (OPF) file. This format is based on XML and is extensible.

Simeo deals with the scene level (see section II.C). The adaptive memory management is part of Simeo's specifications: the memory footprint is reduced and the detailed information (e.g. 40 adult trees equivalent to 3.8 Giga bytes on the file system) can be reloaded when needed for finer computations or at export time.

Specific scene level features help to add, remove, edit and move plants. It is possible to lay out plants in the scene by selecting configurable plantation patterns. The simplest one lays out by clicking in the scene with the mouse. New patterns can be added in new plugins when needed. The features which are provided in *Xplo* for a single plant may be provided in *Simeo* for the whole scene (e.g. radiative balance, export to an external realistic viewer). The embedded sketch view is adapted for large amount of data: the fine geometry of the plant organs is replaced by simpler geometries to be able to display more plants.

The scene can be saved to a text-based *Open Plant Scene* format (OPS) listing the OPF plants in the scene with their locations and other global attributes.

Xplo and *Simeo* also have common features. Thus for complex or repetitive tasks, it is possible to write scripts in *Java* or *Groovy* language. High-level functions are available to help for plant geometry reconstruction, data analysis and various other tasks. Moreover, plugins can be added in the two applications to import or export data to other standard file formats (e.g. MTG files). Their graphical user interfaces share components and features like the 3D preview panel, the common scenario manager, the multi projects management and the undo / redo based commands.

C. Different views for various purposes

Xplo and *Simeo* provide several kinds of views of the current plant or scene (Fig. 2).

The user can have a general overview of his plant or his scene in a schematic three dimensional view. Such a view is very efficient to quickly check the main shape of the object(s). Some refinements can be made available to change the rendering and colors of the viewed items in order to check things visually. These 3D previews are part of the main components of the graphical user interfaces of *Xplo* and *Simeo*.

In some cases, it may be more suitable to use tabular views. These tables can indeed show more accurately the

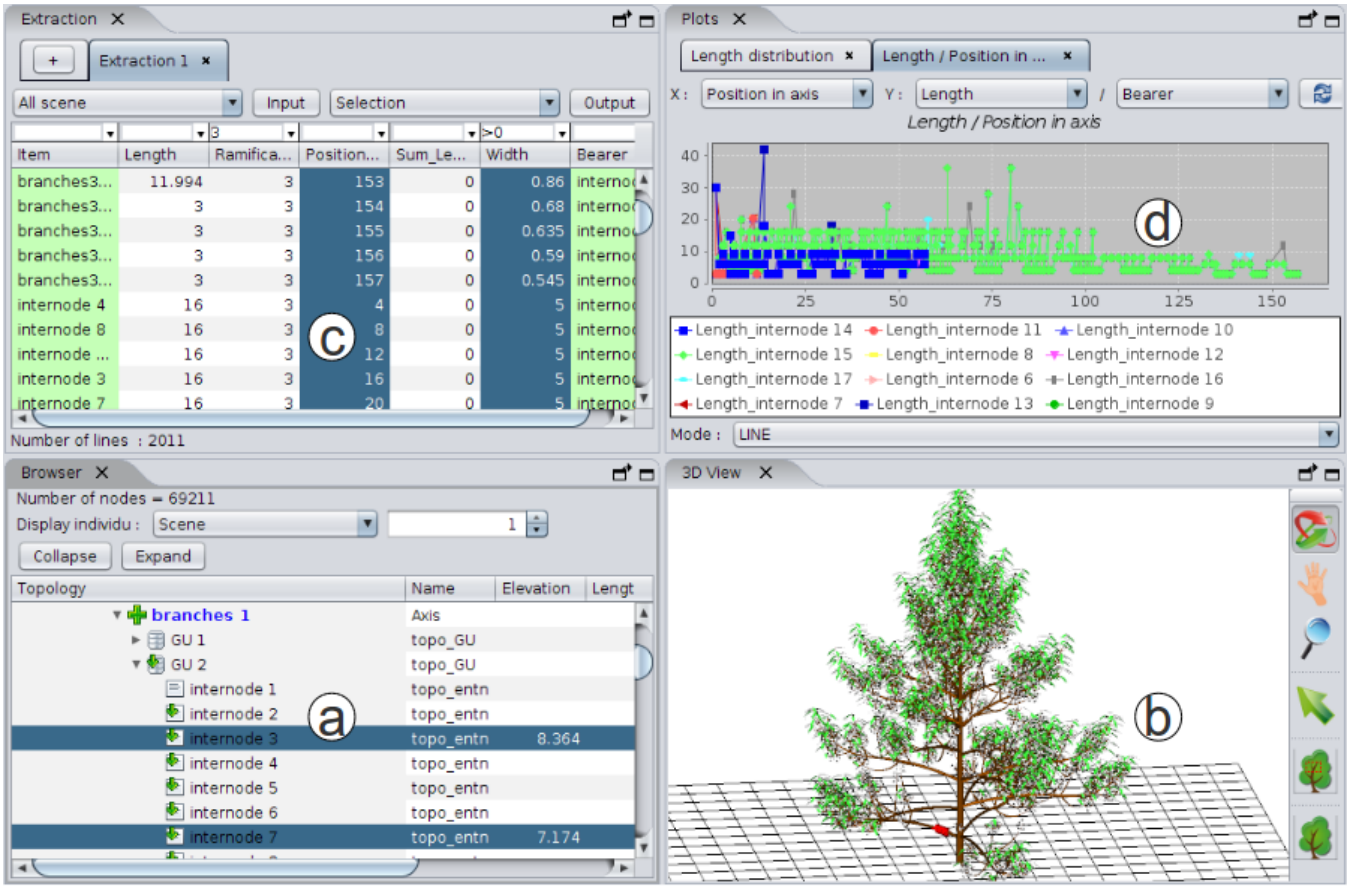

Figure 2. Different views of the same plant in Xplo. The user may (a) browse, select and edit plant topology and attributes, (b) display the plant 3D mock-up and select organs, (c)(d) extract, filter and plot data.

values of the plant organs attributes like sizes, angles, colors or any other attribute in the *ArchiTree*. They can also be used as editors to change the values of these attributes or even add new attributes.

These various components are multiple views of the same data model containing one or several *ArchiTrees*. This means that changing a value from a view directly updates the underlying plant(s) and the other views. In the same way, selecting an organ or a plant in a view automatically propagates selection to the underlying data model and to the other views. This technique is known in Computer Science as the *Model View Controller* design pattern [13] and is implemented in *AMAPstudio* in the *Sketch* reusable library.

Sketch proposes several refinements regarding the 3D schematic rendering. The 3D scene is parted in items with a type (e.g. axis, leaf). *OpenGL* Renderers can be built as so many *Sketch* plugins to render in different manners the items of a given type. These renderers come with their own configuration panels with real time effect. Renderers for the same types can be switched on demand. These capabilities result in a very flexible framework with a lot of possible configurations to provide the most comfortable editing experience.

D. A simulation framework

Xplo and *Simeo* were built according to the *Capsis* framework [11]. This framework helps to design software hosting evolution models. It provides a software kernel with generic built-in features such as choosing a model in an embedded model library, initializing it with a chosen set of parameters, getting an initial situation for a simulation, managing a simulation history in memory (*Session*, *Projects*, *Steps* with a date), building scenarios mixing stages of evolution and user interventions at given times.

Some key software components are directly available, such as the *Engine* to manage the models and the history, various commands to manage *Projects* (New, Open, Save...), an interactive *Project Manager* to be integrated in the target application graphical user interface or a script mode to easily run long or repetitive simulations.

The *Capsis* framework was initially built for an application dedicated to forestry but it can be used for other thematics, like for plant architecture modelling.

Xplo hosts single plant level growth simulators and *Simeo* embeds growth models at the scene level thus enabling to take into account the possible competition between neighbouring plants in future studies.

E. Extensible software

The *Capsis* framework includes an optional *Extension Manager*. This feature is available in *Xplo* and *Simeo*. Extensions are software components which can be built and plugged to a given application at any time. This option is very convenient to postpone the addition of secondary features. *AMAPstudio* includes several types of such extensions: *interventions* can be run at any step of the simulation to simulate a human or a natural event, *imports* and *exports* can interpret files to load plants or write them to files under various file formats (e.g. *Wavefront* OBJ files), *renderers* may run external rendering tools with realistic capabilities.

The *Extension Manager* makes it possible to add tools when needed without having to change anything in the target software, the user will just see a new item in a list of selectable plugins. All extensions which are not directly based on a graphical user interface (e.g. a diagram) can also be used in script mode, especially intervention methods and file exports.

IV. Applications

In this section, two brief presentations of recent scientific works are presented to illustrate some features of *AMAPstudio* regarding analysis and simulation.

A. Retrospective analysis of fir sapling growth versus light interception

AMAPstudio helped to assess the effects of light environment on silver fir saplings architecture [17]. It mainly consists in reconstruction of individual plants, simulation of stands, calculation of light interception and data extraction for further analyses (Fig. 3).

Plant measurements were performed on 4 years old saplings with a comprehensive topological description at growth unit (GU) scale. The lengths, diameters and positions have been recorded with the MTG formalism and imported into *Xplo* for data exploration. A *Groovy* script was written to compute the geometry of each plant. This script provided simple geometrical rules based on additional measurements for branches (insertion angles) and needles (spacing and orientations). Then, virtual scenes were built under *Simeo* by positioning 3D fir models with their observed positions and orientations. Finally, the scene was exported to the integrated *MMR* library for computing the irradiance of plant organs in 3D virtual scenes [18]. The computed irradiance was added as a new organ attribute in the *ArchiTree*, therefore making it accessible by the extraction tools. Once extracted, data were analysed in the *R* software environment.

B. Palmaceae growth simulation

Presently, *AMAPstudio* is also used for modelling and simulating the growth of palm tree aerial vegetative and reproductive parts (Fig. 4). The *Principes* model is developed inside the *MOCAF* network project [19]. It is a continuation of studies that have been carried out for almost 20 years on palm tree architecture [20][21].

Palm tree is characterised by a single meristem with indeterminate growth, lateral fronds and inflorescences. Although the topology is quite simple, based on the *Corner* architectural model [22], an accurate geometrical description is needed. Therefore, the growth model is based on trunk internodes lengths and diameters as well as detailed frond characteristics (e.g. frond nervure length, leaflet number and spacing, petiole and rachis curvature, insertion angles).

A parameter file including these synthetic parameters and piecewise functions (mean and standard deviations according to position or time) is constructed to represent the palm tree architecture variability. Such a parameter file is then used by the simulator to produce a specific palm tree growth scenario, taking into account the observed intra and inter tree variability. Many parameter files can then reproduce specific palm tree variability and help to determine selection criteria.

Each 3D mock-up produced by the simulator can be imported into *Simeo* to build a virtual scene that reproduces real field patterns. Computing the stand

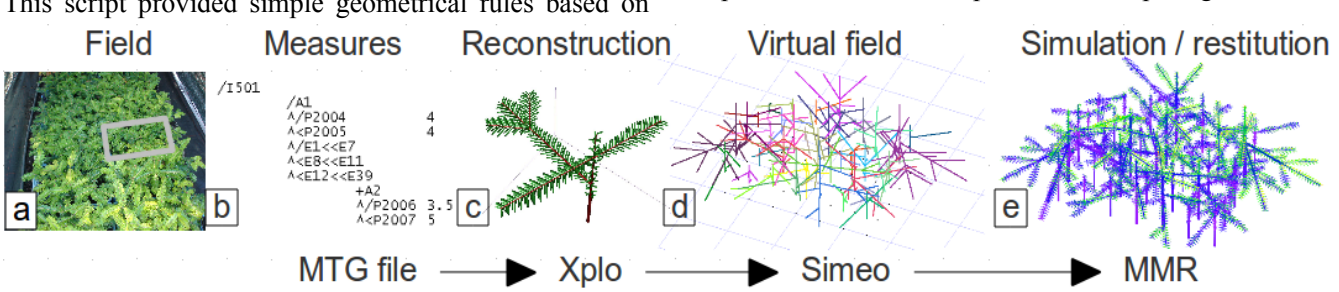

Figure 3. Fir sapling light interception. (a) View of a subplot on the field, (b) MTG file format coding the plant architecture, (c) topological and geometrical plant reconstruction in Xplo, (d) stand reconstruction in Simeo, (e) light interception computed using the MMR library.

radiative balance at different scales (plot, tree, frond, leaflet...) can give information on light efficiency of the architecture of any species, variety or morphotype to be used in agronomic applications, in relation with environmental conditions [23].

Another application of *AMAPstudio* was submitted to the *PMA12* conference: *DRAFt* is a plant growth simulator in *Xplo* based on a minimalist Functional-Structural Plant Model [Taugourdeau O., Barczi J.F., Caraglio Y., Morphogenetical gradients simulation with a minimalist Functional-Structural Plant Model (FSPM)].

V. Discussion

The *AMAPstudio* project offers different levels of services: from a high finition level software suite to edit plant observations, to a powerful environment to implement plant growth simulators. The editing features of *Xplo* and *Simeo* can be used directly to load OPF files (see section III.B) or import files in compatible formats, edit and analyse them. Reading documentations may be needed to help building more complex reconstruction scripts. Finally, a technical assistance is available for the scientists who would like to go further and integrate their own growth model in *Java*.

In case of model integration, the *AMAPstudio* charter [24] must be accepted: it clearly states the roles of the participants: developer, modeller or end-user, and the rules to be respected. It is thus explained that the technical team (developers) may not help designing a model but can help writing the related simulator. A major rule is that scientists (modellers) bringing a model to *AMAPstudio* may share all the source codes with the other members through a common versioning tool. Each member may distribute *AMAPstudio* with only his models to his own partners (end-users) by building specific installers. The charter also precisely indicates which parts are free software, including the *AMAPstudio* kernel and which parts are the property of their authors, especially the integrated growth models.

AMAPstudio presents originalities compared with other approaches. We chose to design interactive software including *Model View Controller* and undo/redo features, quickly accessible for non programmers and students, and adapted for knowledge transfer. We decided to rely on the *Capsis* framework to take advantage of its simulation history manager to help the modellers to better control their models. Furthermore, the project manager makes it possible to build scenarios interactively by combining interventions (e.g. pruning) and growth phases. Finally, we propose solutions to handle very heavy scenes on standard computers.

VI. Conclusion

The *AMAPstudio* software suite can be used for plants architecture modelling at the individual or scene scales. It proposes features to interactively load, view, edit, analyse and reconstruct plants architecture. Plugins can be used and added to add new features, in particular for file loading and exporting, and for connection to other biophysics simulation software like radiative balance or biomechanics.

AMAPstudio software can run on various operating systems thanks to their *Java*-based implementation. This software suite is interoperable with the existing tools of the plant architecture modelling field and offers a user-friendly interface to make its use as simple and intuitive as possible.

Various growth models for single plants or vegetal scenes can be added by candidate modellers who would like to use these simulation capabilities thanks to a free software license and the *AMAPstudio* charter.

A full-featured evaluation version can be downloaded on http://amapstudio.cirad.fr.

Figure 4. An example of a 12 years old simulated palm tree. (a) The simulator computes detailed mock-ups, each containing around 120000 ArchiNodes, (b) which can be planted in staggered rows in Simeo for further biophysics computations.

Acknowledgments

The authors would like to thank all the scientists who took part in the designing of *AMAPstudio* and the writing of this paper, especially Jean-Francois Barczi, Yves Caraglio, Jean Dauzat, Yves Dumont, Patrick Heuret, Hervé Rey and Olivier Taugourdeau.

References

[1] D. Barthélémy and Y. Caraglio, "Plant architecture: A dynamic, multilevel and comprehensive approach to plant form, structure and ontogeny", Annals of Botany, vol. 99, no. 3, pp. 375–407, 2007.

[2] T. Fourcaud, X. P. Zhang, A. Stokes, H. Lambers and C. Koerner, "Plant growth modelling and applications: The increasing importance of plant architecture in growth models", Annals of Botany, vol. 101, pp. 1053-1063, 2008.

[3] P. de Reffye, C. Edelin, J. Françon, M. Jaeger and C. Puech, "Plant models faithful to botanical structure and development", Proceedings of the 15th annual conference on Computer graphics and interactive techniques, pp. 151-158, 1988.

[4] J.-F. Barczi, H. Rey, Y. Caraglio, P. de Reffye, D. Barthélémy, Q. Dong and T. Fourcaud, "AMAPsim: an integrative whole-plant architecture simulator based on botanical knowledge", Annals of Botany, vol. 101, pp. 1125-1138, 2008.

[5] P. Prusinkiewicz and A. Lindenmayer, "The algorithmic beauty of plants", Springer-Verlag, New York, NY, 1990.

[6] O. Kniemeyer and W. Kurth, "The Modelling Platform GroIMP and the Programming Language XL". A. Schürr, M. Nagl, and A. Zündorf, Eds. Lecture Notes In Computer Science, vol. 5088. Springer-Verlag, pp. 570-572, 2008.

[7] C. Pradal, S. Dufour-Kowalski, F. Boudon, C. Fournier and C. Godin. "OpenAlea: a visual programming and component-based software platform for plant modelling", Functional Plant Biology, vol. 35, pp. 751-760, 2008.

[8] J. Weber and J. Penn, "Creation and rendering of realistic trees", Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, pp.119-128, 1995.

[9] Xfrog: http://xfrog.com/

[10] W. Palubicki, K. Horel, S. Longay, A. Runions, B. Lane, R. Mech, & P. Prusinkiewicz. "Self-organizing tree models for image synthesis", ACM Trans. Graph., ACM, 28, pp. 1-10, 2009.

[11] S. Dufour-Kowalski, B. Courbaud, P. Dreyfus, C. Meredieu and F. de Coligny, "Capsis: an open software framework and community for forest growth modelling", Annals of Forest Science, doi: 10.1007/s13595-011-0140-9. 2012.

[12] C. Godin and Y. Caraglio, "A multiscale model of plant topological structures", Journal of Theoretical Biology 191, pp. 1–46, 1998.

[13] E. Gamma, R. Helm, R. Johnson and J. Vlissides, "Design Patterns: Elements of Reusable Object-Oriented Software", Addison-Wesley, 1995.

[14] H. P. Yan, P. de Reffye and C. H. Pan, "Fast construction of plant architectural models based on substructure decomposition", Journal of Computer Science and Technology, Volume 18, Number 6, 780-787, doi: 10.1007/BF02945467, 2003.

[15] The Blender software: http://www.blender.org

[16] The Abaqus product suite: http://www.3ds.com/products/simulia/overview/

[17] O. Taugourdeau, J. Dauzat, S. Griffon, F. de Coligny, S. Sabatier, Y. Caraglio and D. Barthélémy, "Retrospective analysis of fir sapling growth vs. light interception", Proc. 6th International Workshop on Functional-Structural Plant Models (FSPM), September 2010.

[18] J. Dauzat, P. Clouvel, D. Luquet and P. Martin, "Using virtual plants to analyse the light-foraging efficiency of a low-density cotton crop", Annals of Botany, vol. 101, pp. 1153-1166, 2008.

[19] The MOCAF project: http://www.mocaf.net

[20] P. de Reffye, R. Lecoustre, J. Dauzat, S. Ouattara, A. Flori and Y.P. N'Cho, "Modelling of plant architecture. Application to tropical agronomic perennial plants. Particular case of Palmaceae", Oléagineux, vol. 44 (11), pp. 544–546, 1989.

[21] M. A. Elhoumaizi, R. Lecoustre and A. Oihabi, "Phyllotaxis and handedness in date palm (Phœnix dactylifera L.)", Fruits, vol. 57, pp. 297-303, 2002.

[22] F. Hallé and R. A. A. Oldeman, "Essai sur l'architecture et dynamique de la croissance des arbres tropicaux", Masson and Co, Paris, 1970.

[23] J. Dauzat, "Radiative transfer simulation on computer models of Elaeis guineensis.", Oléagineux, vol. 49, pp. 8-90, 1994.

[24] The AMAPstudio charter: http://amapstudio.cirad.fr/doku.php?id=charter