WBCSim: An Environment for Modeling Wood-Based Composites Manufacture

Jiang Shu[a], Layne T. Watson[a], Balazs G. Zombori[b], and Frederick A. Kamke[b]

[a] *Departments of Computer Science and Mathematics, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061-0106 USA*

[b] *Department of Wood Science and Forest Products, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061-0503 USA*

Abstract

This paper describes a computing environment WBCSim that increases the productivity of wood scientists conducting research on wood-based composite materials and manufacturing processes. WBCSim integrates Fortran 90 simulation codes with a Web-based graphical front end, an optimization tool, and various visualization tools. WBCSim serves as an example for the design, construction, and evaluation of small-scale problem solving environments. WBCSim supports six models. A detailed description of the system architecture and a typical scenario of usage are presented, along with optimization and visualization features.

Keywords: Problem Solving Environment; Computing Environment; Wood-based Composite Materials; Visualization; Optimization

1. Introduction

Scientists often conduct experiments involving many parameters, and cannot afford the time and cost to pursue every combination of those parameters. Therefore, they rely on computer modeling and simulation to assist them in exploring the parameter space. However, most of the time, the modeling and simulation codes are developed in-house and have many limitations (relevant commercial codes may not exist or may not be affordable). Often not only is the code tied to a fixed computing environment, making it hard to access or exchange with other research groups, but it is also very common that the code is not integrated with any visualization and optimization tools. Moreover, since the developers of a code are often the users of that system, the result is poor documentation and an unsophisticated user interface, so that only the developers themselves can utilize the system efficiently. This situation obviously hinders the productivity of many research groups. This paper describes a computing environment WBCSim that increases the productivity of wood scientists conducting research on wood-based composite (WBC) materials and manufacturing processes. WBCSim integrates Fortran 90 simulation codes with a Web-based graphical front end, an optimization tool, and various visualization tools.

Goel et al. [27] described an early version of WBCSim in 1997. Since then, WBCSim has evolved, taking different approaches to its architecture, adding more sophisticated models, and switching from experiment-oriented to manufacturer-oriented. However, the original goals remain the same: (1) to increase the productivity of WBC research and manufacturing groups by improving their software environment, and (2) to continue serving as an example for the design, construction, and evaluation of small-scale problem solving environments (PSEs). In this paper, a detailed description of the current system architecture, six different WBC models, and a typical scenario of usage are discussed, along with optimization and visualization features.

Goel et al. [27] give a more complete description of what constitutes a PSE and why WBCSim is a PSE. In general, a PSE provides an integrated set of high-level facilities that support users engaged in solving problems from a proscribed domain [24]. Also a PSE commonly addresses the following issues: Internet accessibility to legacy code, visualization, experiment management, multidisciplinary support, collaboration support, optimization, high performance computing, usage documentation, preservation of expert knowledge, recommender systems, and integration [54]. WBCSim qualifies as a PSE because it makes legacy simulation codes available via the WWW, is equipped with visualization and optimization tools, is multidisciplinary in supporting different disciplinary simulation models, and has experiment management, collaboration support, and high performance computing being added.

The paper is organized as follows. Section 2 reviews some related work in PSE and WBC computer-based systems. Section 3 describes the six simulation models currently supported by WBCSim. Section 4 elaborates on the WBCSim user interface, the model interface similarities, and also the unique aspects of each. Section 5 explains the various architecture layers of WBCSim, and also the new WBCSim telnet communication connection scheme. Section 6 demonstrates a typical simulation scenario. Section 7 describes the WBCSim optimization tool. Section 8 explains the visualization tools involved. Finally, Section 9 outlines some future directions for WBCSim.

2. Related Work

There are many problem-specific PSEs developed for various application domains, and several computer-based mathematical models were developed to solve particular problems in the wood-based composites industry. However, no work known to the authors addressed the integration of mathematical models with a PSE environment. WBCSim is a novel approach to fill this gap and provide a valuable tool for the wood-based composites industry.

Goel et al. [27] describe some early PSEs such as ELLPACK and its descendents [5] for solving two and three dimensional elliptic partial differential equations (PDE), SCIRun [49] developed to interactively compose, execute, and control a large-scale computer simulation, and Linear System Analyzer [11] for manipulating and solving large-scale sparse linear systems of equations. Since then, many new PSEs have been introduced. Gismo [12], created at Washington University, is an object oriented Monte Carlo package for modeling all aspects of a satellite's design and performance. It has played a significant role in the design of the Gamma Ray Large Area Space Telescope, the successor to the Compton Gamma Ray Observatory that was launched into space in 1991 for exploring the gamma ray portion of the electromagnetic spectrum in astrophysics. VizCraft [26], developed at Virginia Polytechnic Institute and State University, provides a graphical user interface to a widely used suite of analysis and optimization codes to aid aircraft designers during conceptual design of a high-speed civil transport. VizCraft combines visualization and computation, encouraging the designer to think in terms of the overall problem-solving task, not simply using the visualization to view the computation's results. There is also a computing environment developed by Chen et al. [17] that combines particle systems, rigid-body particle dynamics, computational fluid dynamics, rendering, and visualization techniques to simulate physically realistic, complex dust behaviors useful in interactive graphics applications for education, entertainment, or training.

Common practice in the wood composites industry is to use mathematical models to describe the pertinent relationships between various manufacturing parameters and final composite properties. The most influential stages of composites production are the mat formation and the hot-compression processes, therefore the modeling effort is concentrated on these two areas. However, most of the mathematical models were implemented as command line based programs with standard text input and output, and only a few attempts were made to create more user friendly environments.

A mat formation model is required to establish the critical relationships between the structure of the composite and the dynamic change of certain physical properties during the mat consolidation. One of the most notable examples is commercially available mat formation software developed by Forintek Canada Corporation. The software, based on Dai's mathematical model [20], incorporates geometric probability theory and simulation techniques for describing the characteristics observable on the surface of the flake mat. The basis of the model is an idealized, randomly formed, flake layer network, where the dimensions of the flakes are uniform, and the number of the flakes is limited in a manner that the sum of the areas of the flakes is equal to the area of the layer. Given these assumptions, several properties of the flake network are mathematically definable using probability distributions and random field theory. The composite mat is constructed from these single layers of flakes. The model can calculate the horizontal density distribution of the mat, the flake-to-flake contact area, and the void volume fraction. The commercial software version of the model is only accessible for a fee.

Another mat formation model created by Lu [45], called WinMat, works as a stand alone application in all Windows 9* operating systems. Besides the major objective of WinMat to generate commands for an industrial robot, which is capable of putting together mat structures from uniform size flakes, several additional features were included in the program. It can create a single or a three-layer OSB structure from flakes either with fixed dimensions or with dimensions described by a normal, uniform, Poisson, or binomial distribution. Additionally, the orientation of the flakes can follow a uniform or a vonMises distribution. Although the interface to the program is easy to use, several aspects of the model are deficient. The density of the flakes is constant, not allowing mats built from mixed wood species. Additionally, the number of the deposited flakes in each layer is controlled by the actual flake number instead of the more realistic cumulative flake weight.

Less computer based then the above two, another mathematical modeling approach to characterize the spatial structure of randomly formed wood flake composites is referred to as the edge model [41], [42], [43]. This model subdivides the mat into imaginary flake columns of finite size. The geometrical properties of the columns are determined by experiments and measured at the edge of the mat. Probability density functions are then fit to the data. The columns are reproduced in the simulation model by sampling the fitted probability distributions. Although the edge modeling approach is unique, no attempt was made to include a graphical user interface or advanced result visualization.

In order to improve on the predictions of the previous models, Zombori et al. [64] developed a more realistic description of the mat structure, where the mat formation includes the geometry of the wood elements as random variables and certain limitations can be imposed on the orientation of the elements. Additionally, a density value is assigned to each individual strand, allowing the simulation of the horizontal density distribution of composites made out of mixed species. The mat formation process model was developed based on data obtained from commercial strand measurements and was validated by comparison of the simulated horizontal density distribution to a measured horizontal density distribution. None of the published models prior to this one could predict the real structure and horizontal density distribution of a commercial, three-layer OSB mat, based on geometry and density measurements of industrial strands.

A hot-compression model describes the heat and mass transfer during the hot-compression manufacturing process of wood-based composite panels. The first model was published by

Humphrey [33] in 1982. Based on this original model a sophisticated model, including steam injection, was put together by Haselein [30]. This was further improved by Thömen [57], who modeled the internal environment in a continuous hot-press. Another model was developed for Simpelkampf Gmbh Corporation, one of the largest hot-press producers in Germany, but this model is proprietary.

The literature is extensive on the modeling of the internal conditions during the hot-compression stage of wood-based composite manufacture. The models are generally based on well-known energy and mass equilibrium equations, but include different heat and moisture transport mechanisms. A survey of the hot-compression model literature is in [63]. The important modeling efforts, which either were not computer based or did not include an integrated environment, include Bowen [10], Bolton and Humphrey [6], Humphrey [34], Bolton et al. [7], [8], [9], Harless et al. [29], Kayihan and Johnson [39], Carvalho and Costa [13], Stanish [56].

All these models had inherent limitations; either they were one-dimensional, or gross simplifications were made about the transfer mechanisms. Additionally, no attempt was made to integrate the mathematical heat and mass transfer models within a PSE. The hot-compression model of Zombori [63] allowed a comprehensive three-dimensional description of the heat and mass transfer phenomena during the mat consolidation. All conceivable simultaneous heat and mass transfer mechanisms were considered, together with phase balancing sorption isotherms. By numerically solving the governing partial differential equations, the evolution of moisture and temperature profiles in the vertical midplane of the board could be predicted.

The oriented strandboard mat formation and hot-compression simulation models provided in the PSE WBCSim are a logical continuation of the previously described models. Generally, both of the models implement a wider range of functionalities than any of the mathematical models used in the wood-based composites industry. WBCSim supports these models with intuitive graphical user interfaces, from a large selection of model input variables to sophisticated visualization and optimization techniques.

3. Simulation Models

WBCSim currently supports six simulation models (CMA and CMA-NEW are the same simulation codes with different output visualizations) that help wood scientists studying wood-based composite material manufacturing. Each of these models is described next.

3.1. Rotary Dryer Simulation (RDS)

The rotary dryer simulation model assists in the design and operation of the most common type of system used for the drying of wood particles [37], [38]. The rotary dryer is an integral part of the processes to manufacture particleboard and strandboard products. It consists of a large, horizontally oriented, rotating drum (typically 3 to 5 m in diameter and 20 to 30 m in length). The wet wood particles are mixed directly with hot combustion gases, in a co-current flow pattern, at the inlet to the rotating drum. The gas flow provides the thermal energy for drying, as well as the medium for pneumatic transport of the particles through the length of the drum. Interior lifting flanges serve to agitate and produce a cascade of particles through the hot gases.

The RDS model consists of coupled material and energy balance equations for each segment along the length of the drum. Each drum segment is defined by the cascading pattern of particle travel. The segment, or cycle, begins when a particle drops off a lifting flange and falls to the bottom of the drum. This is followed by travel along the periphery of the drum, where the particle is caught by a lifting flange. The segment ends when the particle attains its maximum angle of repose and tumbles off of the lifting flange. The user must supply the inlet conditions of the hot gases and wet wood particles, as well as the physical dimensions of the drum and lifting flanges, flow rates, and thermal loss factor for the dryer. The RDS model predicts the particle moisture content, temperature, gas composition, and energy consumption.

3.2. Radio-Frequency Pressing (RFP)

The radio-frequency pressing model [52] was developed to simulate the consolidation of wood veneer into a laminated composite. The energy needed for cure of the thermosetting adhesive is supplied by a high-frequency electric field. Radio-frequency pressing is commonly used for thick composites and for nonplanar laminated composites. The model may be used to help design alternative pressing schedules.

The RFP model consists of a collection of nonlinear PDEs that describe the heat and mass transfer within the veneer layers. The primary variables are temperature and moisture content. The moisture content is further divided into three phases: bound water, liquid water, and water vapor. These water phases must satisfy a criterion of local thermodynamic equilibrium as represented by a nonlinear algebraic equation. The model is one-dimensional, with a fixed resistance to heat and mass flux at the boundary. The RFP model predicts the time-dependent temperature and moisture content profiles in the veneer layers, as well as the extent of adhesive cure. Among the user-supplied input data are the initial density, thickness, moisture content, and temperature of the veneer, as well as the electric field strength.

3.3. Composite Material Analysis (CMA)

This model is being replaced with the new version in Section 3.6.

3.4. Oriented Strandboard Mat Formation (OSB)

The mat formation model creates the three-dimensional spatial structure of a layered wood-based composite (e.g., oriented strandboard and waferboard) and calculates certain mat properties by superimposing a mesh on the mat structure. The model is based on a Monte Carlo simulation technique. It is assumed that the dimensions of the constituents (e.g., strands) of the mat follow certain probability distribution functions. The parameters of the probability distribution functions have to be determined on a subsample by measurement. The user inputs the number of layers. The total number of strands is determined by specifying that number, or specifying the weight, volume, or area of each layer. The probability distribution functions of the length, width, thickness, and density of the strands are given by the user.

Output from the mat formation model includes the horizontal density distribution, proportion of void space, and potential bonded surface area. Output data is available in a numerical format and two- or three-dimensional visualizations. The output from the mat formation model is required for the hot compression model.

3.5. Hot Compression (HC)

The hot compression model simulates the mat consolidation and adhesive cure that occurs during industrial hot-pressing of wood-based panels. The model is based on fundamental engineering principles and uses the output from the mat formation model to establish the starting spatial structure of the mat. Six primary variables are considered: mat density, air content, vapor content, bound water content, and temperature within the mat, and the extent of the cure of the adhesive system characterized by the cure index. The heat is transported by conduction and bulk flow, while the water phases are transported by bulk flow and diffusion. A nonlinear viscoelastic relationship was used to describe the compression behavior of the mat. This relationship separates the geometric nonlinear response of the cellular structure of the wood elements from the linear viscoelastic response of the wood cell wall polymers. The behavior of the cellular structure is modeled with a modified Hooke's Law. The viscoelastic properties of the flakes are described by the time-temperature-moisture equivalence principle of polymers.

The resulting differential-algebraic system of equations is solved by a semicontinuous finite difference method. The spatial derivatives of the conduction terms are discretized according to a central difference scheme, while the spatial derivatives of the bulk flow terms are discretized according to an upwind scheme. The resulting ordinary differential equations (ODEs) in the time variable are solved by DDASSL, a freely available differential-algebraic system solver. The model predicts temperature, moisture content, partial air and vapor pressures, total pressure, relative humidity, extent of adhesive cure, and the spatial density profiles within the mat. A set of three-dimensional graphical profiles illustrate the evolution of these variables with time, in the thickness and width dimensions of the mat. The model assists users in the understanding of the interacting mechanisms involved in a complex production process. The model may also be used to optimize the hot-pressing parameters for improved quality of wood-based panel products, while minimizing processing cost.

3.6. Composite Material Analysis—New (CMA-NEW)

The composite material analysis (CMA) model was developed to assess the stress and strain behavior and strength properties of laminated materials (e.g., plywood and fiber-reinforced composites). The user defines the layer sequence of the laminate. The graphical user interface was designed to allow easy specification of the material, thickness, and orientation at each layer. The mechanical and failure properties of the layer materials may be selected from a predefined list. The model can perform design and analysis functions, where the user either defines the loading condition, or defines the deformation.

The calculations are based on the classical lamination theory (CLT) and the Tsai-Wu failure criteria. The "Design" and "Analysis" models calculate the induced stresses and strains in the primary laminate (x, y) and principal fiber $(1, 2)$ directions within each layer based on the CLT, and check for the integrity of the layers of the laminate by the Tsai-Wu failure criteria. In the design mode, the model calculates the stresses and strains caused by the combination of different loading conditions, such as tension, moment, torque, or shear. In the analysis mode, the normal and shear stresses, together with the strains and curvatures induced by a user-defined deformed shape, are calculated. The model predicts the tensile strength, bending strength, and shear strength of the composite material.

4. User Interfaces

WBCSim is Web-based; therefore, its user interface is composed of a Web browser and Java applets. A user launches the WBCSim Web page from a browser window, and then invokes applets from the Web page. The very first applet allows the selection of a simulation model. From that point on, all user interaction with the system is via applets. There are some common features among all the model interfaces (see Fig. 1). At the top left corner of a model interface, there is a "Usage Instructions" button, whose pop-out window explains the model behavior at an abstract level, and also gives general directions about how to use the model. The middle region of each model interface is model-dependent, allowing a user to specify simulation parameters in any number of text boxes, drop down lists, radio buttons, or plain buttons, which possibly open pop-out windows

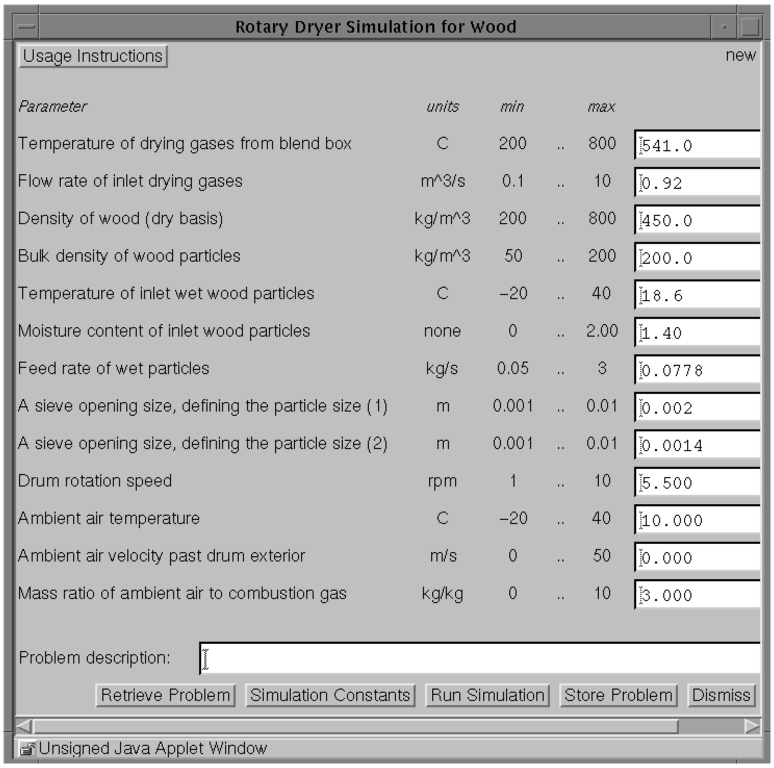

Fig. 1. The RDS model user interface.

for editing even more simulation parameters. Near the bottom of the interface, there is a long text box where a user can describe the simulation; this description is saved when the simulation data is saved. At the very bottom of an interface, there is a row of buttons, which control model-level actions for a simulation. Among those buttons are the four: “Retrieve Problem”, “Run Simulation”, “Store Problem”, and “Dismiss”. The “Store Problem” button is used to store the current set of input values (along with the simulation description text), which can be retrieved later using the “Retrieve Problem” button. Clicking on the “Run Simulation” button fires input values defined in the interface through a telnet connection to the WBCSim server, which in turn calls a UNIX or Perl script to execute compiled FORTRAN code, and additional optimization or visualization codes. (The details of this communication process will be described in the next section.) The “Dismiss” button dismisses the interface without saving any input. Other than these four main control buttons, there are some other buttons varying from model to model that may appear in the same row, such as the “Simulation Constants” button (lists some simulation parameters not accessible by the user), and the “Set Default” button (sets all the model variables to a predefined set of values). The results from running a simulation model are usually available in both textual (normally tables of numbers) and graphical (VRML files, GIF files) forms, and displayable in browser windows.

The RDS (Fig. 1) and RFP models have the simplest user interfaces in WBCSim. The user simply enters values for various input parameters through text boxes, then runs the simulation. There are two CMA models—“old” and “new”. Depending on the context, “model” refers to the computer simulation code itself, or the interface to that code. The old CMA model supports five calculations: “analysis”, “design”, “tensile strength”, “bending strength”, and “shear strength”.

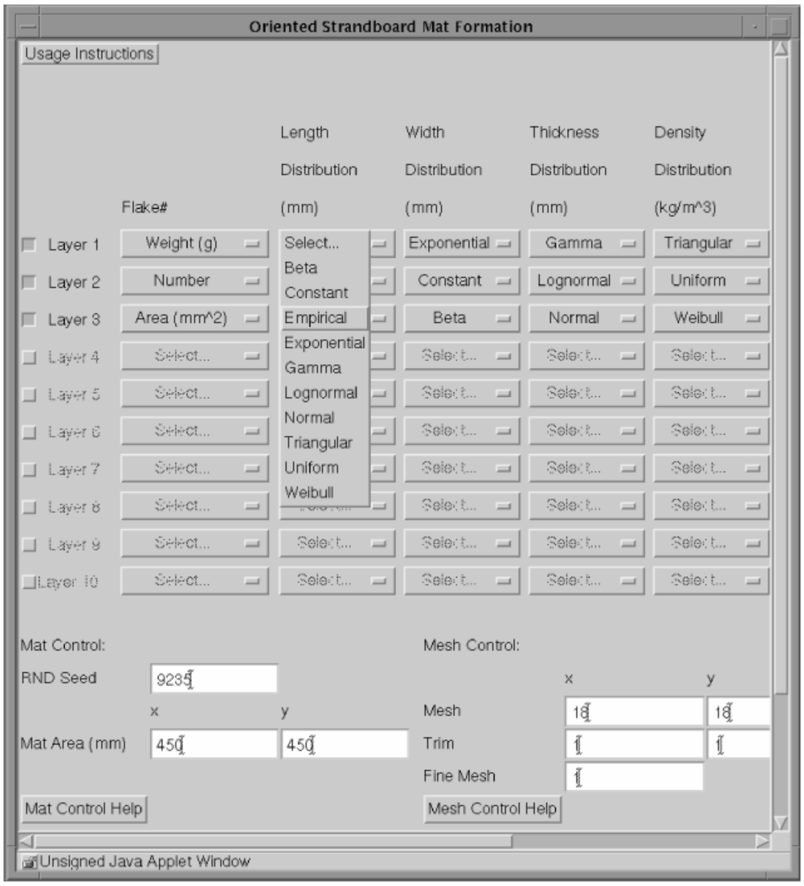

Fig. 2. The OSB model user interface.

When the user selects any of the five, the user is locked to that calculation and cannot switch from one calculation to another without returning to the beginning. Also, in each calculation interface, many parameters are fixed or assumed, whereas in the CMA-NEW model, this is greatly improved by providing text boxes for those previously fixed parameters. The CMA-NEW model added a new calculation "general strength", and integrated all the calculations into one interface, allowing the user to select a dynamically loaded calculation from a drop down list. The CMA-NEW model also included two sets of defaults, structure and loading (both in drop down lists). This new integrated interface allows the user to cross-reference different sets of defaults by setting one structure parameter set and then running any calculation with any loadings.

The OSB model (Fig. 2) is similar to the CMA model with rows of input data representing layers of the wood composite. The user activates a layer by selecting the leftmost checkbox for that layer. Also as in the CMA model, the OSB model provides a set of defaults rather than a single default like most of the other models. However, the rest of the OSB model is more complex than that of CMA, since it involves many pop-out windows when the user chooses an item from a drop down list or clicks on a button. For example, selecting the "Empirical" distribution in the "Length Distribution" drop down list, a pop-out window containing forty-two text boxes appears, allowing the user to define the number of intervals and each interval's starting and ending points. The OSB

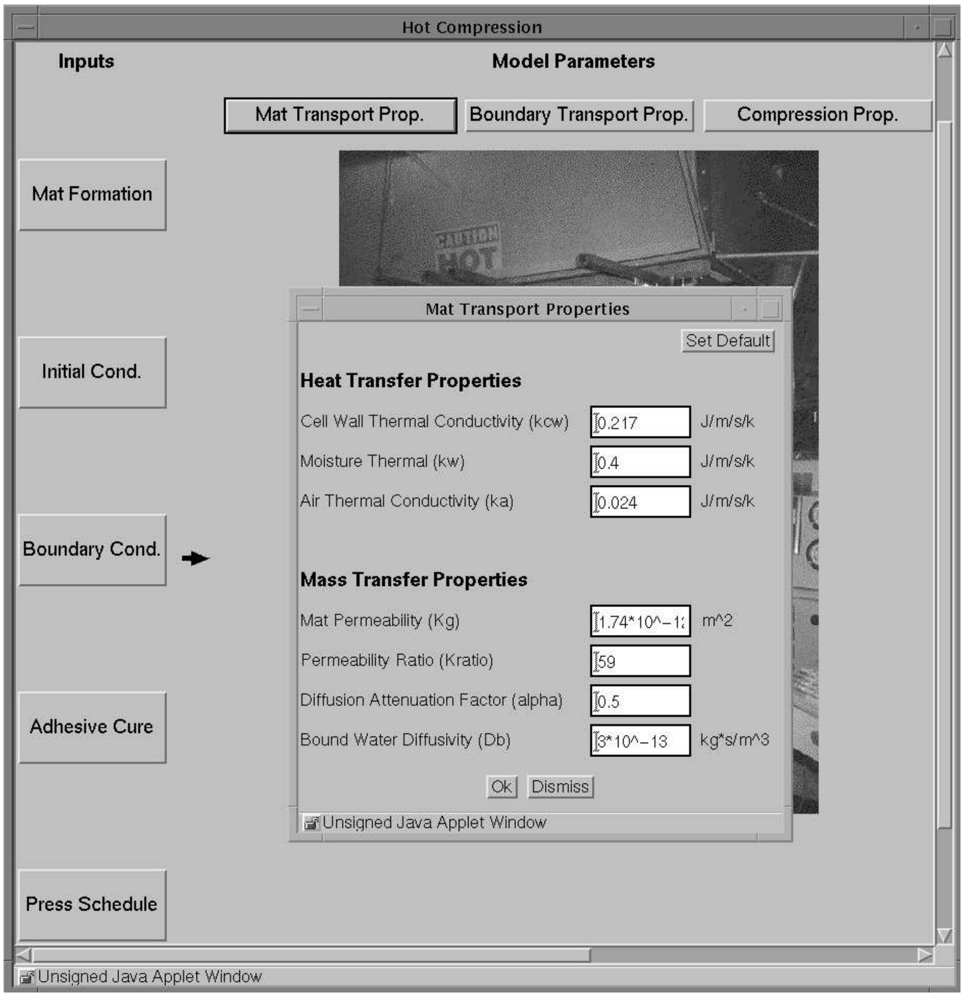

Fig. 3. The HC model user interface.

interface can also save the values defined in pop-out windows, which means those pop-out windows can be closed and then opened again for editing.

The HC model (Fig. 3) interface differs from the other models' interfaces by dividing the simulation parameters into three groups partitioned into columns: (1) model input, (2) model execution specification, and (3) model output. The first column contains buttons for specifying the parameters related to material, initial condition, boundary condition, adhesive cure, and press schedule. The second column defines the HC model execution by specifying mat transport properties, boundary transport properties, and compression transport properties. The last column specifies the output such as color or black and white, the time interval for generating frames for animation, and other parameters determining which data/image files are generated.

5. Software Architecture

The current software architecture of WBCSim follows the three-tier model described by Goel et al. [27], for which the tiers are (1) the developer layer—legacy simulation codes and various

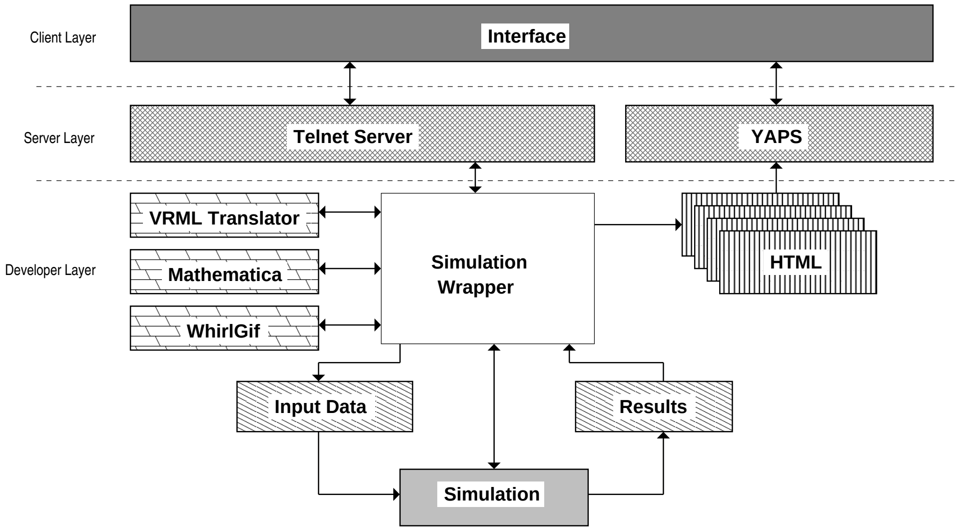

Fig. 4. WBCSim architecture overview.

optimization and visualization tools running on the server, (2) the client layer—user interface, and (3) the server layer—telnet server and a custom shell. These layers are shown in Fig. 4.

5.1. Developer Layer

As its name suggests, the developer layer consists of legacy programs created by researchers to model WBC materials and manufacturing processes. These legacy programs are the heart of WBCSim. In general, WBCSim supports legacy programs written in any programming language as long as the program takes its input parameters from UNIX stdin. The input parameters can be a few numbers or a large data file. In particular, WBCSim supports six FORTRAN 77 and Fortran 90 simulation programs corresponding to the six models described in Section 3.

While each Fortran program has its own input format (stdin could be redirected to a file), the server layer communicates data with the developer layer via strings of parameters separated by white space. Therefore, in order to cope with this string format, each legacy program is "wrapped" with a customized Perl script. The script receives this string of parameters from the server, and converts those parameters into an appropriate format for that Fortran program. Then the script calls the legacy program into action feeding it the input, invokes any required optimization and visualization tools, and finally packs all the Fortran output in HTML files and passes their URLs to the server. With this architecture, the developer layer is independent of the other layers, which makes the process of designing, and integrating new simulation codes relatively easy.

The developer layer also includes optimization and visualization tools to maximize the simulation's value to the user. Optimization and visualization tools will be described in Sections 7 and 8, respectively.

5.2. Client Layer

The client layer is the only layer visible to end-users and typically the only layer running on the local machine. Besides the user interfaces described in Section 4, the client layer also contains viewers for one of the visualization tools—the VRML Translator. WBCSim requires a VRML 2.0 viewer for the RFP model and a VRML 1.0 viewer for the CMA model. Both VRML viewers serve as plug-ins to the Web browsers.

The client layer also handles the communication with the server layer. After the user enters all the necessary parameters and triggers the "Run Simulation" button, the client sends those parameters along with a request for executing the corresponding simulation to the server layer via a telnet connection. When the simulation terminates, the client takes the URLs returned by the server layer, and directs them into the user's browser. Of course, the client may also send requests for save and retrieve operations.

The telnet connection method replaces the old Javamatic server and socket communication paradigm described in [27], and facilitates setting up WBCSim guest accounts on the server machine that do not require full account privileges. The guest account and normal user accounts (for commercial, paying users) permit users to save and track their different simulation runs, and protect their data from being removed or overwritten by others. The client side of this telnet connection manages the pool of connections and provides a means for executing remote commands on the server.

5.3. Server Layer

The server layer, by separating the legacy simulation codes from the user interface, is the key to how WBCSim can run a text-only application from a Web browser. The server layer consists of two components, a telnet server and a custom shell to facilitate this server-client communication.

The telnet server is not a replacement for a standard telnet server, implementing only enough of the telnet protocol to work with the WBCSim telnet client. The telnet server supports guest and regular logins and all the operations provided by the previous Javamatic server, which could direct execution of multiple simulations and accept multiple requests from the client concurrently [27].

Yet Another PSE Shell (YAPS) is a simple Perl script that the client invokes when it logs in via the telnet connection. Therefore, the client talks to this shell instead of the UNIX login shell of the account. Among the commands that YAPS supports, the following four are most important.

(1) The *store* command is used to permanently store a problem (input parameters, text description, simulation output) on the server. Upon request, a temporary archive file is created. The problem description and simulation parameters are written to this file. Notice that it is possible to store problem parameters without running the simulation, in which case there is no simulation output.

(2) The *remove* command removes one or more stored problems. It actually removes files from the file system, so the results are irreversible.

(3) The *load* command retrieves all stored problems from a directory. This command prints a collection of problem IDs (each problem definition or simulation run has a unique problem ID), parameters, descriptions, and URLs, which could be picked up by the telnet client.

(4) The *clean* command deletes temporary files. Normally cleanup is requested when the "simulation results" window is dismissed. However, cleanup requests are only executed if there is an idle connection. If no idle connection exists, cleanup requests are queued until a connection becomes available. The goal of this policy is to clean up only when doing so incurs no extra connection. Making an extra connection involves a lengthy bootstrap process and possibly passing the user password again.

There is no command to run a simulation because the simulation wrapper can be invoked directly by YAPS based on the requests coming from the telnet client.

6. Simulation Scenario

Describing a typical usage scenario of WBCSim is instructive. Consider research into the properties of oriented strandboard products, which would use the OSB model. First, a scientist collects a subset of the face and core layer flakes from an industrial production line. The scientist measures the geometry of the flakes and the weight by using an image analysis system and a scales, respectively. The flake property data sets are then used to estimate the statistical probability density functions of the flake properties. Then, the scientist opens a browser window and launches WBCSim. The guest account is the default login option, which stores all the input and output data in a temporary directory that will be cleaned periodically. The scientist selects the OSB model from a dropdown list of all the available models, and launches the OSB model interface (shown in Fig. 2). By specifying the probability density functions of the flake properties obtained offline earlier, the scientist can recreate the spatial structure of multiple layers of flakes, each with various parameters defining the number of flakes, length distribution, width distribution, thickness distribution, density distribution, orientation, flake color, and number of flakes shown. The latter two are solely for visualization.

Next, the input parameters (Boolean, numeric, or alphanumeric) are sent to the WBCSim server as a long string (parameters are separated by white spaces) via the telnet connection. Then the OSB wrapper converts this string into a data file designated for the OSB Fortran 90 simulation program. Also, since the OSB simulation code has its own text-based user interface taking input from stdin, a temporary file is generated to contain all the appropriate commands, and stdin is redirected to this temporary data file for the simulation. Then the OSB wrapper calls the OSB simulation code with this temporary file as stdin along with the properly formatted data file.

When the simulation code is executing, the OSB wrapper listens to the simulation output stream for strings indicating execution milestones. The OSB wrapper uses the standard error stream for sending these messages to the client, because Java buffers the standard output stream until the process terminates, while the contents of the standard error stream are sent immediately. The OSB wrapper also sends other messages (such as when visualization or optimization tools are invoked) to the client, displaying the simulation status to the scientist. However, due to network delay, a group of messages generated at different times can arrive at the client at the same time, in which case the client will only display the latest message and discard any old ones.

When the OSB Fortran 90 simulation program terminates, the OSB wrapper calls Mathematica to read the simulation results and generate plots with various Mathematica commands such as ListContourPlot, ListPlot3D, MultipleListPlot, and Graphics3D. Mathematica also converts these internal graphics data structures into GIF format so they can be viewed in a browser. Fig. 5 shows a three-dimensional visualization of a three-layer random flake mat created with Graphics3D by Mathematica. Finally, the OSB wrapper embeds these GIF files in HTML files and returns the URLs of these HTML files to the client.

Upon closing the interface windows, the scientist has a choice of storing (archiving) this particular run's input values and results, or discarding this run (the default). Based on the user's request, the YAPS shell will take the proper actions as described in Section 5.

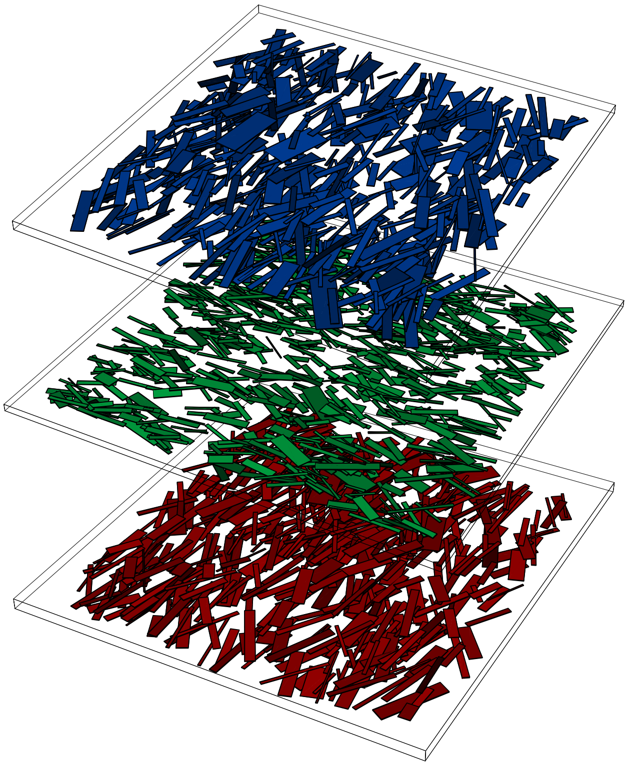

Fig. 5. Visualization of the three-layer random flake mat: the area of the mat is 450×450 mm. Only a fraction of the flakes are shown, and the shading designates the density of the flakes.

7. Optimization

Scientists typically use a PSE like WBCSim iteratively with some design or system behavior goal in mind. Their exploration is by trial and error, guided by intuition that improves as more computational experiments are performed. Mathematical optimization is a powerful tool for automating this tedious search process. Currently the optimization tool provided with WBCSim is the program DOT (Design Optimization Tool) [59], based on sequential quadratic programming and the method of feasible directions. DOT is a robust sophisticated FORTRAN 77 subroutine for nonlinear constrained optimization, and is widely used in engineering.

WBCSim supports two models, RDS and RFP, that are linked to DOT. Each has a different interface for optimization than for analysis. Comparing the interface without the optimization option to the one with optimization, the former provides only one text box for every simulation parameter, while the latter provides three, one for specifying the initial value for that parameter, one for specifying the maximum value, and one for specifying the minimum value. The optimization interface also allows the user to specify which parameters are fixed and which are to be varied. The optimization interface also supplies a number of predefined objective functions (typically computed physical quantities, such as temperature or pressure), for which the user can specify either maximization or minimization. The optimization interfaces necessarily provide parameters to control the convergence of the optimizer, such as “Maximum absolute change in the objective function”, “Maximum relative change in the objective function”, “Relative finite difference step for gradients”, and “Maximum number of iterations”.

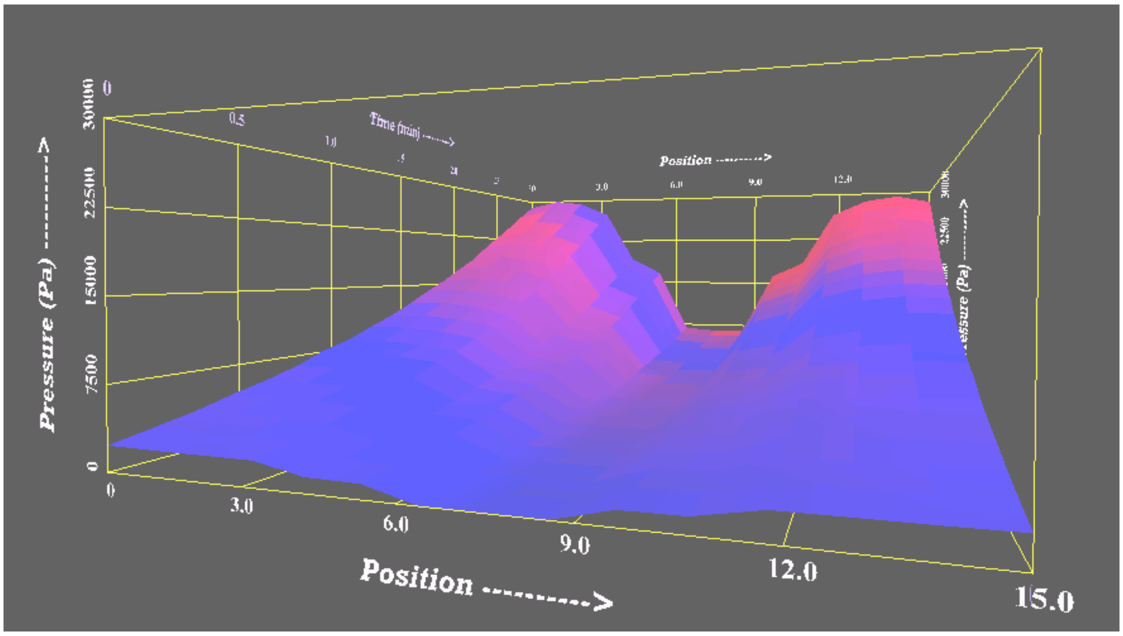

Fig. 6. VRML pressure graph showing the variation of pressure with increasing distance from the laminate surface and with time (the receding axis).

8. Visualization

In principle, WBCSim supports the addition of any visualization tools to convert simulation output to a format meaningful to the user. However, it does require the user to have the corresponding viewer for the format of that visualization result installed and accessible from the Web browser. An example is the VRML viewer described in Section 5.2.

WBCSim currently use three visualization tools: VRML [2], Mathematica [62], and the UNIX utility WhirlGif. With a VRML viewer, the output converted to VRML can be viewed from various directions in the three-dimensional viewspace. Mathematica is only used to generate static two or three-dimensional graphs of simulation results. WhirlGif is a GIF to animation converter, which simply picks up the GIF format frames generated by Mathematica and converts them into an animation file, also in GIF format.

VRML was chosen to interactively visualize some WBCSim simulation output, because VRML is a well-accepted standard for three-dimensional visualization, available on many platforms, easy to use, and easily generated. A custom VRML translator was built to provide better control over the description of the three-dimensional output than third party translators could provide. WBCSim generates VRML code for the RFP and CMA model output. For the RFP model, the output data is a matrix of numbers where each row contains two independent variables and the corresponding dependent variable. This three-dimensional data is conveniently modeled as a VRML ElevationGrid. Each row in the matrix can be represented either as a point or a square that is colored based on its height, where blue denotes the lowest point on the grid, red the highest, and the rest are colored by linearly interpolating between blue and red (HSB) hues, as shown in Fig. 6. The CMA model tests a composite for strength based on its material properties, thickness of layers, and orientation of the fibers in each layer. Therefore, the VRML translator is used to decide which layers have failed by looking at the FORTRAN-generated output files. Each time a layer fails, a new VRML model is generated containing information about the forces on each layer at that instant. Failed (inactive) layers are displayed in red, whereas active layers are displayed in black.

Mathematica was chosen to generate static two or three-dimensional graphs of WBCSim simulation results, because Mathematica has sophisticated graphics tools, and is commonly used by wood scientists for graphic representations. When Mathematica was first used in conjunction with WBCSim, the Mathematica commands were hard coded (in Perl) in the wrapper. Every time a simulation was invoked that required Mathematica's services, the wrapper would write the Mathematica commands into a text file embedded with the dynamically generated filenames for the GIF output files (using the simulation ID) . Then, the wrapper would give this file as input to Mathematica, which would generate the GIF files accordingly. However, as more complex simulations such as CMA-NEW were added to the system, the Mathematica input file became large and unwieldy. Compounding this size problem was the fact that every time the Mathematica code changed, the Perl wrapper also had to be changed. The current WBCSim approach keeps the Mathematica code as a blueprint external to the wrapper. Then in every simulation run, the wrapper makes a copy of that blueprint code, uses *sed* to insert the dynamic graphics output filenames, and then feeds it directly to the Mathematica kernel. Fig. 5 (explained in Section 6) is an image file generated by Mathematica from the OSB model.

Since some simulation results, such as the CMA-NEW model's stress and strain distributions, are time series, WBCSim also provides time animation of those results by using WhirlGif. WhirlGif is a small UNIX utility for generating a single animated GIF image file from a sequence of GIF image files. The reason Mathematica is not used for generating those animation files is that the time to generate an animation in Mathematica is significantly longer than the time by using WhirlGif. Because the number of frames may change from one simulation run to another, the wrapper reads the number of frames from a data file. Then the wrapper constructs a command like "`WhirlGif -loop 1000 frame1.gif frame2.gif frame3.gif frame4.gif ... > animation.gif`", where `frame*.gif` are the frame images generated by Mathematica. Then this WhirlGif command is fired off to the UNIX shell by the wrapper to generate the animation file. By embedding each frame into a HTML file, another option is also provided whereby the user can use hyperlinks such as "Next" and "Previous" to view the frames one at a time.

9. Future Work and Conclusions

Of the PSE characteristics mentioned in Section 1, experiment management, collaboration support, and high performance computing are the most appealing at the current stage of WBCSim development.

9.1. Experiment Management

Any given simulation run is just an evaluation of a single point in a multidimensional space, and optimization can search this space for an optimal point with respect to an objective function [54]. While WBCSim supports automated optimization, often it is not possible to precisely articulate an objective function, and there may be multiple conflicting objectives. Furthermore, design trends and data patterns are unlikely to be revealed by a few isolated runs or by automatic optimization. What is required is that the results of simulation runs be stored automatically in a systematic way, which permits annotating the parameters that define a run as well as its results, and further allows searching, comparison, and other data mining of a data base of numerical experiments. Toward this end, an open source database—PostGres—is currently being integrated with WBCSim. A sophisticated experiment management subsystem should significantly enhance the utility of a PSE like WBCSim.

9.2. Collaboration Support

Even though the current telnet server and YAPS allow user/guest logins and multiple users at multiple workstations working on WBCSim simultaneously, more sophisticated real-time collaboration support (e.g., Sieve [35]) is needed. This would allow multiple geographically distributed users to jointly modify the same model interface, and concurrently view and annotate the same simulation output. Other features indirectly supporting collaboration are user control over where their data is stored (locally on the client machine vs. remotely on the server), and support for exchanging data with other researchers.

9.3. High Performance Computing

Simulation codes used by wood scientists often require access to significant computing resources such as a parallel supercomputer. WBCSim is no exception. The more advanced models in WBCSim, such as the CMA-NEW Fortran 90 core or its visualization/optimization tools, can take hours to run on a fast (AXP 21064) workstation. By utilizing either parallel or distributed computing, WBCSim can reduce the turnaround time to minutes, making interactive use feasible even for complex three-dimensional models. A planned next step is to use a 200 node Beowulf cluster as a compute engine for WBCSim, transparent to the user, when a requested simulation can be estimated to require such resources.

9.4. Conclusions

WBCSim has evolved steadily from a prototype PSE intended as a tool for computer science PSE research and a Web-based interface to a few legacy computer programs, to a manufacture-oriented near commercial quality PSE seriously used by wood science researchers in industry and academia. Since interesting computational capabilities are still lacking (cf. Sections 9.1–9.3), WBCSim will remain an object of computer science research for some time to come. Yet the interfaces, models, and output visualizations are now good enough to be used as production tools by the wood scientists. The directions in which computer scientists would like to take WBCSim (collaboration, data mining, grid computing) are quite different from the directions the wood scientists would prefer for WBCSim (new models, refining existing models, more interface and visualization options). The present layered architecture of WBCSim supports this dichotomy well.

From the user perspective, by providing a Web-based graphical interface to legacy command-line FORTRAN programs, WBCSim leverages the accessibility of the Web to make these simulations easily usable by scientists and engineers away from their laboratories. Via the concept of remote compute cycles WBCSim frees users from software installation on the client machine. The optimization tool and the simulation output visualizations increase the intellectual utility of the legacy simulation codes.

From the developer perspective, the simulations are wrapped and isolated from users. This has the advantage that if there are changes to a legacy simulation program, such as bug fixes, recompilation with new libraries or a new compiler, or implementing a different algorithm, the new program can be installed in WBCSim without additional work. Adding a different visualization tool or a new optimization algorithm is easy since only wrappers need to be modified.

The original stated goal of WBCSim, to provide "an integrated set of facilities allowing wood scientists to concentrate on high-level problem solving rather than on low-level programming details and application scheduling/execution, allowing users to define, record, and modify problems, and visualize and optimize simulation results," now seems eminently achievable.

10. Acknowledgement

The authors wish to thank Alex Verstak for the telnet server in WBCSim, and Lauren Keller for TEX assistance. This work was supported in part by a Virginia Polytechnic Institute & State University 1997 ASPIRES grant, USDA Grant 97-35504-4697, and DOE contract DE-AC04-95AL97273-91830.

References

[1] R. Akers, E. Kant, C.J. Randall, S. Steinberg, R.L. Young, SciNapse: a problem-solving environment for partial differential equations, IEEE Computational Science and Engineering 4 (1997) 32–42.

[2] A.L. Ames, D.R. Nadeau, J.L. Moreland, VRML 2.0 Sourcebook, 2nd ed (John Wiley & Sons, Inc., New York, 1996).

[3] S. Balay, B. Gropp, L. Curfman McInnes, B. Smith, A microkernel design for component-based parallel numerical software systems, Technical Report ANL/MCS-P727-0998, Argonne National Laboratory, 1998.

[4] L. Beca, G. Cheng, G.C. Fox, T. Jurga, K. Olszewski, M. Podgorny, P. Sokolowski, K. Walczak, Java enabling collaborative education, health care and computing, Concurrency Practice and Experience 9 (1997) 521–534.

[5] R.F. Boisvert, J.R. Rice, Solving elliptic problems using ELLPACK (Springer-Verlag, New York, 1985).

[6] A.J. Bolton, P.E. Humphrey, The hot-pressing of dry-formed wood-based composites. Part I. A review of the literature, identifying the primary physical process and the nature of their interaction, Holzforschung 42 (1988) 403–406.

[7] A.J. Bolton, P.E. Humphrey, P.K. Kavvouras, The hot-pressing of dry-formed wood-based composites. Part III. Predicted vapour pressure and temperature variation with time, compared with experimental data for laboratory boards, Holzforschung 43 (1989) 265–274.

[8] A.J. Bolton, P.E. Humphrey, P.K. Kavvouras, The hot-pressing of dry-formed wood-based composites. Part IV. Predicted variation of mattress moisture content with time, Holzforschung 43 (1989) 345–349.

[9] A.J. Bolton, P.E. Humphrey, P.K. Kavvouras, The hot-pressing of dry-formed wood-based composites. Part VI. The importance of stresses in the pressed mattress and their relevance to the minimization of pressing time, and the variability of board properties, Holzforschung 43 (1989) 406–410.

[10] M.E. Bowen, Heat Transfer in particleboard during hot pressing, Ph.D. Dissertation, Colorado State University, Fort Collins, CO, 1970.

[11] R. Bramley, D. Gannon, T. Stuckey, J. Villacis, E. Akman, J. Balasubramanian, F. Breg, S. Diwan, M. Govindaraju, The linear system analyzer, Technical Report TR-511, Computer Science Department, Indiana University, Bloomington, IN, 1998.

[12] T. Burnett, C. Chaput, H. Arrighi, J. Norris, D.J. Suson, Simulating the Glast satellite with Gismo, IEEE Computing in Science and Engineering 2 (2000) 9–18.

[13] L.M.H. Carvalho, C.A.V. Costa, Modeling and simulation of the hot-pressing process in the production of medium density fiberboard (MDF), Chem. Eng. Comm. 170 (1998) 1–21.

[14] H. Casanova, J. Dongarra, NetSolve: a network server for solving computational science problems, International Journal of Supercomputer Applications and High Performance Computing 11 (1997) 212–223.

[15] A.C. Catlin, C. Chui, C. Crabill, E.N. Houstis, S. Markus, J.R. Rice, S. Weerawarana, PDELab: an object-oriented framework for building problem solving environments for PDE based applications, in A. Vermeulen, ed., 2nd Object-Oriented Numerics Conference, (Rogue Wave Software, Corvallis, OR, 1994) 79–92.

[16] A. Chabert, E. Grossman, L. Jackson, S. Pietrovicz, NCSA Habanero: synchronous collaborative framework and environment, in Conference Supplement of the Fifth European Conference on Computer-Supported Cooperative Work (ECSCW'97), (nc Lancaster L, UK, 1997) 7–8.

[17] J.X. Chen, X. Fu, Integrating physics-based computing and visualization: modeling dust behavior, IEEE Computing in Science and Engineering 1 (1999) 12–16.

[18] J. Czyzyk, J.H. Owen, S.J. Wright, NEOS: optimization on the Internet, Technical Report OTC-97/04, Argonne National Laboratory, 1997.

[19] D. Dabdub, R. Manohar, Performance and portability of an air quality model, Parallel Computing 23 (1997) 2187–2200.

[20] C. Dai, P.R. Steiner, Spatial structure of wood composites in relation to processing and performance characteristics. Part III. Modeling the formation of multi-layered random flake mats, Wood Science and Technology 28 (1994) 229–239.

[21] T.L. Disz, R. Evard, M.W. Henderson, W. Nickless, R. Olson, M.E. Papka, R. Stevens, Designing the future of collaborative science: Argonne's Futures Laboratory, IEEE Parallel & Distributed Technology 3 (1995) 14–21.

[22] W.R. Dyksen, C.J. Ribbens, Interactive ELLPACK: an interactive problem solving environment for elliptic partial differential equations, ACM Transactions on Mathematical Software 13 (1987) 113–132.

[23] B. Ford, R.M.J. Iles, The What and Why of Problem Solving Environments for Scientific Computing, in B. Ford and F. Chatelin, ed., Problem Solving Environments for Scientific Computing, (Elsevier Science Publishers, North-Holland, 1985) 3–18.

[24] E. Gallopoulos, E. Houstis, J.R. Rice, Computer as thinker/doer: Problem-solving environments for computational science, IEEE Computational Science & Engineering 1 (1994) 11–23.

[25] D. Gannon, R. Bramley, T. Stuckey, J. Villacis, J. Balasubramanian, E. Akman, F. Breg, S. Diwan, M. Govindaraju, Component architectures for distributed scientific problem solving, IEEE Computational Science and Engineering 5 (1998) 50–63.

[26] A. Goel, C.A. Baker, C.A. Shaffer, B. Grossman, W.H. Mason, L.T. Watson, R.T. Haftka, VizCraft: a problem-solving environment for aircraft configuration design, IEEE Computing in Science and Engineering 3 (2001) 56–66.

[27] A. Goel, C. Phanouriou, F.A. Kamke, C.J. Ribbens, C.A. Shaffer, L.T. Watson, WBCSim: a prototype problem solving environment for wood-based composites simulations, Engineering with Computers 15 (1997) 198–210.

[28] D.S. Goldin, S.L. Venneri, A.K. Noor, Beyond incremental change, IEEE Computer 31 (1998) 31–39.

[29] T.E.G. Harless, F.G. Wagner, P.H. Short, R.D. Seale, P.H. Mitchell, D.S. Ladd, A model to predict the density profile of particleboard, Wood and Fiber Science 19 (1987) 81–92.

[30] C.R. Haselein, Numerical simulation of pressing wood-fiber composites, Ph.D. Dissertation, Forest Products Dept., Oregon State University, Corvallis, OR, 1998.

[31] E. Houstis, E. Gallopoulos, R. Bramley, J.R. Rice, Problem-solving environments for computational science, IEEE Computational Science and Engineering 4 (1997) 18–21.

[32] E.N. Houstis, J.R. Rice, S. Weerawarana, A.C. Catlin, P. Papachiou, K.Y. Wang, M. Gaitatzes, PELLPACK: a problem solving environment for PDE-based applications on multicomputer platforms, ACM Transactions on Mathematical Software 24 (1998) 30–73.

[33] P.E. Humphrey, Physical aspects of wood particleboard manufacture, Ph.D. Dissertation, University of Wales, Bangor, UK, 1982.

[34] P.E. Humphrey, The hot-pressing of dry-formed wood-based composites. Part II. A simulation model for heat and moisture transfer, and typical results, Holzforschung 43 (1989) 199–206.

[35] P.L. Isenhour, J. Begole, W.S. Heagy, C.A. Shaffer, Sieve: A Java-based collaborative visualization environment, in Late Breaking Hot Topics Proceedings, IEEE Visualization '97, (Phoenix, AZ, 1997) 13–16.

[36] A. Joshi, S. Weerawarana, N. Ramakrishnan, E.N. Houstis, J.R. Rice, Neuro-fuzzy support for problem-solving environments: a step toward automated solution of PDEs, Special Joint Issue of IEEE Computer and IEEE Computational Science and Engineering 3 (1996) 44–56.

[37] F.A. Kamke, J.B. Wilson, Computer simulation of a rotary dryer: retention time, American Institute of Chemical Engineers J. 32 (1985) 263–268.

[38] F.A. Kamke, J.B. Wilson, Computer simulation of a rotary dryer: heat and mass transfer, American Institute of Chemical Engineers J. 32 (1985) 269–275.

[39] F. Kayihan, A. Johnson, Heat and moisture movement in wood composite materials during the pressing operation - a simplified model, in R.W. Lewis, K. Morgan, B.A. Schrefler ed., Numerical Methods in Heat Transfer, Vol. II., (John Wiley & Sons Ltd., New York, 1983) 511–531.

[40] R.G. Knox, V.L. Kalb, E.R. Levine, D.J. Kendig, A problem-solving workbench for interactive simulation of ecosystems, IEEE Computational Science and Engineering 4 (1997) 52–60.

[41] M.E. Lang, M.P. Wolcott, Modeling the consolidation of wood-strand mat, Mechanics of Cellulosic Materials ASME AMD 209/MD 60 (1995) 153–176.

[42] M.E. Lang, M.P. Wolcott, A model for viscoelastic consolidation of wood-strand mats. Part I. Structural characterization of the mat via Monte Carlo Simulation, Wood and Fiber Science 28 (1996) 100–109.

[43] M.E. Lang, M.P. Wolcott, A model for viscoelastic consolidation of wood-strand mats. Part II. Static stress-strain behavior of the mat, Wood and Fiber Science 28 (1996) 369-379.

[44] K. Long; Van B. Straalen, PDESolve: an object-oriented PDE analysis environment, in Proceedings of the SIAM Workshop on Object Oriented Methods for Inter-operable Scientific and Engineering Computing, (Philadelphia, PA, 1998).

[45] C. Lu, Organization of wood elements in partially oriented flakeboard mats, Ph.D. Dissertation. Dept. of Forestry, University of British Columbia, Vancouver, BC, 1999.

[46] S. Markus, S. Weerawarana, E.N. Houstis, J.R. Rice, Scientific computing via the Web: the Net Pellpack PSE server, IEEE Computational Science and Engineering 4 (1997) 43–51.

[47] J. Marti, Viewing IGES files through VRML, in Proceedings of Visualization '97, (IEEE Computer Society Press, Los Alamitos, CA, 1997) 471–472.

[48] S.M. Mniszewski, P.H. Beckman, P.K. Fasel, W.F. Humphrey, Efficient coupling of parallel applications using PAWS, in Proceedings of the Seventh IEEE International Symposium on High Performance Distributed Computing (HPDC7), (Chicago, IL, 1998).

[49] S.G. Parker, D.M. Weinstein, C.R. Johnson, The SCIRun computational steering software system, in E. Arge, A. M. Bruaset, H. P. Langtangen eds., Modern Software Tools in Scientific Computing, (Birkhauser Press, 1997) 1–40.

[50] C. Phanouriou, M. Abrams, Transforming command-line driven systems to Web applications, in Sixth International World Wide Web Conference, (Santa Clara, CA, 1997) 1–3.

[51] W.C. Regli, Internet-enabled computer-aided design, IEEE Internet Computing 1 (1997) 39–50.

[52] J. Resnik, F.A. Kamke, Modeling the cure of adhesive-wood bonds using high frequency energy, Final Report, U.S.-Slovene Joint Board on Scientific and Technological Cooperation, Project 95-AES10, University of Ljubljana, Ljubljana, Slovenia, 1998.

[53] W.E. Schiesser, Computational Mathematics in Engineering and Applied Science: ODEs, DAEs, and PDEs (CRC Press, Boca Raton, 1994).

[54] C.A. Shaffer, L.T. Watson, D.G. Kafura, and N. Ramakrishnan, Features of problem solving environments for computational science, in A. Tentner ed., Proc. High Performance Computing Symp. 2000, (Soc. for Computer Simulation Interna, San Diego, CA, 2000) 242–247.

[55] A. Shah, Symphony: A Java-based composition and manipulation framework for distributed legacy resources, M.S. Thesis, Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA, 1998.

[56] H. Thöemen, Modeling the physical processes in natural fiber composites during batch and continuous pressing, Ph.D. dissertation. Forest Products Dept., Oregon State University, Corvallis, OR, 2000.

[57] Y. Umetani, M. Tsuji, K. Iwasawa, H. Hirayama, DEQSOL: A numerical simulation language for vector/parallel processors, in B. Ford, F. Chatelin ed., Problem Solving Environments for Scientific Computing, (Elsevier Science Publishers, North-Holland, 1987) 147–162.

[58] Vanderplaats Research & Development, Inc., DOT Users Manual, Version 4.20 (Colorado Springs, CO, 1985).

[59] S. Weerawarana, E.N. Houstis, J.R. Rice, A. Joshi, PYTHIA: a knowledge based system to select scientific algorithms, ACM Transactions on Mathematical Software 22 (1996) 447–468.

[60] C.F. Weggel, Versatile 2-D analysis, IEEE Spectrum 34 (1997) 92–93.

[61] S. Wolfram, The Mathematica Book, 3rd ed. (Wolfram Media/Cambridge University Press, 1996).

[62] B.G. Zombori, Modeling the transient effects during the hot-pressing of wood-based composites, Ph.D. Dissertation, Virginia Polytechnic Institute and State University, Blacksburg, VA, 2001.

[63] B.G. Zombori, F.A. Kamke, L.T. Watson, Modeling the transient effects during the hot-pressing of wood-based composites—Part I: simulation of the mat formation process, Wood Fiber Sci. to appear.