MatrixPro - User's Manual

Version 1.2

Ville Karavirta et al.

December 19, 2005

© Helsinki University of Technology, 2005

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with Invariant Section "Introduction," no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Contents

1 Introduction

MatrixPro is a tool for visualizing and animating data structures and algorithms. It can be used by teachers for demonstrating algorithms and by students for experimenting with algorithms and solving exercises. The most widely taught tree and graph algorithms are built-in, so that creating a visualization can be done interactively without programming. Visualizations can be saved and reloaded, and can be exported in SVG or LaTeX formats.

A novel feature of MatrixPro is the ability to simulate algorithms by direct interaction.

The website of MatrixPro is `http://www.cs.hut.fi/Research/MatrixPro/`.

2 Installing MatrixPro

This section describes how to download and run MatrixPro. Advanced material on building and installing the software can be found in Appendix C.

MatrixPro requires Version 1.4 of the Java SDK or JRE be installed.

Download the file `matrixpro-full.jar` from the MatrixPro website. On many platforms you can create an icon for this file and then run the program simply by double-clicking on the icon. Alternatively, you can start the tool by typing the following at a command prompt:

```
java -jar matrixpro-full.jar
```

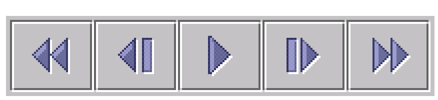
For this to work, the Java JRE or SDK must be in the search path.

3 Quick start

This chapter gives you a brief explanation of how to use MatrixPro. Only a few of the features will be presented, but it should be enough to get you started without reading more than a few pages.

3.1 Interacting with MatrixPro

Figure 1 shows the main window of MatrixPro. Along the top is the menu bar (Section 5). Below it on the left is the *animator* containing VCR-like controls for the visualization:



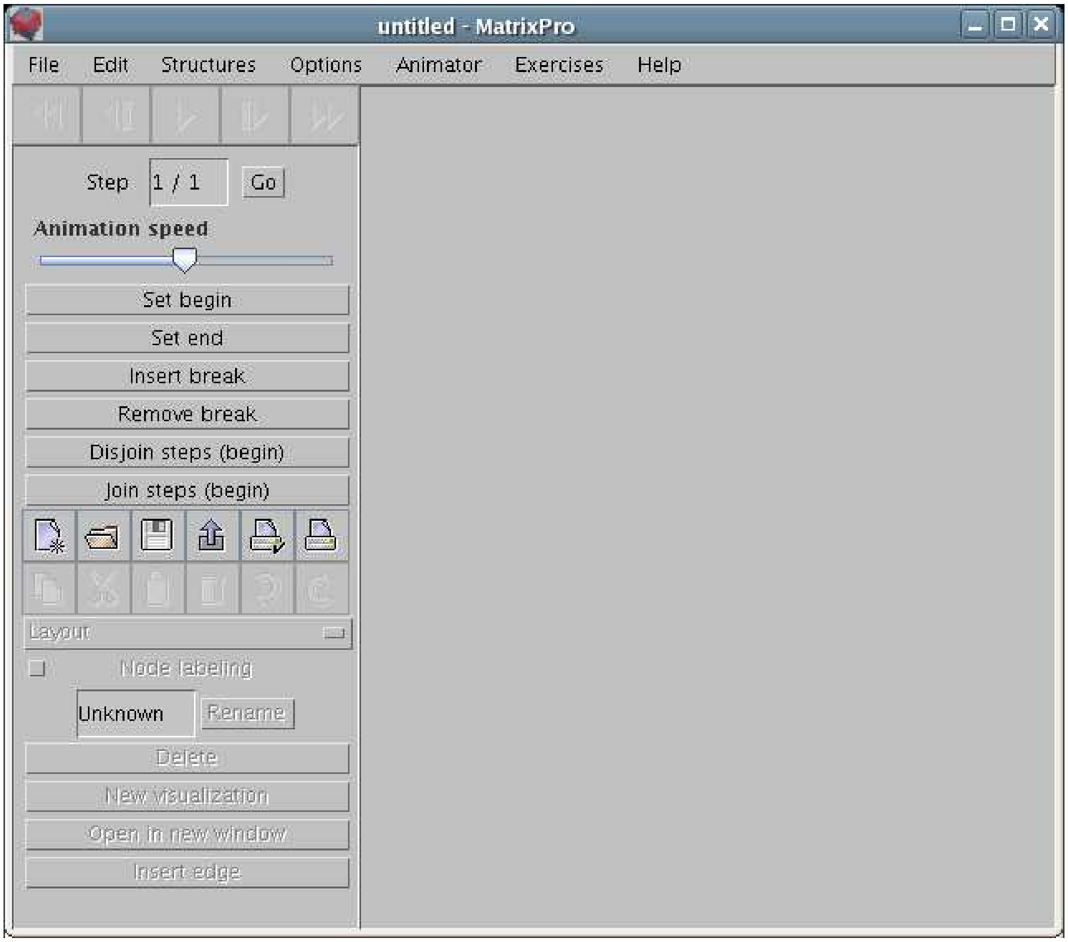


Figure 1: The MatrixPro main window.

and below the animator is the *toolbar* with additional controls (Section 9). On the right is the structure panel where all the visualizations appear; these are explained in Sections 7 and 8. Not shown are context-sensitive popup menus (Section 6).

The functions of the animator buttons (from left to right) are:

- Begin will undo all operations in the history.
- Backward will undo one operation.
- Play will run the animation continuously from the current position to the end of the scenario. The button will change to a Stop button that stops the animation.
- Forward will redo one operation.
- End will redo all operations in the current scenario.

3.2 The structure panel

The Structures menu contains four submenus for creating various data structures:

Fundamental data types (FDT) These are "raw" data types that have no predefined semantics, for example, binary trees or graphs. The elements of FDTs are normally *keys*, though they can also be other FDTs.

Conceptual data types (CDT) These are data structures that have constrained values and operations defined upon them. For example, binary search trees are constrained so that values in the left subtree of a node are less than values in right subtree, and the operations of insert, delete and search are defined upon them.

Utilities These are primarily structures containing keys for use in FDTs and CDTs.

Datatype with data These are pairs of array of keys and commonly used CDT structures.

Multiple data structures can be created in the structure panel; the panel can be scrolled or resized as needed.

Let us work through an example. Select Structures/Utilities/Array of Random Keys. The array will appear in the upper part of the structure panel. Next, select Structures/Conceptual data types/Binary Search Tree. An empty BST will appear below the array in the structure panel. Drag the keys one-by-one from the array of keys and drop each one *onto the title bar* of the BST. While an object is being dragged, a frame appears around it. The keys are inserted into the correct positions, according to the definition of a BST (Figure 2). Instead of dragging keys one by one, you can drag the entire array of keys (from its title bar) and drop it onto the title bar of the tree. All the keys in the array will be inserted one by one into the binary search tree.

Now you can step through the animation by clicking the Backward and Forward icons of the animator; clicking Begin will start the animation from the beginning and clicking Play will run it without stopping.

Move the mouse over a node until **the node label is highlighted** (e.g. turns red). Press the right mouse button and select Delete from pop-up menu. The node will be deleted from the tree according to the definition of BSTs. Alternatively, move the mouse over a node until **the node background is highlighted** (e.g. turns blue) and select Delete; the subtree rooted at that node is deleted.

In the next section, we will desribe in detail how to interact with data structures.

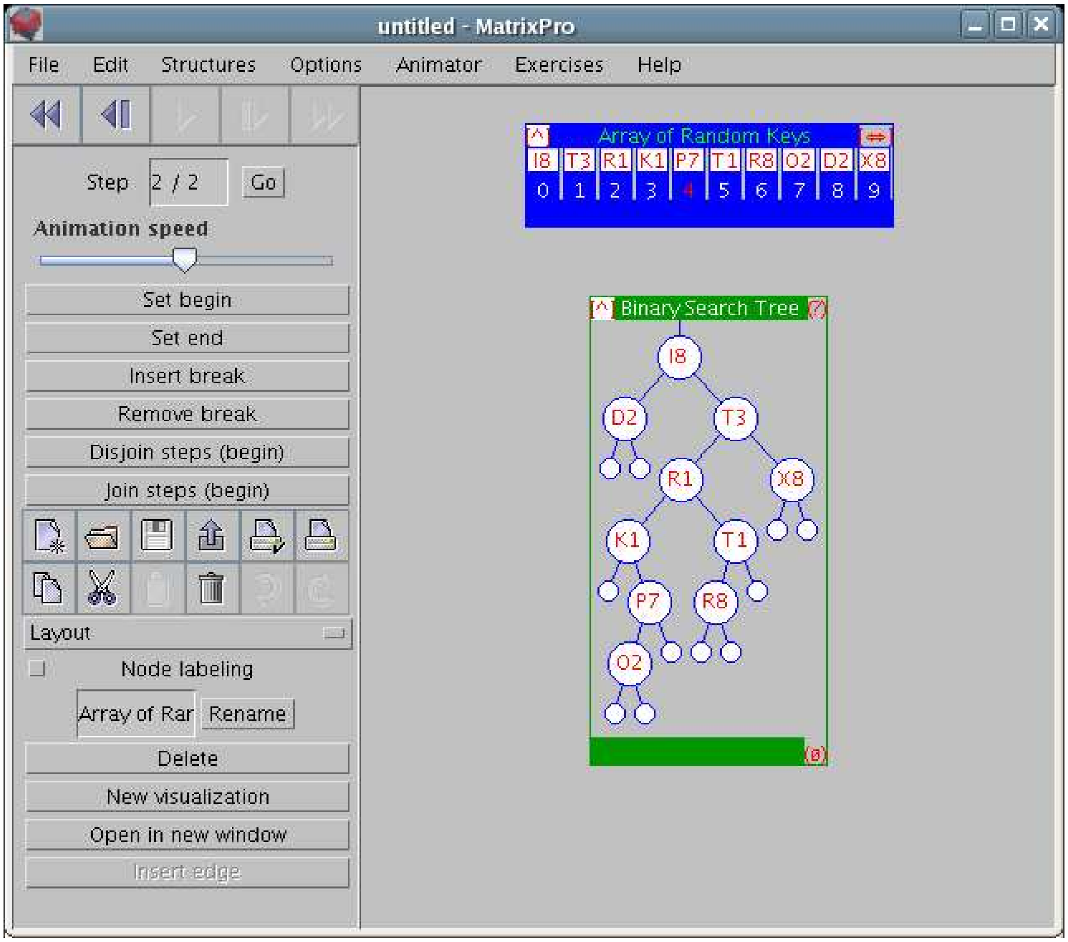


Figure 2: Inserting keys into a Binary Search Tree.

4 Interaction

4.1 Hotspots

Data structures with a visible title have a hotspot in the left corner of the title bar, denoted by a square containing a red "[^]". The popup menu can be opened by left-clicking on this hotspot (Section 6).

Arrays contain another hotspot (two-headed arrow) in the right corner of the title bar. The number of elements displayed in the array can be changed by dragging this hotspot left or right.

There are also two additional hotspots. In the top right corner of the CDT structures there is the search hotspot "(?)" which can be used to call the search method of the CDT structure by dropping the searched key on the hotspot. In the bottom right corner of structures (except arrays) there is the null hotspot "(0)" which can be used to modify references point to empty nodes. To use it, drag and drop a reference to the hotspot and update references from Options menu.

See Figure 2 for examples of the hotspots.

4.2 Inserting keys into a data structure

In the example of the binary search tree, we showed how keys can be inserted into a data structure by dropping them onto the title bar of the data structure. The insertion routine of the corresponding data structure will then be invoked and the visualization will be updated. It is also possible to insert a key in a *specific* node of a data structure, by dragging and droping the new key into the desired node.

For example, create an array of keys and an AVL tree and insert `M` into the tree. You can now experiment with dragging and dropping another key (say `P`) either onto the title bar or into the root node or one of the leaves (Fig. 3). In each case the results will be different.

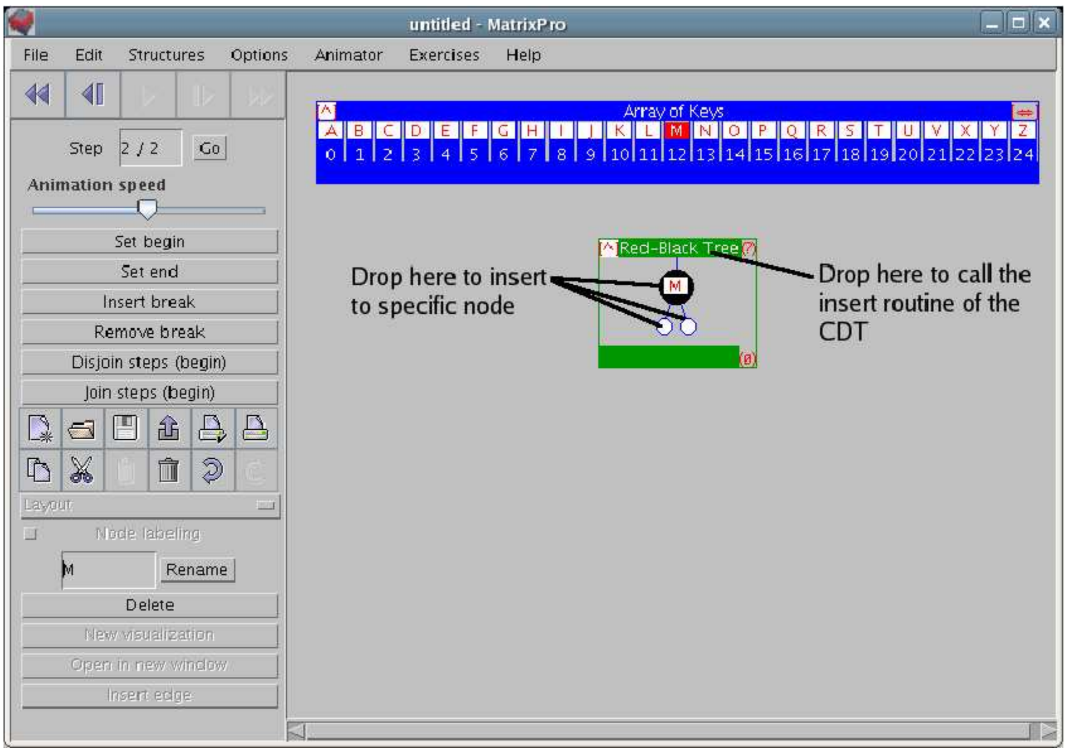


Figure 3: Inserting keys into a specific node or by invoking the insert routine

MatrixPro supports nested data structures of arbitrary complexity. You can store an array inside a node of a graph or a tree in as an array element; for example, the B-tree implementation (2-3-4 Tree) uses arrays nested within trees to hold the keys. Some fundamental data types, such as arrays or binary trees, have no semantics for inserting keys "into the structure." For such structures, keys must be inserted at a specific position; in case of inserting a set of keys, the entire set will be inserted as a unit.

4.3 Deleting keys and nodes

Elements can be deleted from a data structures by using the Delete command in the popup menu. What the Delete command actually does depends on the data structure or structure component upon which it was invoked. Deleting a data structure will remove the whole structure, including its visualization from the current frame; it is not possible to undelete it by going backward in the animation. Deleting a tree node removes the subtree rooted at the deleted node, while deleting a graph node causes that node and all references to or from it to be deleted. Deleting a node in a linked list will remove that node from the list. Some components of structures, such as array indices, cannot be deleted.

The effects of deleting a node from a CDT depend on the CDT in question. The two ways to delete an item described above both act as if the deletion were performed on the FDT upon which the CDT is based. A different way to delete an item from a CDT is to hold the Shift key down while dragging the item away from the CDT and dropping it somewhere else (such as an empty part of the animation window). This deletes the item from the CDT, and if the item is dropped on another structure, it is inserted as usual. Note that the CDT's delete routine is used to perform the delete in with Shift held down; for example, Shift-dragging an item from a stack will always cause the topmost item on the stack to be deleted regardless of which item was dragged.

Elements can also be deleted by dragging and dropping them in a trash can, which can be created by selecting Structures/Utilities/Trash.

4.4 Other operations

To copy a subtree, drag the root node of the subtree to the desired position. If you copy a subtree to a different position in the same tree, the first subtree (that will be found in a DFS) will be displayed, and the other copies will be shown minimized. The copied tree points to the original tree, so changes in either of the visualizations affect both the original and the copy.

References in graphs and trees can be moved to point to another node by dragging and dropping them on the new target node. Tree nodes that have no references are removed. References must be explicitly updated after a drag and drop operation by selecting Options/Update References.

In MatrixPro operations are grouped into animation steps that can contain other, smaller, steps. The smallest possible steps (atomic steps) may not have any visible effect on the visualizations. Normally, the animation control buttons work on non-atomic steps, but they work on atomic steps if Shift is pressed when selecting Backward or Forward.

5 Menu commands

5.1 File menu

New Window (Ctrl+N): Opens a new animation window.

Open (Ctrl+O): Opens a new data structure. Java class files, saved Matrix animations and text files containing the string representation of a data structure can be opened. MatrixPro knows how to visualize saved animations and strings automatically, but Java classes must implement the visualization interfaces correctly. Appendix A describes the text file formats as well as an extended text file format. Animations are saved as serialized Java objects, so there will be problems if an object's class has changed after it was saved. Customizations to the visualization are not saved with the serialized animations.

Open recent: Opens one of the recently opened or saved files.

Save As...: Saves the data structures either as Serialization or ASCII. The type can be set from the Files of Type drop-down list.

Close (Ctrl+W): Closes the current window; if there is only one window, this is the same as Exit.

Clear: Clears the current active structure panel.

Export...: Exports the current animation or the view of one step of the animation in one of the formats selected from Files of Type. These are described in Appendix B.

Page Setup...: Opens the page setup dialog for printers.

Print...: Prints the current window.

Print animation...: Prints the entire animation, each step on its own page.

Exit: Exits the program.

5.2 Edit menu

Font: Changes the font used by visualizations.

Font size: Changes the font size used by visualizations.

Show: Selects the visible user interface components: either the toolbar *including* the animator or only the animator.

Copy (Ctrl-C): Copies the selected structure to the clipboard.

Cut (Ctrl-X): Copies the selected structure to the clipboard and then deletes the original structure.

Paste (Ctrl-V): Pastes the structure from the clipboard. This calls the insert routine of the selected structure, so the behaviour depends on the implementation of that structure. Note that the pasted structure and the original structure are both visualizations of the same structure, so modifications done to one change the other.

Paste as duplicate: Pastes the structure from the clipboard as a new visual structure; only whole structures can be pasted this way, not keys or nodes. A structure can be selected by clicking on its title bar. Changes in the new visualization affects the original and vice versa.

Delete: Deletes the selected structure. Again, the effect depends on the underlying structures.

Undo: Undoes the last user interface operation. Not all operations can be undone and visualization customizations such as rotations are lost.

Redo: Redoes the last undone user interface operation. Visualization customizations are lost.

5.3 Structures menu

The Structures menu is used to instantiate the data structures built into MatrixPro. For more information about the different structures, see Section 7.

5.4 Options menu

The options menu contains special commands for simulation purposes

Update references: Updates the references between elements and repaints the visualized data structures. You can change several references at a time by moving them to point to the desired target. The underlying structure (and thus, the visualization) is not updated until the update references operation is called, as described in Section 4.4.

Swap: Changes the semantics of drag and drop. The default semantics is Insert which moves an element from the source location to the destination but does not change the original source structure. The semantics of Swap cause the source and target elements to be swapped. Swap is intended for use on keys in arrays and FDTs.

Preferences..: This opens a dialog with two tabbed panes.

- **Toolbar** Customizes the toolbar (Figure 4). A component can be hidden by selecting it from the list of visible components and clicking Hide, and a hidden component can be made visible by selecting it and clicking Show. Components can be moved up or down in the toolbar by selecting them and clicking Move up or Move down. Add opens a dialog for adding a component; you have to select its class file.

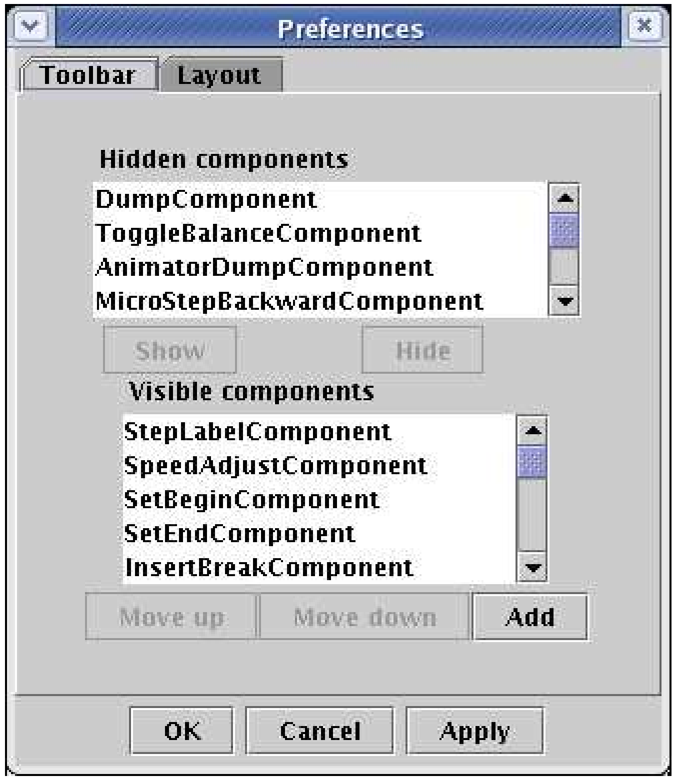


Figure 4: Customizing the toolbar.

Layout Customizes the positions and insets of the visualizations.

The values that can be set for each visualization are:

- *gridx* — x position of the visualization (left=0)
- *gridy* — y position of the visualization (top=0)
- *gridheight* — number of rows for the visualization
- *gridwidth* — number of columns for the visualization
- *ipadx* — space to add to the components' width
- *ipady* — space to add to the components' height
- *weightx* — how to distribute extra horizontal space
- *weighty* — how to distribute extra vertical space
- *top inset* — amout of external padding above the visualization
- *bottom inset* — amout of external padding below the visualization
- *left inset* — amout of external padding left of the visualization
- *right inset* — amout of external padding right of the visualization

5.5 Animator menu

The Animator menu contains commands to control and modify the animator. In addition to menu equivalents of the buttons on the animator, there are two additional selections:

Set beginning here: Sets the current state to be the beginning of the animation. The previous states can no longer be reached.

Set end here: Sets the current state to be the end of the animation. The following states can no longer be reached.

5.6 Exercises menu

This menu contains a large number of algorithms and data structures, categorized in submenus. Selecting one of them causes a new window to be displayed containing a random instance of an exercise for that topic. On the left a panel appears with instructions on the exercise, while on the right a structure panel is initialized.

For example, select Exercises/Basic algorithms/Binary search. The structure panel is initialized with an array of keys and an empty linked list; the instruction panel contains the value to search for, and the instructions are to drag the visited keys from the array to the linked list. When you have finished select actions from the Exercise menu:

Reset: Resets the exercise to a new instance.

Model answer: Displays an animation of the correct answer to the exercise.

Grade my solution: Displays a frame with the number of correct steps in the solution.

5.7 Help menu

Help: Displays information about where to find MatrixPro User's manual and some tutorials.

About: Displays the copyright notice and the version.

6 Popup menu

Popup menu can be opened by right-clicking on a component. That component can be a structure, a node or a key. The available operations in the popup menu depend on the component which was right-clicked.

New visualization: Creates a new visualization of the data structure in the current animation window. Changes in the new visualization affects the original and vice versa.

Delete: Invokes the delete method for this object. By default this removes the selected structure or component from the underlying data structure.

Change layout: Changes the layout for the data structure.

Visualization: This submenu (described in Section 6.1) contains commands that directly modify how the data structure is visualized.

Filters: This submenu (described in Section 6.2) contains commands that—depending on the data structure—filter out the structure's details or select only a part of it to be represented.

Rename: Renames a data structure. This only affects keys, data structures with a header, or labeled nodes. This command is also used to modify the value of a key.

Rename all keys (arrays only): Opens a dialogue to rename all keys of the array. The keys must be separated by spaces.

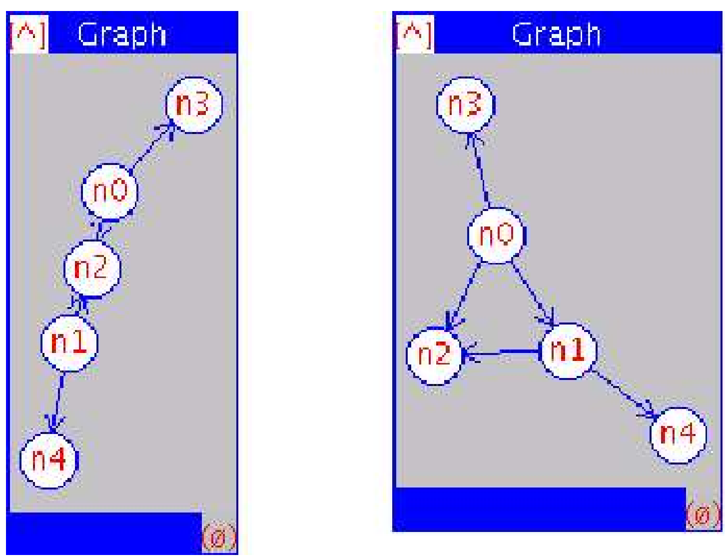
Labeled (nodes only): Chooses whether labels next to nodes are displayed.

InsertEdge (graph vertices only): Inserts an edge between two vertices after the destination vertex has been clicked.

Refresh: Refreshes the visualization. It will also create new keys for an array of random keys.

Call: Calls a user-defined method (without parameters) if they have been defined for this object.

Change Edge Length: (For graphs using either the Kamada-Kawai or the Fruchterman-Reingold layout.) Opens a popup window where you can type a new edge length used in the algorithm. Changing its value can have dramatic effect on the layout:



6.1 Visualization Menu

The visualization menu contains commands that modify the way the structure is visualized:

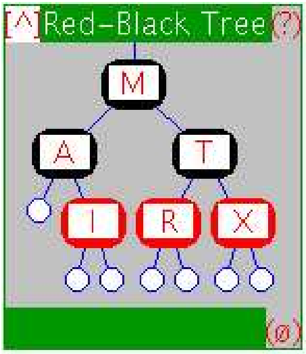
Minimized: Minimizes or maximizes a visualization.

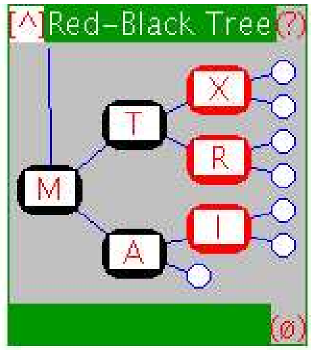
Alive: Enables and disables a visualization's response to simulation operations such as dragging and dropping.

Enable: Enables and disables direct access to the subcomponents of a visualization.

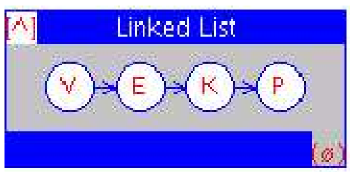
Titled: Displays the title bar in a data structure.

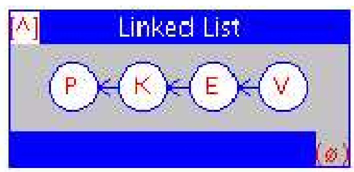
Rotated: Rotates the visualization:



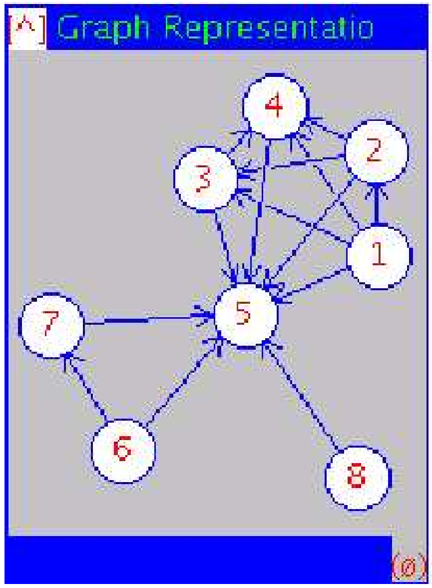


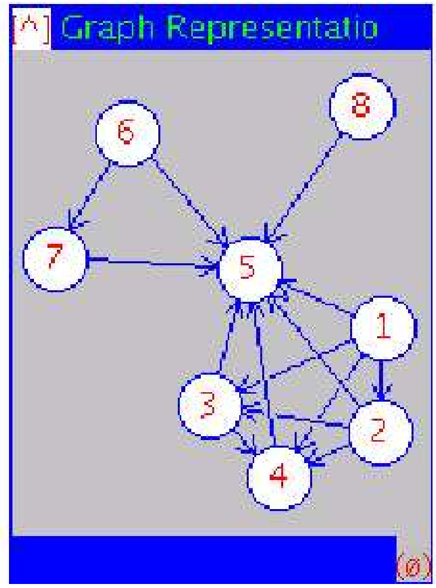
FlipX: Flips the X coordinates of the visualization:





FlipY: Flips the Y coordinates of the visualization:





Indexed (arrays only): Displays or hides the indices of an array.

6.2 Filters menu

The filters menu depends on the data structure.

Directed (trees and graphs only): Edges are directed.

EmptyLeaves (trees only): Show empty leaves.

DFSvalidate (graphs only): Validate the graph in DFS order; otherwise, validate in BFS order.

BackEdges: Show back edges for graphs.

ForwardEdges: Show forward edges for graphs.

CrossEdges: Show cross edges for graphs.

Increment (arrays only): Increment the size of an array.

Decrement (arrays only): Decrement the size of an array.

Double (arrays only): Double the size of an array.

Halve (arrays only): Halve the size of an array.

RaiseIndex (arrays only): Shift array indexes right by one.

LowerIndex (arrays only): Shift array indexes left by one.

7 Structures

7.1 Fundamental data types

Fundamental data types (FDT) include the basic structures like binary trees, arrays, linked lists and graphs:

Array Inserting keys in an array can be done by dropping them either on the key (initially empty) or the index.
Default layout: array
Possible layouts: array

Linked List Inserting keys in the list can be done by dropping them onto the structure. This always inserts the keys as the first element of the list. To insert a key in the middle of the list, drop the new key onto the node *after* which you want the new key to be inserted.
Default layout: list
Possible layouts: list

Dynamic Binary Tree A dynamic binary tree starts with a single node that is the root. Dropping a key into a leaf, creates a node with two leaves, enabling construction of arbitrary binary trees.
Default layout: layered tree
Possible layouts: array, layered tree, leaf tree, layered graph vertex

Static Binary Tree (8) This is a binary tree with exactly eight nodes. This is an array representation of a tree.
Default layout: layered tree
Possible layouts: array, layered tree, leaf tree

Common Tree This can be used to construct arbitrary trees. A new node is inserted as a child of an existing node by dropping a key onto the existing node. Be sure to drop the key on the *node* (the background turns blue), not onto the *key* (the key turns red).
Default layout: layered tree
Possible layouts: layered tree, leaf tree, layered graph vertex

Directed Graph Nodes can be inserted by dropping them onto the graph. Inserting edges can be done in three ways:

- Select Insert edge from the source node's popup menu and then click on the target node.
- Select the source node, click Insert node on the toolbar and then click on the target node.
- Select the source node with Shift key held down and then click on the target node.

Default layout: layered graph
Possible layouts: layered graph, Kamada-Kawai graph, Fruchterman-Reingold graph, dummy graph, array

Undirected Graph Nodes and edges are inserted in the same way as for directed graphs.

Default layout: layered graph

Possible layouts: layered graph, Kamada-Kawai graph, Fruchterman-Reingold graph, dummy graph, array

7.2 Conceptual data types

Conceptual data types (CDT) are more complex structures that have a predefined set of operations whose implementation depends on the CDT. Inserting keys should be always done by dropping the keys on the title bar of the CDT. Keys can be deleted by selecting either Delete from the popup menu of a key, or by selecting a key and then deleting it using toolbar button, or by holding the Shift key while dropping them outside the structure. For more information on deleting parts of the structure see Section 4.3.

Binary Search Tree

Default layout: layered tree

Possible layouts: array, layered tree, leaf tree, layered graph vertex

2-3-4 Tree

Default layout: layered tree

Possible layouts: layered tree, leaf tree

Red-Black Tree

Default layout: layered tree

Possible layouts: array, layered tree, leaf tree, layered graph vertex

Digital Search Tree

Default layout: layered tree

Possible layouts: layered tree, leaf tree

Radix Search Tree

Default layout: layered tree

Possible layouts: array, layered tree, leaf tree, layered graph vertex

Binary Heap

Default layout: layered tree

Possible layouts: array, layered tree, leaf tree

AVL Tree

Default layout: layered tree

Possible layouts: array, layered tree, leaf tree, layered graph vertex

Splay Tree

Default layout: layered tree

Possible layouts: array, layered tree, leaf tree, layered graph vertex

Stack(list)

Default layout: list
Possible layouts: list

Stack(array)

Default layout: array
Possible layouts: array

Queue

Default layout: list
Possible layouts: list

7.3 Utilities

Trash Visual objects that are dragged and dropped onto the Trash are deleted.

Array of Keys An array of all the (capital) letters of the alphabet.

Array of Random Keys An array of random keys of three alphanumeric characters.

7.4 Datatype with data

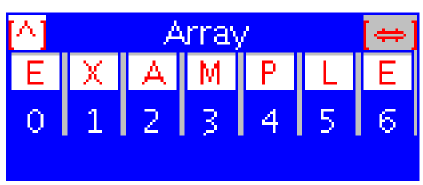
Array of Keys + ... A shortcut to create both a Array of Keys and one of the CDTs.

8 Layouts

There are several different layouts that can be used to visualize the data structures. The layout of a structure can be changed using the Change layout submenu of the popup menu or using the Layout toolbar component.

8.1 Array

The layout Array can be used to represent arrays and trees:



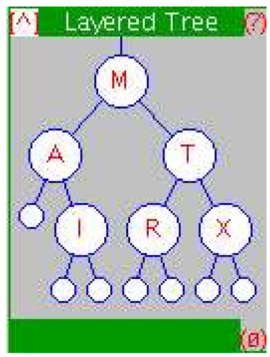
See Section 4.1 for information on the hotspot used to change the size of an array; the size can also be changed using the popup menu's submenu Filters (see Section 6.2).

8.2 List

The layout List can be used to represent linked lists, stacks and queues:

8.3 Trees

Layered Tree The Layered Tree layout:

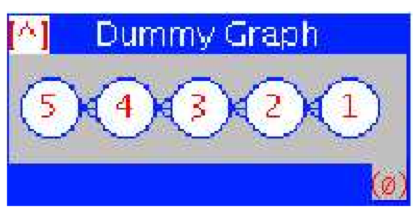


draws a tree using the Layered-Tree-Draw algorithm, extended to support non-binary trees and variable-size nodes.

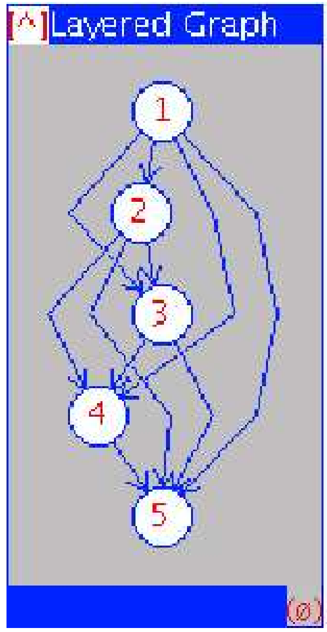
Leaf Tree

8.4 Graphs

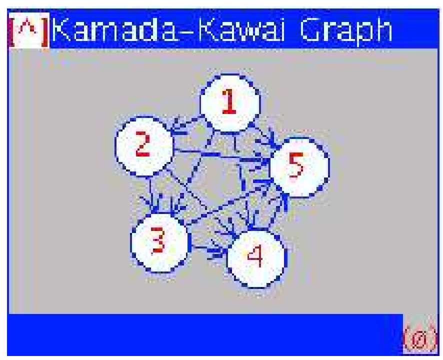
Dummy Graph The dummy graph layout is a simple layout, where all the nodes are positioned in a horizontal line:



Layered Graph The layered graph layout uses a directed acyclic graph algorithm supporting arbitrary graphs and variable-size nodes:

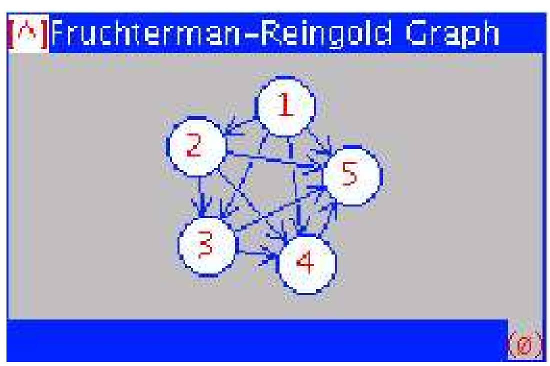


Kamada-Kawai Graph This layout uses the Kamada-Kawai layout algorithm:



The layout can be modified by Change edge length from the graph's popup menu.

Fruchterman-Reingold Graph This layout uses the Fructerman-Reingold layout algorithm:



The layout can be modified by Change edge length from the graph's popup menu.

9 Toolbar

The contents of the toolbar can be customized (Section 5.4) and not all components described here appear by default. The toolbar components are described in a set of tables:

Animation control Table 1.

Animation modification Table 2. All these commands can be undone by selecting undo.

Structure modification Table 3. If no structure is selected, these buttons will be disabled (unless otherwise noted).

Miscellaneous Table 4.

Developer features Table 5.

Some structures can have operations that can be added as buttons to the toolbar. There is a special toolbar component for these components called ContextualPanel. The components for the operations appear in this toolbar component.

Some toolbar components appear or are enabled if they are relevant to the structures appearing in the structure panel; however, the toolbar is not updated until after moving the mouse outside a structure.

Table 1: Animation control

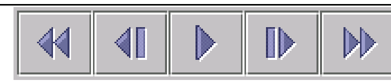
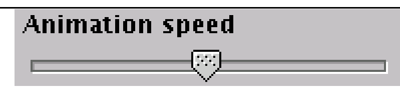
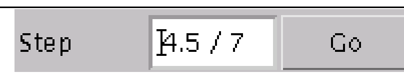
Component	Explanation	Picture
Animator	Begin undoes all the operations. Backward undoes one operation (one enclosed animation step). If the data structure is modified while there are undone operations, these operations can no longer be redone. Holding Shift when pressing Backward undoes one atomic step at a time. Play executes the animation from the current state to the last one. Play changes to Stop for stopping the animation. Forward redoes one operation, or one atomic operation if Shift is held. End redoes all.	
Animation Speed	The speed of the animation can be controlled: right for faster and left for slower.	
Step View	The current step and the number of steps in the animation are shown. Enter a step number and press Enter or click Go to jump to the desired step in the animation.	
Microstep backward	Move one atomic operation backward	
Microstep forward	Move one atomic operation forward	

Table 2: Animation modification

Component	Explanation	Picture
Set begin	Sets the current state to be the beginning of the animation. The previous states can no longer be reached.	
Set End	Sets the current state to be the end of the animation. The following states can no longer be reached.	
Insert break	Add a new break in the animation: the animation will promote the given step to a top level step and make the animator stop at this position when moving Backward or Forward.	
Remove break	Remove a break in the animation so that Backward and Forward no longer stop at this step.	
Join steps	Join several steps in the animation into one step: (1) Go to the step you want to start the join at; (2) Press Join steps; (3) Go to the step you want to end the join at; (4) Press Join steps again. It does not matter which of the selected steps comes first in the animation.	
Disjoin steps	Break up several steps in the animation into distinct steps: (1) Go to the step you want to start the disjoin at; (2) Press Disjoin steps; (3) Go to the step you want to end the disjoin at; (4) Press Disjoin steps again. It does not matter which of the selected steps comes first in the animation. (This command might have no visible effect on the animation.)	

Table 3: Structure modification

Component	Explanation	Picture
New visualization	Creates a new visualization of the selected data structure in the current animation window. Changes in the new visualization affect the original and vice versa.	
Open in new window	Opens a new visualization of the selected data structure in a new animation window. Changes in the new visualization affect the original and vice versa.	
Delete	Invokes the delete method for the selected object. By default this removes the selected structure or component from the underlying data structure.	
Insert edge	Adds edges to graphs. First select the source node, then click Insert edge and finally click on the destination node. If something else than a node of a graph is selected, this button is disabled.	
Rename	Renames a data structure. This affects only keys, data structures with a header and labeled nodes.	
Layout	Changes the layout for the selected data structure; select a layout from the drop-down list.	
Set Edge Length	Enter a new edge length and press Enter or Set edge length. Enabled only for graphs using either the Kamada-Kawai or the Fruchterman-Reingold layout.	
Label Nodes	Automatically label the nodes in a structure with unique numbers beside every node. For an example, see Figure 5. For arrays this feature is available but it does nothing.	

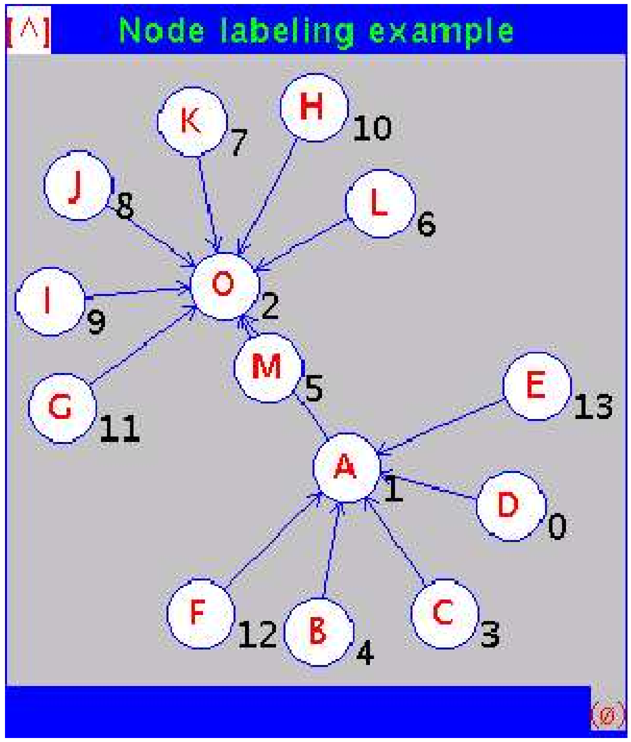


Figure 5: Example of the automatic node labeling.

Table 4: Miscellaneous components

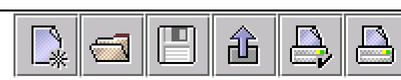
Component	Explanation	Picture
Edit	Quick access to Copy, Cut, Paste, Delete, Undo and Redo operations.	
File	Quick access to New, Open, Save animation, Export, Page Setup and Print.	
Save	Saves the current structures.	

Table 5: Developer's components

Component	Explanation	Picture
Animator dump	Shows debug information for the animator.	
Debug	Switches debug output on or off.	
Dump	Shows debug information for a selected structure or for all objects if no structure is selected.	

A Text File Formats

Section 5.1 described the three representations supported by MatrixPro text files. All examples in this appendix can be found in the `$MATRIX/code/examples/` directory.

edge list The edges of the graph are listed with one node pair per line; each node pair corresponds to an edge in the graph. (Default.)

```
#matrix graph
1 2
1 3
1 4
1 5
2 3
2 4
2 5
3 4
3 5
4 5
```

adjacency-list Each line contains a node and the nodes adjacent to that node; the node and its list of adjacent nodes define edges in the graph.

```
#matrix graph adjacency-list
A:B C D
B:C
C:E
D:E
E:B
F:G
G:H
H:F
```

array Each line contains one key, starting from index 0.

```
#matrix array
A
B
C
D
E
F
```

There is also an extended text file representation which makes it possible for one file to contain an arbitrary number of structures. It also enables keys in the structures to contain special characters like spaces. With this representation, structures can be nested. For example:

```
#matrix structures                //header of the file
test#1                            //name of the first main structure
#matrix graph adjacency-list      //header of the structure
a:test#1_1 c                      //keys in adjacency-list format
test#1_1:e                        //test#1_1 is an inner structure
c:d
e:test#1_2
#EOS                              //end of structure -character
  test#1_1                        //description of the inner structure
  #matrix array
  a
  b
  c
  #EOS
  test#1_2                        //another inner structure
  #matrix graph adjacency-list
  key1:key2 key3 key4
  key2:test#1_2_1
  key4:key6 key7
  #EOS
    test#1_2_1
    #matrix array
    d
    e
    f
    #EOS
test#2                            //another main structure
#matrix array                     //header of the structure
asdf                              //keys in array format
test#2_1
qwerty
#EOS
  test#2_1                        //inner structure
  #matrix array
  aa
  bb
  cc
  dd
```

```
#EOS
```

The names of the main structures must end with a number (for example, `structure#1`) to distinguish them inner structures. Names of the inner structures should be chosen so that it is easy to recognize their parent structures (for example, `structure#1_1`). The description of the structure comes after its name. This can be either in adjacency-list or in array representation. The end of a structure is marked by `#EOS`.

If the keys contain special characters like spaces, _ or #, they must be preceded by a single quote `'` (for example, `inner' structure`, `a'_b`, `'#matrix`). The quote character is removed when the keys are used. If many nodes have the same key in graph, these duplicates can be marked by adding a number at the end (for example, `key`, `key_2`).

Indentation can be freely used, for example, to set off inner structures from outer structures. Comments start with string `//`. Space is also removed from the end of each line.

Some information about the visualization of each structure (representation, rotated, minimized, and name) can be saved into a file. For example:

```
test#1                              //name of the structure in ASCII file
#representation layered graph
#rotated true
#minimized false
#name Binary Tree                   //title of the opened structure
#matrix graph adjacency-list        //header of the structure
```

The structure created from a text file can be visualized as either a tree or a graph. However, if a structure is saved to a text file, its functionality is not saved, only its structure. Additional information can be saved about some structures. Currently, this works only for binary trees: if a BST is saved to a text file and then reopened, it is opended as a binary tree not as a general structure. For example:

```
#matrix structures
structure#1
#name Binary Tree
#matrix autopolymorph matrix.structures.FDT.probe.BinTree   //header
Ns1:5QH BpW
5QH:null s8J
s8J:null GFc
GFc:null null
BpW:Ns1_2 Ddv
Ns1_2:null null
Ddv:null null
#EOS
```

This header contains information about what structure should be loaded, in this case, a binary tree. Structures that are implemented as binary trees (AVL Trees, Red-Black Trees, Splay Trees, etc.) can be saved and loaded as a binary trees.

The first key in the structure description (`Ns1` in this example) becomes the root node of the binary tree. The string `null` is recognized while loading the ASCII file and will be replaced with an empty node in the structure. This is only necessary when the left child of a node is empty and the right child contains some key.

B Exporting

MatrixPro can export visualizations in one of the following formats:

LaTeX Export the current view in LaTeX format. The Complete document checkbox selects whether or not to export a complete LaTeX document or just a document fragment. Exporting in LaTeX creates a TeXdraw representation of the current view. TeXdraw can be found at `http://ctan.org`.

SVG Export the animation in SVG format. Exported SVG animations can be viewed with the Adobe SVG Viewer browser plug-in that can be obtained from `http://www.adobe.com/svg`. The SVG can be configured (Fig. 6). It can be compressed with gzip that is supported by Adobe's SVG plug-in, an animator panel can be included (Fig. 7), and the animation can be scaled. If the animator panel is not added to the animation two more options are available: the length of the pause between steps (in seconds), and the length of one step of animation (in seconds).

PNG Export the current view or the animation in PNG format. You can select whether to export the current view (default) or the entire animation as a series of pictures. If the picture series is selected, the files will be named `<name><step>.png`.

For more information about exporting TeXdraw pictures and SVG animations, see the tutorial at `http://www.cs.hut.fi/Research/MatrixPro/tutorials/export_tutorial.shtml`.

C Advanced Installation

If you already have a copy of the Matrix framework that is suitable for this version of MatrixPro, you can download this type of release. If you have the matrix.jar file in the same directory as matrixpro.jar, you can start it by typing:

```
java -jar matrixpro.jar
```

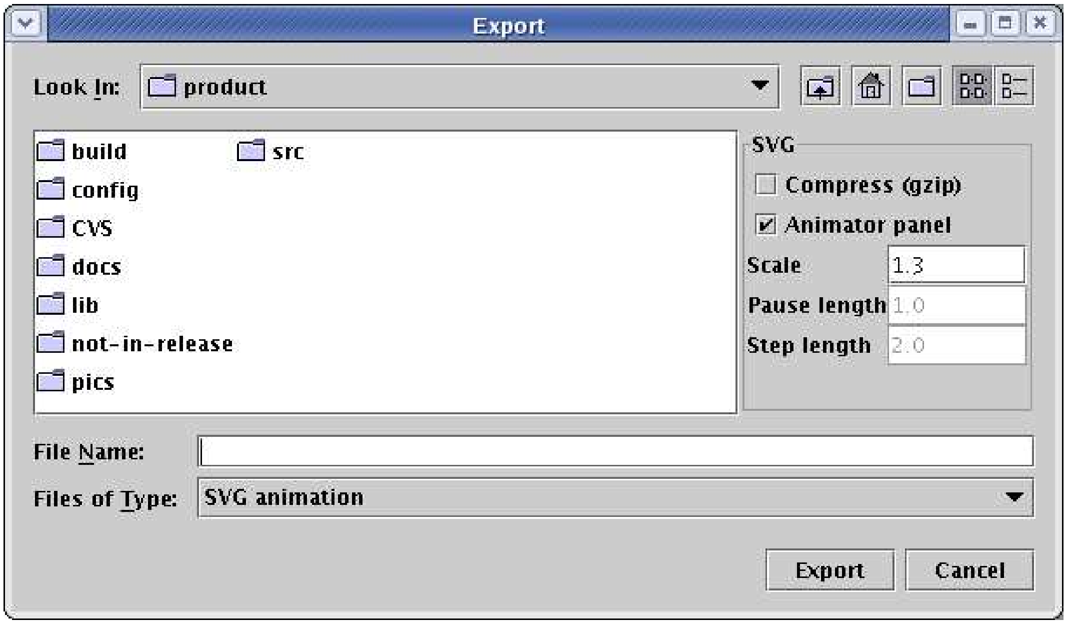


Figure 6: SVG export dialog

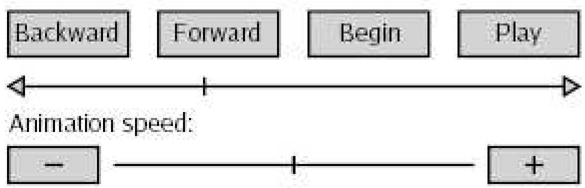


Figure 7: Animator panel in SVG animation

In general, you can start it by typing:

```
java -classpath <path-of-matrix.jar>:matrixpro.jar
matrixpro.ui.MainFrame <configuration-file>
```

Substitute the path of the matrix.jar file for <path-of-matrix.jar> and the path and filename of the configuration file for <configuration-file>. You can get the configuration file from the download page.

The source release does not contain the Matrix framework. To be able to compile the system, you should download an appropriate version of Matrix. By default, the MatrixPro lib directory is expected to contain matrix.jar. The source release of MatrixPro is available as a gzipped TAR or as a ZIP file. To build MatrixPro you should have Apache Ant installed. The build file (`build.xml`) contains the following targets:

clean Removes the compiled class files.

compile Compiles all source code.

javadoc Generates the Java API documentation for the system.

run Compiles and runs the application.

manual Compiles PDF, PS and HTML versions of the manual from LaTeX source (LaTeX, dvips, ps2pdf and latex2html must be in the path).

The important directories are:

build/classes This directory contains the compiled classes, and is created only if you use the Ant build file provided with the source.

docs/javadoc Contains the Java API documentation of the system.

docs/manual Contains the user's manual.

lib Empty. This directory is where the ant build file searches for the matrix.jar file.

src Contains the source code of MatrixPro.

If you don't want to copy the matrix.jar file into the lib directory (or create a symbolic link if your file system supports them), you can modify the build.xml file. Change the value of the property named matrix.jar:

```
<property name="matrix.jar" value="\${lib.dir}/matrix.jar"/>
```

If you don't have Ant and you don't want to install it, you can try the following in the MatrixPro root directory (in Unix):

```
javac -classpath src:lib/matrix.jar src/*/*/*.java src/*/*/*/*.java
java -classpath src:lib/matrix.jar matrixpro.ui.MainFrame
```

Windows users should change the colons in the above commands to semicolons and the slashes to backslashes.

D Acknowledgements

MatrixPro was developed at the Laboratory of Information Processing Science, Department of Computer Science and Engineering, Helsinki University of Technology. The project leader was Lauri Malmi and the lead designer and programmer was Ari Korhonen.

Other persons who have contributed to MatrixPro are Petri Ihantola, Ville Karavirta, Jan Lönnberg, Jussi Nikander, Riku Saikkonen, Otto Seppälä, Panu Silvasti, and Kimmo Stålnacke.

Persons who have contributed to this User's manual are Mordechai Ben-Ari, Ville Karavirta, Ari Korhonen, Jussi Nikander, and Kimmo Stålnacke.

E GNU Free Documentation License

Version 1.2, November 2002

Copyright ©2000,2001,2002 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The **"Document"**, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as **"you"**. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A **"Modified Version"** of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A **"Secondary Section"** is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The **"Invariant Sections"** are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The **"Cover Texts"** are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A **"Transparent"** copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called **"Opaque"**.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The **"Title Page"** means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section **"Entitled XYZ"** means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as **"Acknowledgements"**, **"Dedications"**, **"Endorsements"**, or **"History"**.) To **"Preserve the Title"** of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties–for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright ©YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

> with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.