BON Software Architecture Illustrator (BONsai)

User's guide version 1.4

Kim Waldén
August 2003

TABLE OF CONTENTS

1 Introduction

1.1 Background

There has long been a need for a good drawing tool supporting the BON notation. Ever since the book describing the BON method [1] was published in 1994, I have had requests from BON users all over the world asking for a simple way to draw nice-looking BON diagrams and have them included in system documentation, and never been able to answer satisfactorily.

The only tool available over the years with BON support has been EiffelCase from Eiffel Software Inc, recently integrated into EiffelStudio (see www.eiffel.com). However, this tool is part of a full Eiffel development environment, which may be a bit too heavy for users who only want to communicate designs on paper and those unfortunate users who do not have the option of developing their systems in Eiffel. Also, there is no support in EiffelStudio for dynamic diagrams.

Actually, there is another tool, BON User Support (BONUS), which produces excellent diagrams [5]. I developed BONUS in 1993 and used it to produce all the diagrams in the BON book, and all slides for all BON tutorials that I gave at various TOOLS conferences in 1994-1998. The trouble is that it is not really usable for anyone outside the community of Troff wizards (a shrinking lot these days). However, BONUS still serves as reference for the quality of printed BON diagrams.

Over the years, engineers at Enea have drawn BON diagrams extensively on whiteboards in all projects where Eiffel was used as development language, but also with Smalltalk, C++ and Java. Various general drawing facilities has been used to include the diagrams in system documentation, such as IslandDraw, FrameMaker and MS Word, but the result has never been satisfactory. General drawing tools have proven just too awkward to use for sizable diagrams, and some BON primitives just impossible to get right. Particularly the client/supplier double arrow, which is a major component of all static diagrams, usually comes out very ugly, especially when folded.

I therefore became quite excited when I looked into the functional spreadsheet model and drawing primitives in Visio for creating specialized extensions (called Visio Solutions), and the possibilities of integrating VBA macros to do more complicated things. It seemed that Visio could be an adequate platform for creating a BON drawing tool with reasonable effort, but this would have to be tested in practice and the trouble was to find the time.

Around Christmas 1999, I suddenly had an opportunity to spend three weeks working on a BON solution with Visio 5.0, and after some initial struggle understanding the model (being new to both Visio and VBA) I progressed very quickly. However, just as I thought I was more or less done with a first version, I suddenly ran into severe problems with stochastic behavior and strange crashes and finally had to give up. Since the Visio documentation (including the MSDN groups) did not give enough details about the underlying execution model to explain this behavior and there was no one to ask, I just couldn't proceed and soon more pressing issues took over.

Some two years later, when Microsoft (who acquired Visio Corporation around 1999) released its version Microsoft Visio 2002, I decided to have another go, and in June 2002 I was again able to concentrate on the problem. And lo and behold, after the initial effort of recalling the details of my previous work, and with some minor changes, the problems were gone! I was able to finish a first version in less than a week, and it turned out that since I had initially developed most of the code for Visio 5, it was possible to port the finished version back to Visio 5 with only straightforward modifications having to do with some changed Visio syntax.

I have consciously refrained from using the new features of Visio 2002, since a BON drawing tool will be of more use to people if its runs on as many Visio installations as possible. Since BONsai also seems to work without problems on Visio 2000 (released between Visio 5 and Visio 2002), the whole range of Visio installations from about 1997 until present is covered.

Because the exact execution model of Visio is not documented, it is impossible to make the Visio Solution totally robust. So I am sure there are still ways to use the tool that will lead to strange behavior, but for normal practical purposes it seems to be stable enough for public release.

There are some problems with the print quality of scaled diagrams using Visio 5 or Visio 2000, which are common to all Visio diagrams. A later section will summarize my experiences with BONsai using different versions of Visio.

1.2 Scope of BONsai

The drawing tool is called BON Software Architecture Illustrator (BONsai). The bonsai metaphor is meant to suggest a carefully trimmed and nourished structure of minimal size, yet with all the features necessary to capture the essential properties of object-oriented software.

BONsai is implemented as an extension of the general-purpose drawing tool Microsoft Visio 2002, as a Visio Solution [2, 3]. This means that it is run from inside an execution of Microsoft Visio, so a user needs to install the latter tools first, which in turn means that there is a cost involved. However, the standard version of Microsoft Visio, which is not that expensive, is sufficient to run Visio solutions (as opposed to developing new ones) and the good news is that the BONsai extension is free.

My main objectives for a BON drawing tool have been:

- High-quality diagrams (at least close to those in the BON book)
- Easy to learn and (more importantly) easy to use once learned
- Easy to make small changes to complex diagrams
- Possible to work fast
- Easy to include BON diagrams as scaled figures in documentation (MS Word, FrameMaker)
- Support for both static and dynamic diagrams
- Available to all BON users at a price that most can afford

I think all the above objectives have been met with BONsai.

Beginning with version 1.4, BONsai includes support for graphical class interface descriptions, which was initially left out because of the poor text handling support in Visio. This means that the graphical notation of BON (the difficult part) is now covered almost completely. The only thing lacking is the multiplicity symbol for client relations, which in my experience is very rarely used, and the static and dynamic Charts (Class, Cluster, Events, Scenarios).

However, because of Visio's poor text handling, it doesn't seem feasible to create good support for these charts using Visio. They are much easier to create by other means, and have successfully been implemented in the past as predefined templates in MS Word and FrameMaker. The BONsai package may include such templates in a later release. Support for the BON process may also be desirable in the future, but in my view, the biggest obstacle by far to using BON has been support for drawing good quality static BON diagrams, which is not a trivial task.

1.3 Scope of guide

This guide explains how to use BONsai graphically for drawing BON diagrams with Visio. It does not discuss the BON notation as such. The reader is assumed to be already familiar enough with BON to know what diagrams he or she would want to produce. For details about the BON method and notation, see [1].

Some concepts of Visio will be discussed, but only with respect to the use of BONsai. For a general text on how to use Visio and its many accompanying specialized Solutions, see for example [4]. For a complete guide on how to develop Visio Solutions in general, see [2, 3].

2 Working with Visio Solutions

Visio is a powerful general purpose drawing tool with many features, which take some time to master. However, the general idea with Visio Solutions is that they can be made very much self-contained. A user should be able to just combine and modify predefined graphical objects within a solution, and not worry about the rest of Visio, which makes it very easy for newcomers to get started.

Still, there are some basic Visio concepts a Solution user should know about. Below is a brief overview of the most important of these for a BONsai user. For complete information, the reader is referred to [2, 3, 4] and the on-line help of Visio.

2.1 Shapes

A geometrical object in Visio is called a *shape*. Each shape has its own *shapesheet* (quite similar to an Excel spreadsheet), which controls the appearance and behavior of the shape. A shapesheet contains a number of predefined sections specifying how the shape should be drawn on the drawing page, what will happen when you resize or rotate the shape, at what points other shapes may be attached to it, the size and other properties of the text attached to the shape, if any, etc. As with spreadsheet tools, the individual *cells* of the sections of a shape in Visio can be made to depend on each other trough functions, or *formulas*, so that when a part of the shape changes, other parts may automatically follow.

Whenever the user moves, resizes, or drags a shape on the screen, the formulas in the underlying shapesheet will be recalculated in real time to reflect the new appearance of the shape (and possibly other shapes which may be affected). This makes the manipulation of graphical objects direct and intuitive. The underlying shapesheet may be very complex, but the user doesn't need to know about it.

Each shape usually has an *alignment box* drawn around it when the shape is selected on the screen. The alignment box has a number of *handles* on it that the user may click and drag with the mouse to resize the object, provided resizing is permitted according to the shapesheet. Alignment boxes are only there for user manipulation, and never show up in a printed diagram.

2.2 Connectors

There are two basic types of shape in Visio, one-dimensional (*1-D*) and two-dimensional (*2-D*) objects. The two types behave differently, and their alignment boxes look different. 2-D shapes have no specific direction and their alignment boxes look like rectangles with eight handles for resizing (unless its height or width is set to zero, in which case the eight handles become three). 2-D shapes usually occupy two dimensions on the screen, like rectangles and ellipses, but the geometry section may specify any shape.

1-D shapes, on the other hand, have a *begin* point and an *end* point, which show up in the alignment box as small squares containing the symbol × for the begin point and the symbol + for the end point. They usually look one-dimensional on the screen, like lines and arrows, but may have any shape.

The begin and end points of 1-D shapes can glue to specified *connection points* on 2-D shapes. When the 2-D shape is moved, the corresponding glued point on the 1-D shape moves with it stretching the 1-D shape (rubber banding). Therefore, 1-D shapes are often called *connectors*. In BONsai the client, inheritance, and message relations are defined as connectors, while clusters, classes, class interfaces, objects, object sets, and object groups are 2-D shapes which may be glued together by relations.

2.3 VBA macros

The shapesheets may specify that certain VBA (Visual Basic for Applications [6]) functions and procedures, known as macros, be called as a result of certain events, such as the user moving or resizing an object, selecting an entry in a shortcut menu, or dragging a new object onto a drawing page. The VBA routines have a full API (the Visio Type Library) at their disposal for manipulating the shapesheets of the objects in a drawing in order to achieve more complex behavior than is possible by just entering static formulas once and for all in the shapesheets.

The continuous reevaluation of shape sheet formulas and corresponding redrawing of objects combined with dynamic manipulation of the formulas through VBA macros triggered by certain events is a very powerful drawing engine for building specialized graphical environments such as BONsai.

2.4 Masters and stencils

A shape or group of shapes may be saved on a *stencil* as a *master*. A stencil may contain any number of masters, which may be dragged onto a drawing page, thus creating copies of the corresponding shapes. These copies are called *instances* of the corresponding masters, and serve as canned starting points to be further manipulated by the user to make up the final drawing.

Each master has a name and can have an icon and explanatory text attached. The instances of a master initially inherit all the master's fields, which means that if the master is later edited, all fields of all instances created from that master in a drawing that has not been altered, will automatically follow.

Editing a master means editing its shapesheet manually, which is not recommended unless you know exactly what you are doing. The inner workings of some of the BONsai masters with attached VBA macros is quite complex, and the smallest mistake may result in havoc.

2.5 Document stencils

Each Visio drawing has a dedicated stencil attached to it, the so called *document stencil*. When you create a new drawing, an empty document stencil is automatically created. You may open any number of auxiliary stencils and use their masters as part of your drawing. Whenever you drag a particular master onto the drawing page for the first time (regardless of from what stencil), a copy of the master is stored in the document stencil. Since the document stencil is always stored with each drawing, this has the effect of making all Visio drawings self-contained.

You may also drag masters from other stencils directly onto the document stencil for later use. As long as a master is in use (at least one instance remains in the drawing page), Visio will not let you delete it from the document stencil without a warning. On the other hand, deleting all instances from the drawing does not automatically delete the corresponding masters. To get rid of unused masters that are no longer of interest, you must delete them manually from the document stencil.

In order to view the document stencil of a drawing (if it is not shown already), select menu *File* ▶ *Stencils* ▶ *Document Stencil* (in Visio 5: *Window* ▶ *Show Master Shapes*). At any point, you may save a shape that you have manipulated on the drawing page as a new master by dragging it on to the document stencil. There you may change its name, edit a suitable icon for it, and use it as any other master to build new drawings. This makes it easy to work incrementally with Visio.

A word of warning: if you create a new icon for a saved master by editing a bitmap for it, make sure to uncheck the box *Generate icon automatically from shape data* in the *Master properties* dialog (select the *Update: manual* radiobutton in the *Icon* box of *Properties* in Visio5). Otherwise your carefully designed bitmap will get overwritten by Visio the next time you edit the master's shapesheet, which can be very annoying.

2.6 Visio files

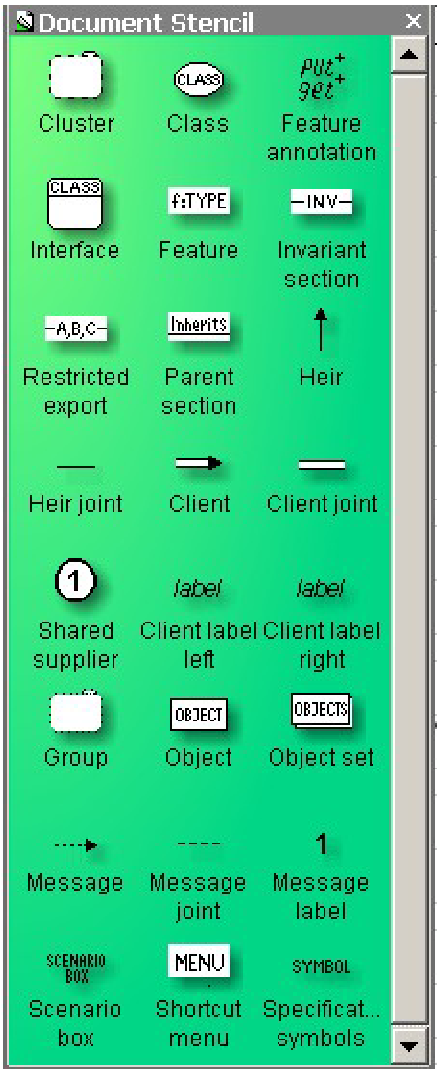
All Visio files have the same internal format, which includes a drawing page, its document stencil, a default VBA project, and a workspace list. You may save your Visio drawing in three different ways: as a *stencil*, as a *drawing*, or as a *template*.

If you save the drawing as a stencil (file extension *.vss*), the idea is to use the masters of the stencil as building blocks in other drawings. This is also referred to as a *standalone stencil* to differentiate it from a document stencil. When you open a stencil file, the default behavior is to open it read-only, so its masters can be used in other drawings without fear of destroying the original. If you want to modify the stencil you have two options: either open a copy of the stencil, or open the original for edit. The choice is made by selecting from the dropdown list next to the Open button in the Open Stencil dialog (or selecting a radio button in the same dialog in Visio 5).

If you save your work as a drawing (file extension *.vsd*), the drawing and its document stencil are saved, just as when saving as a stencil, but in addition, references to all auxiliary stencils and other windows that may be open are saved too. So when you later open the drawing file, these auxiliary windows are opened again (provided the corresponding files have not been deleted or moved), and you may continue to work where you left off.

Finally, if you save your drawing as a template (file extension *.vst*), things work exactly like when saving as a drawing, with the difference that when the file is later opened, an unnamed copy of the drawing is opened rather than the drawing itself. The idea is to use the drawing as a starting point to create new drawings without fear of destroying the original.

3 Working with BONsai



When you open the BONsai template (*Bonsai.vst)*, a blank drawing page occupies most of the Visio window and the document stencil is shown on the left. The document stencil may be shown as a floating window or docked as a toolbar, and may be resized according to taste. Below it is shown floating with its masters arranged in three columns

The stencil contains 24 masters, whose icons and names are fairly self-explanatory (provided you know the BON notation). The heir, client and message joints are shapes that can be tailored to join several relations into a complex forked relation. An explanatory text is displayed as a tool tip if you hold the mouse cursor over an icon for a few seconds.

You create a drawing by left-clicking on masters and dragging them onto the drawing page. We will explain the details for each one of the masters in the sections to follow. The first fifteen are for static BON diagrams, the next seven for dynamic diagrams, and the two last ones for setting shortcut menu options for your drawings and for copying and pasting special symbols into your class specifications, respectively.

When you have finished working with a BON diagram, you should save it as a Visio drawing (extension .vsd). If later reopened, the masters and macros will still be there, so you can continue to make modifications and save new results.

Besides their normal formats, Visio drawings may be saved in a multitude of forms, such as bitmap (.jpeg, .gif, .tiff, .bmp), PostScript (.ps, .eps), web page (.htm, .html), enhanced metafile (.emf), and more, which makes them easy to include as part of documentation, for example in Microsoft Office documents or Web pages.

An even easier way to create figures in, for example, an MS Word document, (if the figures are not to be maintained for use elsewhere) is simply to select a number of elements in a Visio drawing, copy them to the clipboard, and then paste them directly into the document.

I have experienced some problems with cut and paste of Visio 2000 drawings using COM objects, but pasting in .emf format using *Paste special* (which is usually preferable anyway) seems to work fine.

There is a shortcut menu that is shown when you right-click on a shape. The entries above the first horizontal line on the menu (if any), are BONsai entries, while the rest are Visio standard entries, such as Cut, Copy, Paste etc. To remove the Visio standard entries (except help), which is usually desirable to get cleaner and more comprehensible menus, first double-click on the MENU master (with Visio 5 you need to select the master and then choose *Edit master* from the shortcut menu), then right-click on the rectangle in the popup window and select *Minimal menu*, then close the popup window. The full menu can be restored, if needed, by repeating this procedure and select *Original menu*.

Most of the masters have a *Custom properties* menu entry, which when selected will bring up a dialog box with a number of settable fields. Each field has a short explanatory label, and if you select a field by clicking into its value part, a more comprehensive explanation is displayed in the *Prompt section* of the dialog box.

3.1 Supplementary Visio drawing facilities

Although combining BONsai masters is all you need to do to draw BON diagrams, all the standard Visio drawing primitives are still available as supplement. Usually, you do not need them, but there are some basic Visio facilities that are very useful, and should not be overlooked.

Selection

Shapes are selected or deselected by *left-clicking* on them. *Shift-click* will add an object to a previously selected set, or remove it from the set if it was already in the set.

A very convenient way of selecting a great number of objects in one area is clicking and dragging the mouse on the drawing page, which will produce a *rectangular lasso* that will select every object that is completely inside the rectangle when the mouse button is released.

Grouping

A set of selected objects may be *grouped* by selecting *Shape* ▶ *Grouping* ▶ *Group* in Visio's main menu. From then on the selected objects are kept together as one unit. The group gets its own alignment rectangle that can be moved or resized as if it was one shape. You can ungroup them again whenever needed by selecting *Shape* ▶ *Grouping* ▶ *Ungroup*.

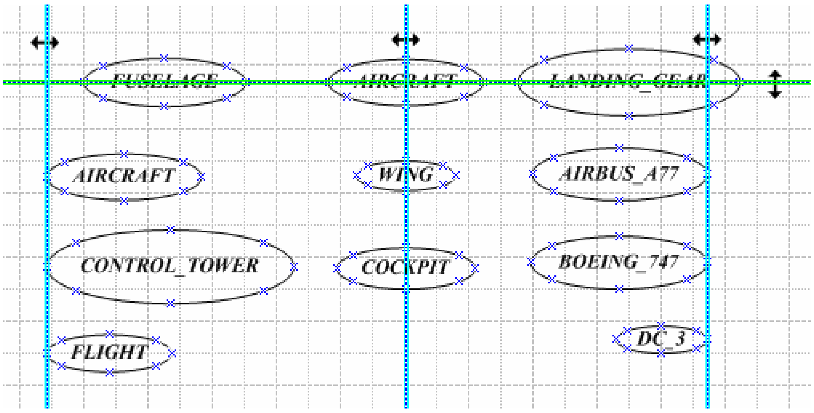
You do not need to ungroup each time you want to make a local alteration to one of the shapes in the group. The first time you click on a grouped object, the alignment box of the group lights up indicating that the group has been selected. If you immediately click again on the same object, Visio assumes that what you are really after is that particular object, and selects in inside the group.

Guides

Guides are horizontal and vertical rulers that can be rolled onto the drawing page and used for easy alignment of objects. You obtain a new guide by dragging it from one of the scaled ruler areas (top horizontal or left vertical) onto the drawing page. 2D objects can be glued to a guide, so that when the guide is moved all objects will follow (unless the gluing option has been turned off).

Guides are only used as a drawing aid, and are not visible on the printed diagram. When selected they light up in green or blue, otherwise they appear as faint blue lines. You can hide them altogether by toggling the menu entry *View* ▶ *Guides*, so they will not clutter up your diagram while you work with other parts. If you want to completely remove a guide that is no longer needed, just select it and press *delete* on the keyboard.

Below is an example with three vertical and one horizontal guide (the horizontal guide is green simply because it was selected first). Whenever one of the vertical guides is moved the corresponding set of left-aligned, centered, or right-aligned classes will follow. When the horizontal guide moves, the three topmost classes will follow. Class *AIRCRAFT*, being attached to both guides, will follow if either of the horizontal or middle vertical guide is moved.



Snap to grid

To aid the user with quick free-hand alignment of objects without having to go through the special alignment menu described below, the Visio *snap-to-grid* feature is on by default. This means that objects that are moved by dragging the left mouse button will exhibit a jerky behavior, only stopping at offsets that are multiples of the current grid spacing (often something like every 5 pixels).

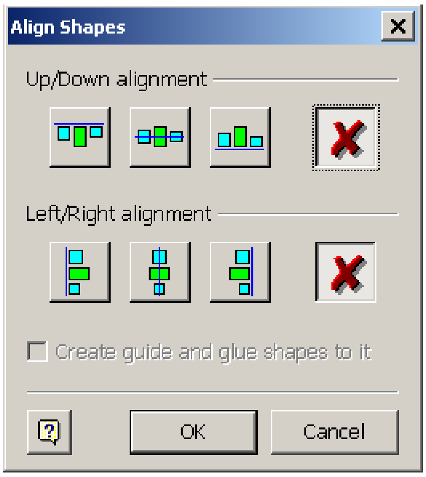
This is often convenient, but it may sometimes drive you crazy when you try to position an object at an exact point, only to find that some invisible force is moving the object into either just after or just before the point where you want it to be. Then it is nice to know that you may turn off the feature by selecting *Tools* ▶ *Snap & Glue* and uncheck the *grid* box in the *snap to* column.

Exact positioning

A selected object or group of objects may be moved one pixel at a time by pressing the arrow keys of the keyboard (Visio 2000 or later). Keeping an arrow pressed down will repeat the movement. This is very practical for exact positioning, particularly when *snap to grid* is in effect, since this pixel movement bypasses any snapping.

Alignment

You may align a selected set of shapes by choosing *Shape* ▶ *Align shapes* on the main menu. From the popup dialog you may then pick your mode of alignment, as shown below.



3.2 BONsai and MS Word

Perhaps the most important application of BONsai is to create BON diagrams for inclusion in systems documentation written using an electronic word processor, which in Windows environments usually means MS Word. Two typical ways of accomplishing this works well with Visio 2002 / Word 2002:

1. During a Visio session, copy any desired diagram or part of a diagram from the drawing page to the clipboard, and then paste it directly into an MS Word document. After pasting, the corresponding figure may be scaled to the desired size by right-clicking on the resulting picture object and selecting Format Picture in MS Word. This is a very fast and convenient way of working when the figures you are creating are not to be reused in some other context.

   The pasted figures become COM objects, which means that you can actually edit them later from inside MS Word, automatically invoking Visio in the process (provided Visio is installed on the machine where the document is edited). However, the necessary VBA macros are not automatically included as a result of the pasting, and the document stencil of the figure only contains the BON masters that you have dragged onto its drawing page. So if you want to edit a BON diagram in your Word document, and possibly add some shapes that weren't there before, you need to proceed as follows.

Start a parallel Visio session and open the Bonsai.vst template in order to access its stencil and VBA macros. In the Visio main window, select Tools ▶ Macros ▶ Visual Basic Editor. This brings up an editor window with a sub window to the left titled "Project – Bonsai". Open the subfolder Modules (by clicking on the corresponding + symbol) until the module Bon_macros becomes visible.
Then, in your Word document, double-click on the figure you want to edit. This starts Visio on that BON diagram, which will automatically modify the menus of MS Word giving access to additional Visio menu entries. Select Tools ▶ Macros ▶ Visual Basic Editor, which will bring up an editor window similar to the one in the parallel Visio session with a sub window titled "Project – VisioDocumentVisioDocument". Then left-click and drag the Bon_macros module in the previous editor window and drop it in this sub window. This will copy the VBA macros and make a Modules folder containing Bon_macros appear also in the latter window.

Now you can start editing your BON diagram adding new shapes if needed by dragging the corresponding masters from the BONsai stencil in the parallel Visio session. If you want to edit more diagrams, you must repeat the copying of the module Bon_macros for each separate figure. When you save your Word document, the macros are saved with it, so for each figure the copying need only be done once.

2. If you find the above too heavy, an alternative way is to save a separate Visio drawing containing a copy of all the figures used in your document. Then if you later want to modify a figure, just select and edit the corresponding elements of this drawing and paste the result into your document thus replacing the modified figure.

   In this case, you will not utilize the COM editing facilities directly on the figures in your Word document, so you may instead of ordinary paste use *Edit* ▶ *Paste special* to save space. Choose "Picture (Enhanced Metafile)" from the format list of special paste.

3.3 Conversion to PDF

If you want to give people easy access on the web to documents containing BON diagrams, converting them into PDF (Portable Document Format) is a good way of enhancing portability. The full Acrobat tool from Adobe contains Acrobat Distiller, which can convert PostScript into PDF, as well as PDFMaker, a plug-in to MS Office which can generate PDF directly from a Word document.

Unfortunately, things are not as simple as one would want. It turns out that the PDF generated by PDFMaker from Word documents doesn't handle Visio diagrams properly, yielding scrambled results for the BON diagrams. The same happened when I tried to use the Acrobat Distiller virtual printer to create PostScript. The only way I have found that will produce good quality results is to proceed as follows.

Download the Adobe PostScript driver from *www.adobe.com* (currently version 1.0.6) and install a supplementary printer in your system by double-clicking on the downloaded file (*winsteng.exe*, if you selected the english version). During the installation procedure you will be asked to supply a PPD file (PostScript Printer Description) for your printer. It doesn't really matter what physical printers you have on your system, because this printer is only going to be used for printing to file.

Therefore, you can supply any valid PPD file that describes a printer supporting Adobe PostScript Level 2 or higher. After installation you can use the new printer to print your document to a file, and then run Acrobat Distiller on that file to produce the final PDF-file. The intermediate .ps file may not be printable on your printer if you used a foreign PPD file, but once the file has been converted to PDF, that file can be printed on any printer through the freely available Adobe Acrobat Reader.

If you include color (as in this document) and you want to preserve color in the resulting PDF document, you need to use a PPD file for a color printer. The current manual was produced with the technique described above, supplying the PPD file corresponding to the color printer HP LaserJet 2500 (downloadable from www.hp.com).

If you also want to be able to print the intermediate .ps file on your own printer (which may be an advantage for testing purposes), you may have to install the Adobe PostScript driver twice: once for generating color PostScript and once for generating black-and-white PostScript, using a PPD file for the second installation that is consistent with your printer.

3.4 BONsai with older versions of Visio

BONsai seems to work well with both Visio 2000 and Visio 5, although I have only done a limited amount of testing using these older versions. Visio 5 requires a separate template *bonsai_v5.vst* in order to work, because of syntax changes that were made between Visio 5 and Visio 2000.

Visio 5 also lacks certain features, like the *Enable live dynamics* of Visio 2000/2002, which means the redrawing will only take place when a dragged object is released. Also, you cannot use the keyboard arrows to move objects a pixel at a time, which is a drawback.

Styles used, like *Dotted*, *Dashed* and *Dashed2* show up as icons in the document stencil of Visio 5, which clutters the stencil somewhat. Just ignore them.

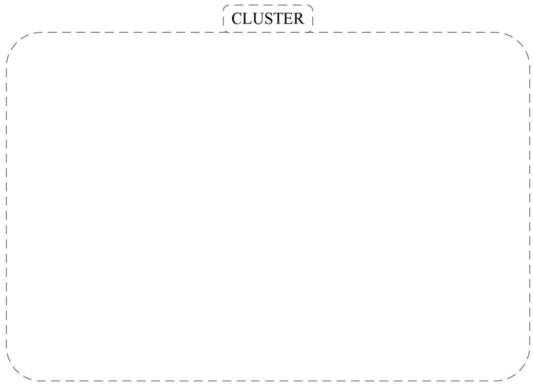
The print quality is definitely better with Visio 2002, since the older versions produce ellipses that are only approximately rounded and cruder spline drawings, somewhat blurring for example the graphical difference between a deferred class marker (asterisk) and a persistent class marker (filled blob).

So upgrading to Visio 2002 is worthwhile if you can afford it. Finally, a word of warning: I have only tested the Visio versions with Word 2002. Hopefully, earlier versions of Word should be ok, but you never know until you have tried in practice.

4 BON static diagrams

4.1 Clusters

The cluster master looks as follows when dragged onto a drawing page:



To give it a proper name, double-click on the name tag and change the name. All letters become upper case automatically, so you do not have to hold down the shift key when entering a name.

There are eight predefined *connection points* corresponding to the compass directions *n, s, e, w, ne, nw, se, sw*, to which you may glue client arrows. These are shown as thin light blue × markers.

When the shape is selected, a yellow *handle* (green in Visio 5 and Visio 2000) is shown on the name tag. You may drag this handle to slide the name tag along its cluster borderline to obtain a better position to suit your diagram.

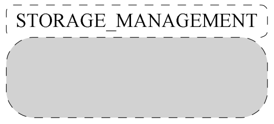
The way Visio works, it is not possible to constrain the movement of a handle to a desired curve, but only to lock one of its x- or y-coordinates. This means you cannot slide the tag around the corners. In order to change sides for the tag, right-click on it and select one of *Name tag above, Name tag below, Name tag right, Name tag left* from the shortcut menu. For the same reason, there is no way to prevent you from dragging the name tag too far so it will actually leave the cluster rectangle, making the drawing look broken. If this happens, just drag the tag back again.

If you have been dragging the tag and want to position it in the middle of the chosen side, right-click and select *Center name tag*.

To resize the cluster, select it and drag one of the eight green handles of the cluster rectangle. The cluster body is transparent, and whatever classes and other clusters may be inside the cluster will not be affected by the resizing. If you drag one of the handles *n, s*, or *e, w*, the cursor turns into a ↕ or a ↔ respectively, and only that side of the rectangle whose handle you are dragging will move. If you drag one of the corner handles, the cursor will turn into a slanted ↔ and the two adjacent rectangle sides will move, resizing in two directions at once (standard Windows behavior).

To move the whole cluster, drag it by grabbing either a point on the rectangle border a bit away from the resize handles, or anywhere on the cluster name tag (the cursor will turn into four small arrows, one in each compass direction). With Visio 2000 or later, you may use the arrow keys to move a selected cluster (or any group of objects) in small steps, which is very convenient for fine-tuning.

The cluster can also be compressed by selecting *Compressed form* from the cluster's shortcut menu:



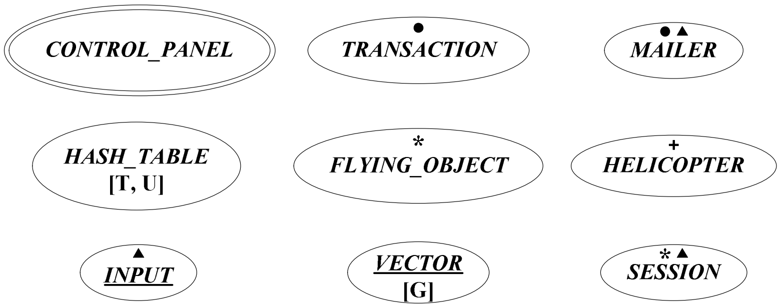
In the custom properties dialog, you can set individual values for the sharpness of the roundings of the cluster body and name tag, as well as horizontal and vertical margins for the name inside its tag. For a compressed cluster, the body's north and south connection points are moved to the outside on the name tag, if it is on the same side, to make the connection point accessible for client arrows.

4.2 Classes

The class master looks as follows:

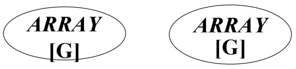
To set its name, double-click on the ellipse and change the text. You may annotate the class with the BON attributes *deferred, effective, persistent, external, root class* by selecting the corresponding entry from the shortcut menu of the class ellipse. Some attributes are mutually exclusive, such as deferred vs. effective, deferred vs. persistent, and then setting one will automatically unset the other.

Annotations for *reused* and *generic* classes are made in the class name by choosing underline mode or adding the bracketed generic parameters, respectively. To get the normal font style for the generic part, double-click and select only this part of the text and uncheck the Italic, and possible Underline modes. Below are some examples of annotated classes:



Based on the width and height of the class name, BONsai attempts to adjust the size of the class ellipse so the result will look good. However, Visio returns the width and height of the text as the width and height of the smallest surrounding rectangle, which means there is no difference between, for example, a two-line class name consisting of a long first line + a short second line, two long lines, or a short first line + a long second line. The requirements for such names to look good inside its surrounding ellipses, however, are different because of the rounded shape of the ellipse. So BONsai cannot make all combinations look good, but only try to use default adjustments that will work for most cases.

To deal with cases where the default adjustments does not look good, you may alter some of the values by setting custom properties of individual classes. The following figure shows the class *ARRAY* [G] with no adjustment to the left, and with custom properties *Line spacing* set to 100% (instead of default 120%), *Name adjustment* set to -0,2 mm, and *Height/width (2 rows)* increased from 0,4 to 0,45 yielding better adjustment to the right.



One nice thing is that if you have a number of classes in your diagram that need the same adjustment, you can select them all and then select *Custom properties* from the shortcut menu invoked by right-clicking on the whole group of objects. Any field changes in the resulting popup dialog will be applied to all the selected classes.

There are eight connection points on the circumference of the class ellipse, corresponding to the compass directions *n, s, e, w, ne, nw, se, sw*, to which you may glue client arrows.

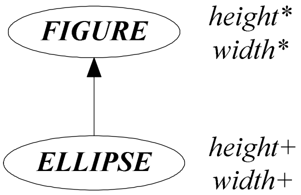
You can move a class ellipse by grabbing and dragging it, but you can also move it by selecting it and then use the arrow keys (Visio 2000 or later). The latter method gives much better precision in positioning a class exactly where you want it in a diagram (holding an arrow key down will make the selected class slowly move across the page).

You cannot resize a class ellipse directly, since it will always follow the size of the enclosing text block. Instead, you may set various adjustment values in the Custom Properties dialog. Besides the line spacing, vertical name adjustment, and eccentricity of the ellipse for classes whose name contains two rows, you can set the eccentricity value for classes with one and three-row names, add extra width to the ellipse, and change the width and adjustments for the class annotation markers.

4.3 Feature annotation

Sometimes it is desirable to attach some very simple feature annotations to classes in static diagrams to show, for example, at what level certain features get effected. The feature annotation master is a plain textbox that can be used for this purpose. To make an annotation, drag a corresponding master on to the drawing page, double-click on it, and change the text to suit your feature(s).

Initially, you may glue the left side of a feature annotation to a connection point on a class. To change this, select one of *Glue right side, Glue top, Glue Bottom* from the annotation's shortcut menu. An example with two annotations glued to the east connection point of the two classes is shown below:



In the Custom properties dialog box (accessible from the shortcut menu) you may change the bottom, top, left and right margins of the annotation text block, as well as its line spacing, in case you need special adjustments.

4.4 Class interfaces (background)

The BON graphical class interfaces was initially excluded from BONsai because of Visio's poor text handling support. Each shape can only contain one text block, and it is not possible to integrate text into geometry sections. In fact, you cannot even change the text automatically from VBA macros or by any other programmatic means. The only way to create or modify a text block is by opening it (usually by double clicking on the shape) and manipulating the text interactively using mouse and keyboard.

Since graphical feature specification in BON with signatures and contracts typically involves many special symbols and frequent font changes in the text to look good, one would need shortcut keys for quick insertion of individual text elements. However, nothing like that is available in Visio. You cannot even change the font for the next character to be typed, since that changes the font for the entire text block. To enter a character in a different font, you must first type it, then select it and then change the font, and the only way to input special characters like mathematical symbols is by cut-and paste.

This makes it tedious to input large specifications, which is why I gave up on providing graphical interface support for the first released version of BONsai in 2002. It didn't seem that important, since there is always the possibility to use the BON textual interface descriptions instead, for which templates can easily be built using MS Word or FrameMaker.

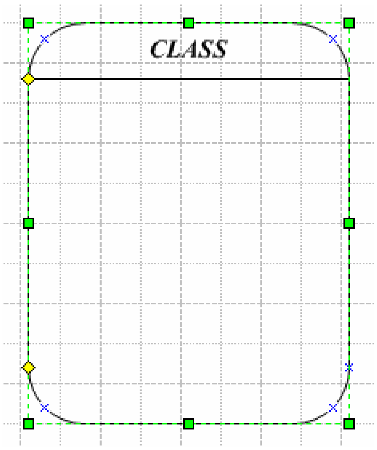
However, after having some requests from BONsai users, I decided to give it a new try in 2003. There are still some advantages with including support for graphical class interfaces:

1. It is sometimes worth a little extra point-and-click to be able to produce nice looking interface diagrams for classes central to a design.
2. For class interfaces that do not have many contract elements, the input is easier.
3. There is always the chance that Visio will include shortcut keys for symbols and font changes in a future release, which would speed up input by an order of magnitude.

So starting with version 1.4, BONsai does include support for class interface specification, which is described in the next section.

4.5 Class interfaces

The interface master is the expanded form of a class, which is a rounded rectangle with the class name in a header section. When dragged onto the drawing board it looks as follows:



To set its name, double-click on the generic text "CLASS" and change it (the characters will be upper case, so there is no need to press the shift key). The class interface can be decorated with the same markers (deferred, effective, external, persistent) as compressed classes, using the shortcut menu. The markers appear in the rightmost part of the class header.

Feature specifications, which are separate objects, can be glued to connection points along the left border. Each connection point for features has a corresponding handle that may be dragged vertically to adjust the gluing position. There are 15 such handles with their respective connection points, all located at the upper left where the rounding begins, as seen in the figure (you only see one, because they are initially all on top of each other).

To add a feature, get a connection point by pointing to the stack of 15 handles and drag the uppermost handle down into the desired position. Then drag a Feature object (described below) onto the drawing board, fill it with appropriate text, and attach its left side to the connection point. If you ever need more than 15 features in the same interface object, you will have to put more than one feature specification into some of the feature specification objects, in order not to run out of handles.

Besides features, the class interface may contain section delimiters for class invariant, restricted export clause, and inheritance. These delimiters involve lines extending from the left border to the right border of the class interface, and so need to be glued to both sides in order to follow graciously when the interface is resized. To that end there are 5 additional handles stacked in the lower left, each with two corresponding connection points, one on the left interface border and one on the right.

To add a section marker, drag one of the lower handles vertically into position. Then drag the desired section marker object (Invariant, Restricted, or Parent) onto the drawing page and attach its left and right sides to the corresponding connection points.

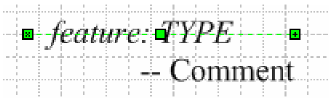
Thanks to the handles, you may easily adjust features and section markers, and rearrange them if desired. When the interface becomes more stable, you can hide the handles for increased readability by deselecting "Show handles" in the shortcut menu. They can be shown again when something needs to be moved.

Besides the connection points for gluing features and section markers, there are connection points at the usual compass directions *n, s, e, w, ne, nw, se, sw,* to which you may glue client and inheritance arrows. To get more connection points for gluing relational arrows, proceed as for clusters and object groups, using *Adjust beginning, Adjust end,* etc on the arrows (see description of client arrows in 4.13).

In the custom properties dialog, you can set individual values for the sharpness of the roundings of the interface rectangle, width and adjustment for class header markers, name adjustment, and line spacing.

4.6 Feature

The feature master looks as follows when dragged onto a drawing page:



feature: TYPE
-- Comment

Since we want the gluing point to be in the upper left corner of the feature, the text pin is set accordingly and the height of the bounding box is set to zero. To set the feature's signature and possible contract, double click on the feature text and modify it.

The default tab positions (not present in Visio 5) is set so you can use one tab character to indent arguments and contract elements, and two tab characters to indent the feature header comment. In order to facilitate pasting of graphical assertion and signature symbols, one item of each symbol is collected in the master Specification symbols (see below).

In the custom properties dialog, you can set top, bottom, left and right margins for the textblock, and its line spacing.

4.7 Invariant section

The invariant master looks as follows:

——— **Invariant** ———

The section delimiter is to be glued between a pair of connection points taken from the handle stack on the lower left of the interface boundary.

4.8 Restricted export section

The restricted export master looks as follows:

——— *A, B, C* ———

The section delimiter is to be glued between a pair of connection points taken from the handle stack on the lower left of the interface boundary. To set the list of client classes, double click on the shape and change the text.

4.9 Parent section

The parent section master looks as follows:

Inherits: *A, B*

The section delimiter is to be glued between a pair of connection points taken from the handle stack on the lower left of the interface boundary. To set the list of parent classes, double click on the shape and change *A, B* part.

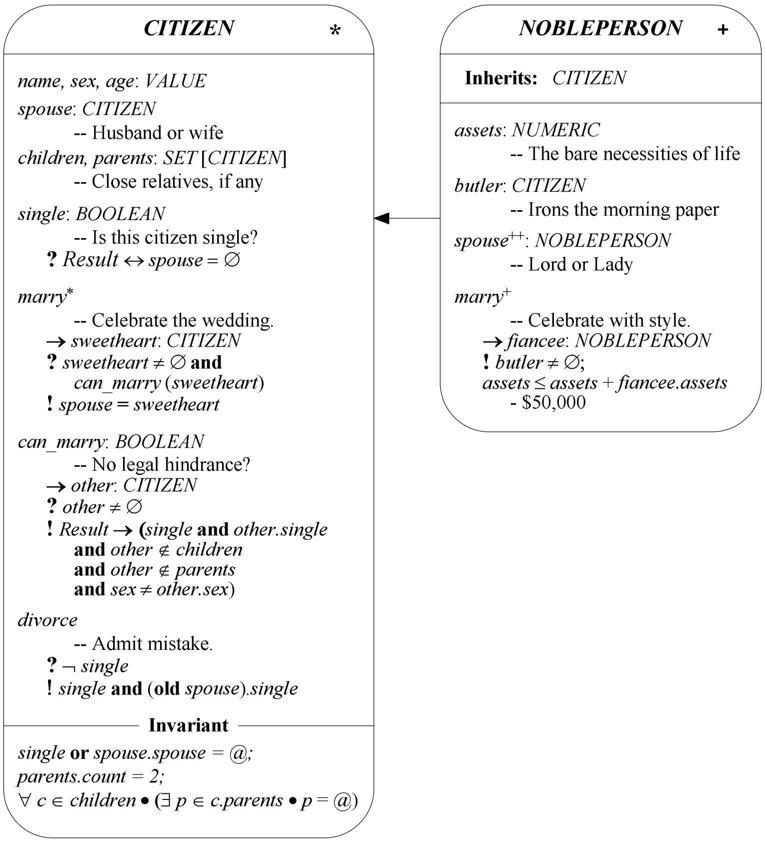
4.10 Specification symbols

This master contains graphical symbols used for feature signatures and contract elements, which are difficult to enter from the keyboard. It includes *, +, ++, which are elevated, and ^, ?, !, which are one point size larger than the rest.

∀ ∃ • ∅ ∈ ∉ ¬ ≤ ≥ ≠ → ↔ $^{*\ +\ ++}$ ^ **? !**

The recommended style for the pre- and postcondition characters **?** and **!** is boldface to make them stand out better (since we cannot get the framing). Unfortunately, boldface characters are pasted as normal characters by Visio, so you need to change to boldface again after pasting.

Below is an example of a pair of class interfaces taken from the BON book and drawn using BONsai, to show that we may come reasonably close to the quality of the book figures, although we have do without the rectangular framing of the pre- and postcondition symbols (? and !).

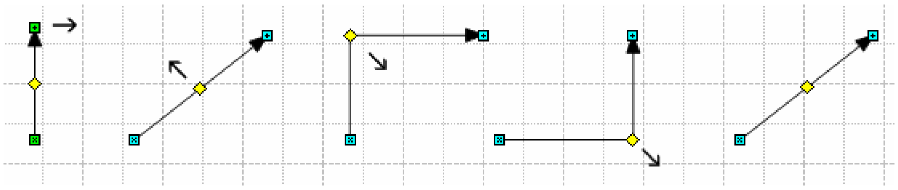


```
CITIZEN                                  *

name, sex, age: VALUE
spouse: CITIZEN
        -- Husband or wife
children, parents: SET [CITIZEN]
        -- Close relatives, if any
single: BOOLEAN
        -- Is this citizen single?
    ? Result ↔ spouse = ∅
marry*
        -- Celebrate the wedding.
    → sweetheart: CITIZEN
    ? sweetheart ≠ ∅ and
        can_marry (sweetheart)
    ! spouse = sweetheart
can_marry: BOOLEAN
        -- No legal hindrance?
    → other: CITIZEN
    ? other ≠ ∅
    ! Result → (single and other.single
        and other ∉ children
        and other ∉ parents
        and sex ≠ other.sex)
divorce
        -- Admit mistake.
    ? ¬ single
    ! single and (old spouse).single
                 Invariant
single or spouse.spouse = @;
parents.count = 2;
∀ c ∈ children • (∃ p ∈ c.parents • p = @)
```

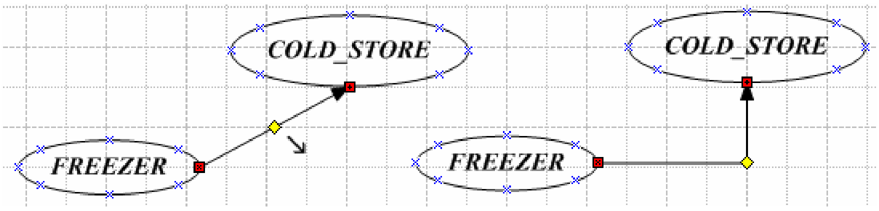
```
NOBLEPERSON                    +

 Inherits:   CITIZEN

assets: NUMERIC
        -- The bare necessities of life
butler: CITIZEN
        -- Irons the morning paper
spouse++: NOBLEPERSON
        -- Lord or Lady
marry+
        -- Celebrate with style.
    → fiancee: NOBLEPERSON
    ! butler ≠ ∅;
    assets ≤ assets + fiancee.assets
        - $50,000
```

4.11 Heir

The heir master is the inheritance arrow, which is vertical when dragged on to a drawing page (below left). You can rotate and stretch it by dragging its Begin point, marked ×, or its End point, marked +. When the arrow is selected, a yellow handle is shown in the middle of the arrow body (green with Visio 5 and Visio 2000). When its orientation becomes slanted, you can turn it into an angular arrow with two legs, one horizontal and one vertical, which is the standard BON convention of displaying inheritance and client relations, by dragging the handle in the direction where you want the corner. This will yield one folded variant, or if you change your mind, the other (below middle). If you drag a corner outwards, it will snap back and become slanted again (below right).



The begin and end points of the arrow, since it is a 1-D shape, can be glued to any connection point of a class or a cluster, which are 2-D shapes (the point on the arrow turning red when the gluing is effected). When the 2-D object is moved, the arrow point glued to it will follow, thus stretching the arrow. So a quick way of drawing an angular inheritance relation between two classes is first to drag a new heir master on to the drawing page, directly attaching one of its end points to one of the desired connection points on one of the classes, then attaching the other end point to the connection point on the other class, and finally drag the folding handle of the arrow to form the resulting relation, as shown below.

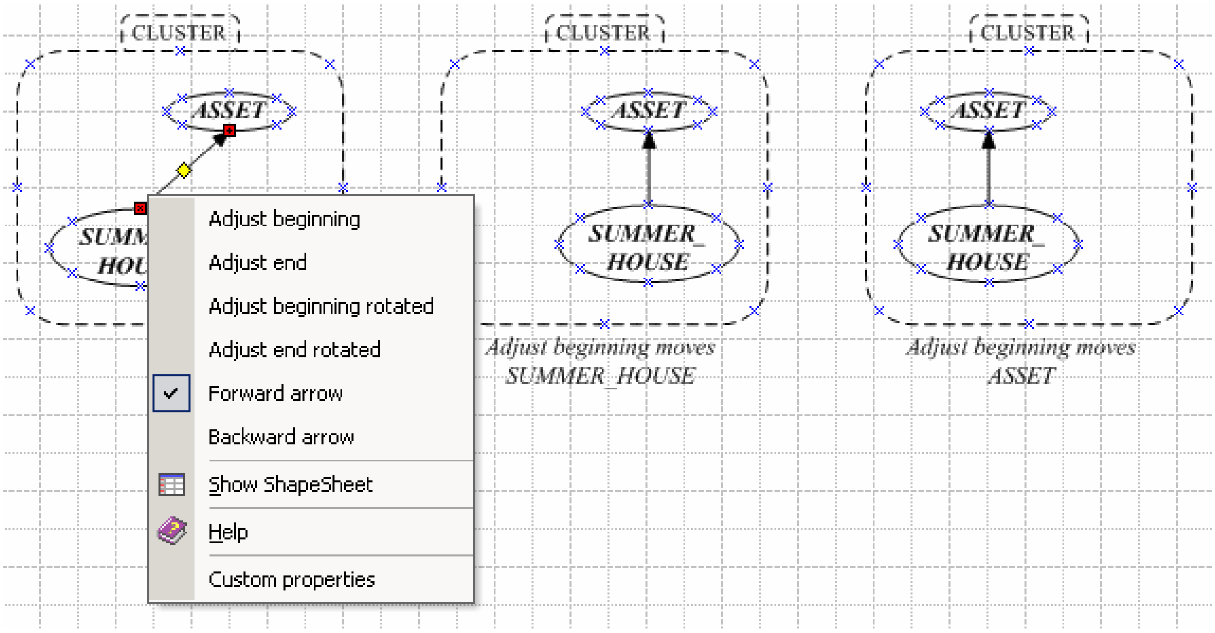


If the arrow is too close to being vertical or horizontal, it cannot be presented as folded, because lack of space for the arrowhead would lead to an ugly result. So if you change the slant of a folded arrow by dragging its arrowhead, you may watch the arrow changing from folded into straight line presentation and then snapping back into folded again as soon as the available space permits.

The minimum fold space required for a slanted arrow to show up folded may be changed for individual arrows in the Custom properties dialog.

Rather than creating an angular inheritance relation, you sometimes want a strictly horizontal or vertical relation between two classes, but may have a hard time moving the classes into the exact positions or to know when the arrow is really horizontal or vertical, not just approximately.

It is then convenient to select one of the entries *Adjust beginning* or *Adjust end* from the shortcut menu, which will move the class to which the corresponding point of the arrow is glued (see illustration below). If the arrow is more vertical than horizontal, an adjustment will yield a vertical arrow, otherwise the arrow will become horizontal.



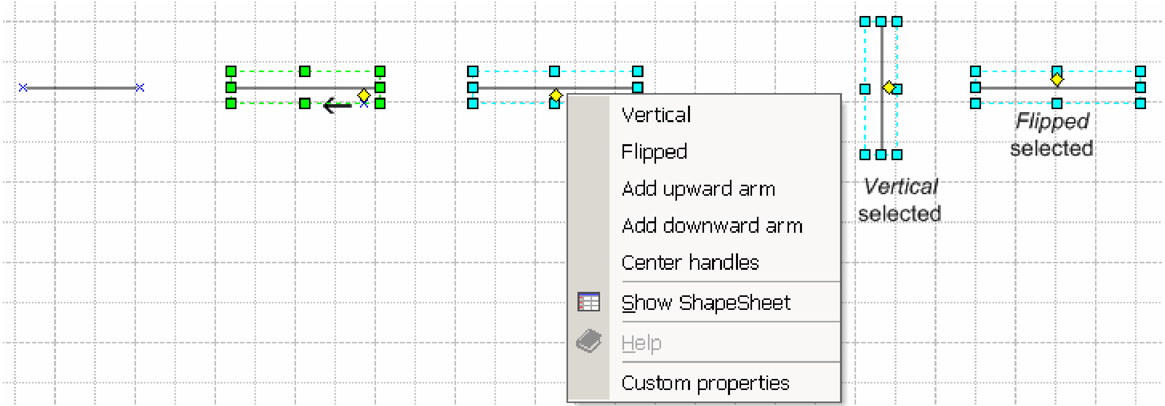
Selecting *Adjust beginning rotated* or *Adjust end rotated* will behave exactly like the corresponding *Adjust beginning* and *Adjust end*, except that the vertical versus horizontal choice is reversed. This is convenient when the connection points lie at an angle with respect to each other that would make the automatic scheme choose the wrong arrow orientation (see in particular gluing to clusters, in the client section below).

The entries *Forward arrow* and *Backward arrow* only make sense with forked relations, and will be explained in the next section.

4.12 Heir joint

The heir joint master is specially designed as a transformable building block for creating forked inheritance relations. When dragged on to a drawing page, it looks initially like a straight line with connection points at each end to which inheritance arrows may be glued. You may then add as many legs as you want, perpendicular to the line in both directions, by selecting from the shortcut menu to create forked relations of any desired complexity. The joint itself is a 2-D object, and cannot be glued to another joint, but there should never be a need for two adjacent joints in a diagram.

Below is an example of use, where a joint is first dragged onto the drawing page showing the two connection points (part 1). Then a downward leg is added (part 2). Each leg has its own handle that can be dragged to position the leg along the line, which are shown when the shape is selected (parts 2-5). The leg is then centered relative to the joint by dragging the handle, resulting in part 3, and the shortcut meny brought up. Parts 4 and 5 shows what happens when the *Vertical* and *Flipped* entries are selected, respectively. These may also be combined, which would yield a vertical joint with the handle to the left.



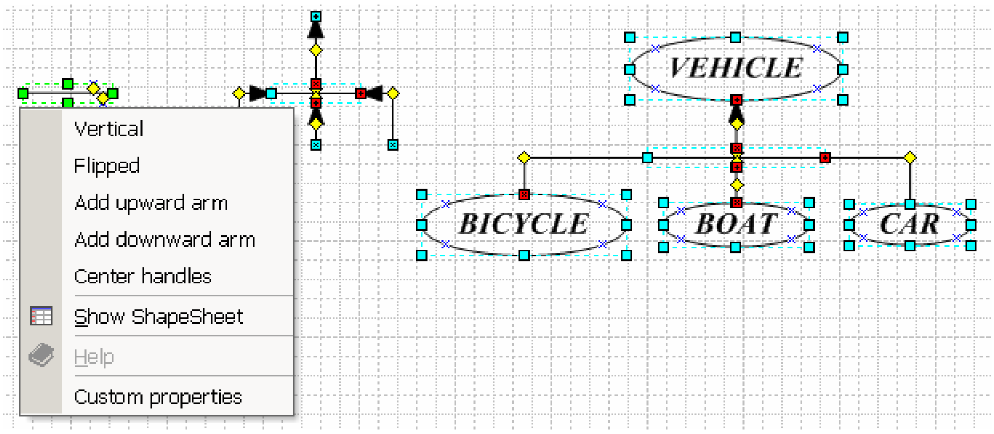
To use an heir arrow as a segment of a more complex inheritance relation, you sometimes need to remove the arrowhead, and it is also convenient to be able to reverse its direction when you are maintaining a large diagram. This is why you can freely select to show and hide arrowheads in each direction independently, even though some combinations will not produce well-formed BON diagrams.

Below is shown how a diagram with three classes inheriting from a common ancestor is gradually built. First an heir joint is dragged on to the drawing page and one leg in each direction is added. Next, *Center handles* is selected from the shortcut menu. This has the effect of moving the handles of the first upward and first downward leg to the center position of the joint.

Aligning handles exactly may be difficult to do just using free-hand move. If you are having difficulties dragging handles because of the resolution in a crowded diagram, you can always tell Visio to show it in larger scale which separates things much better, with Visio 2002, you can scale up to 2000 % without problem.

Then three new heir masters are dragged in to the drawing and attached to the downward leg and to the end points of the joint, respectively (middle part of figure). The arrows attached to the end points of the joint are made angular, as shown in the figure, which is convenient because when you later glue their beginning points to classes, the classes can be freely moved outwards, horizontally, without moving the joint; only the horizontal legs of the arrows will then stretch.

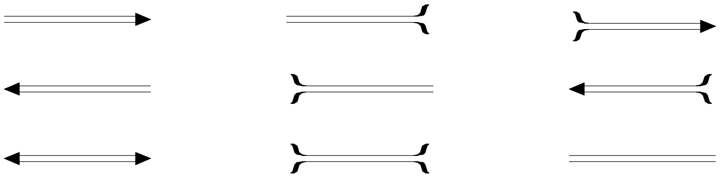
Finally, classes are attached to the end points of the angular arrows, their arrowheads hidden, and the two outer classes are adjusted horizontally using the arrow keys of the keyboard to make the diagram balanced.



In the Custom properties dialog, you can change the height of the legs, but this is usually not needed.

4.13 Client

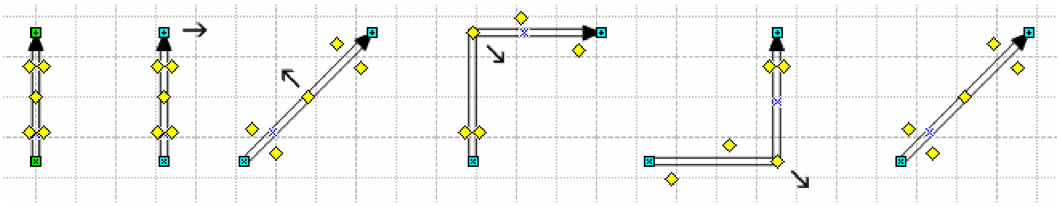
The client master is the client relation arrow, whose dual line appearance is notoriously difficult to handle for almost all general drawing tools. There are three possibilities for the arrowhead of each end, selectable through the shortcut menu: association (plain arrowhead), aggregation, or no arrowhead, yielding the nine variants below:



The variant with no arrowheads is used as a building block for creating forked client relations (see description of client joint in next section).

There are five connection points on each client arrow, four that can be used to glue a client label, and one central point where a shared supplier marker may be glued (see figure below). Each connection point has a corresponding handle, which can be dragged to change the point's position.

The label handles are always shown when the arrow is selected, while the shared supplier handle is hidden until you invoke the corresponding shortcut menu entry. Besides the connection point handles, there is also a folding handle for each arrow. You may create angular client relations in the same way as was described earlier for inheritance relations (see below). The client labels and shared supplier markers are described in a later section.

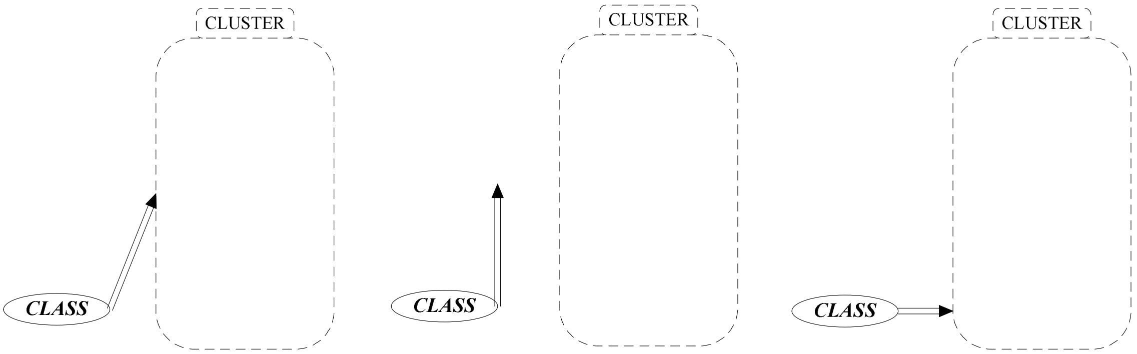


Sometimes you want to draw a client relation from, say, a class to a cluster without moving any of the two objects. Most likely, there will not be a connection point on the cluster's circumference to which you can glue the arrowhead and still keep the arrow horizontal or vertical (see figure below).

To remedy the situation, a special convention has been adopted for the commands *Adjust beginning* and *Adjust end* (and for their companion commands *Adjust beginning rotated* and *Adjust end rotated*) on the shortcut menu. Whenever the arrow point to be moved as a result of selecting one of these commands is glued to an object, and that object is a cluster, an object group, or a class interface, the object is not moved.

Instead, a new connection point is automatically created at the point of intersection between the arrow and the cluster, group, or interface border, provided the intersection point is not within the rounded area of the rectangle, in which case no gluing is effected.

Since clusters (as well as object groups and class interfaces) have a limited number of connection points, it may sometimes not be possible to find a suitable angle in order to obtain the desired arrow orientation with *Adjust end*. Here's where the adjust rotated variants come in handy, as shown in the following figure.

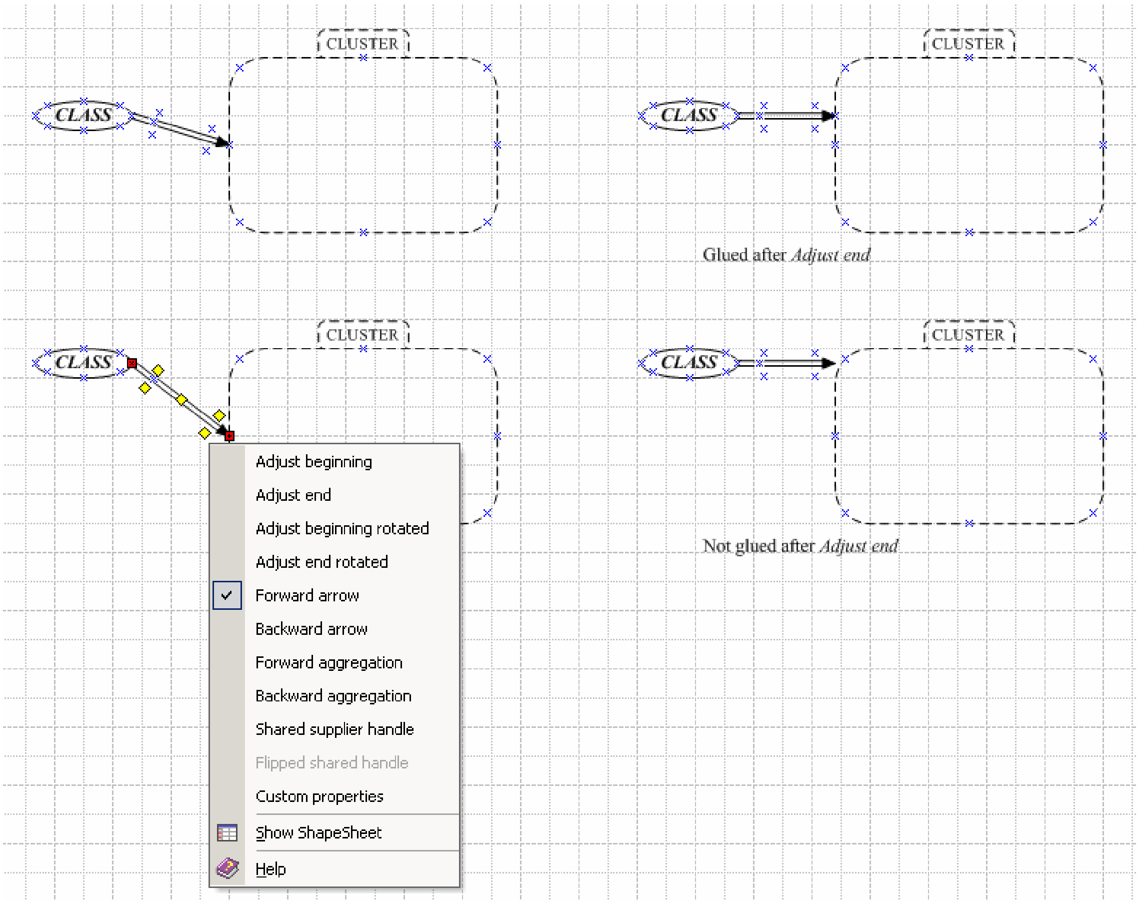


steep angle

after Ajust end
(not what was intended)

after Ajust end rotated
(the desired result)

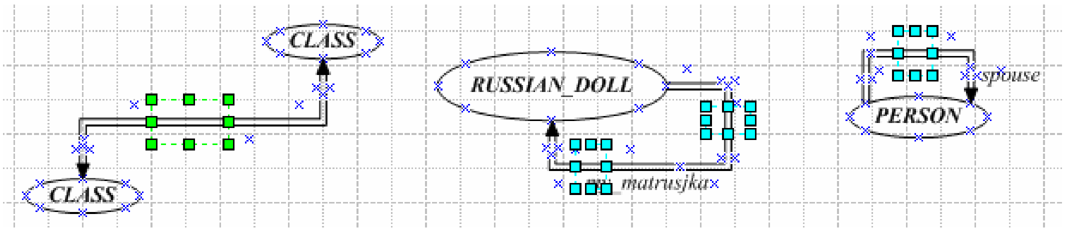
If the connection point to which the arrow was initially glued is a standard connection point for the cluster or group (n, s, e, w), the new connection point is created as a copy, otherwise the connection point is simply moved to its new position. The next figure illustrates the two gluing cases depending on whether the arrow hits the straight or curved part of the cluster (or object group or class interface).



4.14 Client joint

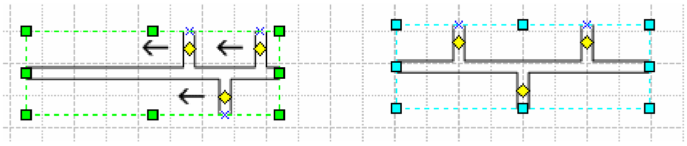
The client joint master is specially designed as a transformable building block for creating forked client relations, or relations that has more than one angle. It looks initially as a dual line segment with connection points at each end to which client arrows may be glued.

The initial shape is enough to create, for example tri-angular arrows and self referential client relations, as shown below. In the middle part of the figure, two joints are used, and *Vertical* was chosen from the shortcut menu for the rightmost joint.



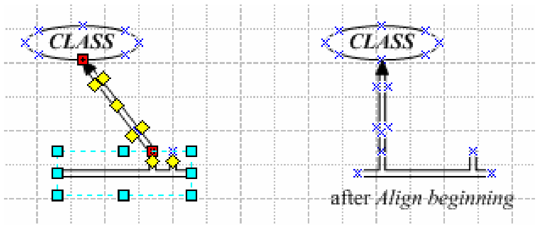
Since joints are 2-D objects, they need to be selected together with the classes and clusters indirectly attached to them when a diagram is moved, to prevent distorsion.

For more complex relations, you may sprout as many legs as you need from the joint, just as with the inheritance joint. Each leg has its own handle, so it can be dragged into position, as shown below.

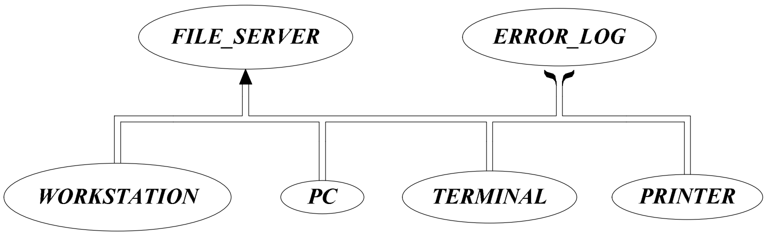


When dragging a leg of a client joint, the interface (in Visio 2000 and later) is much nicer with the *Tools* ▶ *Options* ▶ *Enable live dynamics* option selected. Then the pipe-like openings connecting the legs to the trunk will gently float across the figure in real time as you move the handle.

You may even drag a leg to cross another leg on the same side, and have the legs change places, which is sometimes convenient if they are connected to other structures. Another way of positioning a leg without dragging its handle is to attach a slanted client arrow to the connection point corresponding to the handle, and then select *Align beginning* or *Align end* (depending on which end of the arrow was attached) from the shortcut menu of the arrow, as shown below. This works also for inheritance and message joints.



Below is a figure from the BON book, which was built using one client joint with two upward and two downward legs.

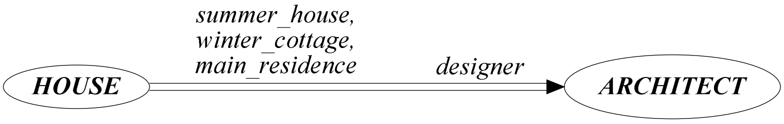


The client classes (2nd row) all have the *Forward arrow* menu entry unselected from the shortcut menus of their connecting client arrows, and the leftmost and rightmost classes are connected to the begin and end points, respectively, of the client joint, using angular arrows. By combining client joints and client arrows, you can draw almost any kind of complex forked relations.

4.15 Client labels

The two client label masters are text blocks initially containing the text *label*, which may be glued to connection points next to client arrows dedicated for this purpose. The connection points have corresponding handles, which may be dragged to adjust the position of a glued label.

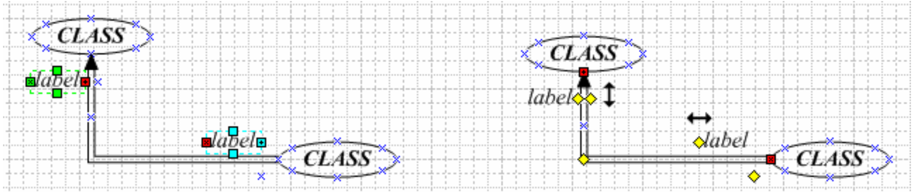
To set the text of a label, double click on it and change the text. The bottom, top, left, and right margins of the text block, its line spacing, and extra label space for horizontal arrows may be changed in the Custom properties dialog, which can be used to adjust multi-line labels. Below, a bottom margin of 24 pt was set to make the left, multi-line label appear in the proper position.



The *extra label space* in the Custom properties is intended to be used when you choose a different font face and/or font size by right-clicking on the selected label text. The white space between a horizontal client arrow leg and the label text may then have to be adjusted to look good.

Since we do not want the labels to be resizable, which would easily lead to inadvertent distortion, one of the end points need to be guarded and therefore cannot be glued. Initially, the left side of the label in the master *Client label left* , and the right side of the label in the master *Client label right* is gluable. This may be changed for each of the labels by checking/unchecking *Glue right* in the shortcut menu. The two variant masters are only there to save an extra right-click when you know beforehand what kind of gluing you want.

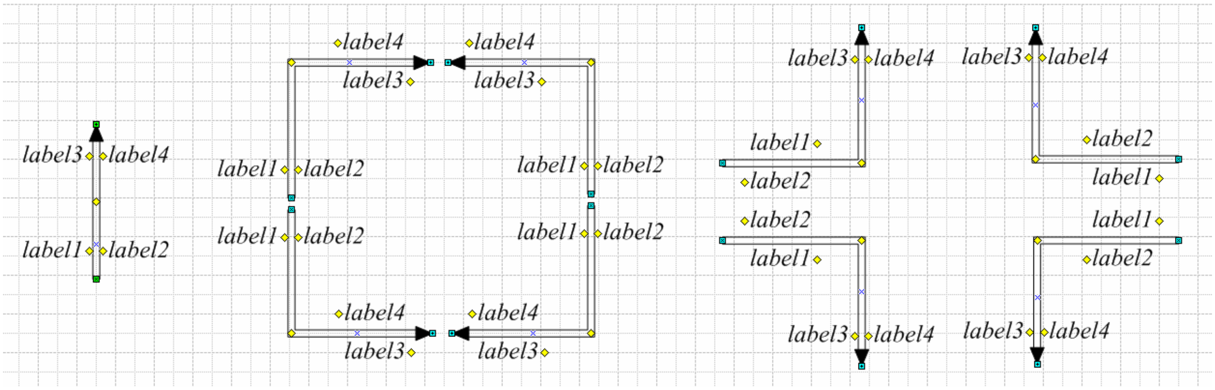
The handles of the label glue points can be slid along the X-axis or Y-axis, depending on to which leg of an angular arrow the handle is attached (see below, right part)



When you manipulate a client arrow without labels by changing its orientation, folding it etc., you may be surprised to find that the label handles are often displayed in assymetric positions with respect to the sides of the arrow legs. This is not a bug, but a way to make it easier to quickly glue labels to the handles, and have the initial result (when the text is still "label") look good.

If you start with the vertical client arrow with numbered labels shown to the left in the figure below, and then without touching the labels drag the arrow head around, folding the arrow into various positions using the folding handle, you will see the effect. No matter into which variant the arrow is transformed, the result is nicely aligned labels.

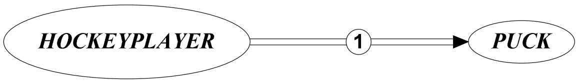
Changing label text will adjust the handle positions accordingly. Without this facility, transforming a labeled arrow would mostly lead to unreadable diagrams with overlapping label texts and random positions, making it more difficult to untangle the mess. When you have decided on orientation and folding of the arrow, you may easily do the final touch up by dragging the handles.

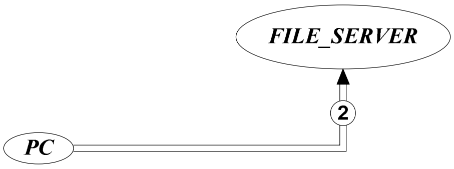


4.16 Shared supplier

The shared supplier master may be attached to the shared supplier connection point of a client arrow, whose handle becomes visible when *Shared supplier handle* is selected in the client arrow's shortcut menu. The handle may be used to reposition the marker. When the handle is visible and the arrow is folded, the *Flipped shared handle* can be selected making the marker change position to the other arrow leg.

The shared marker initially contains 1, but this may be changed by double-clicking on it entering another digit. Two examples are shown below:

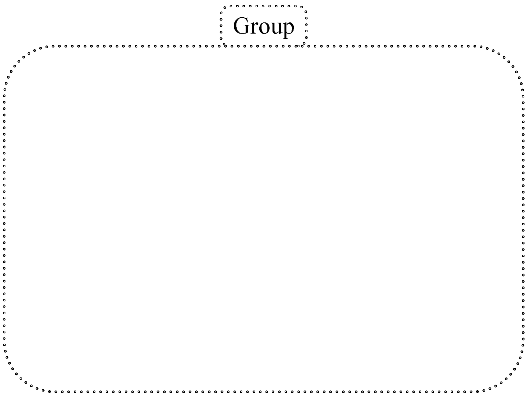




5 BON dynamic diagrams

5.1 Group

The object group master is similar to the cluster, but has a dotted borderline rather than a dashed:

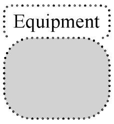


To name the object group, double-click on the name tag and change the text (first letter becomes capitalized automatically). Like for a cluster, there are eight predefined connection points corresponding to the compass directions *n, s, e, w, ne, nw, se, sw*, to which message arrows can be glued.

The name tag may be moved in either direction along the rectangle side on which it sits, by dragging the tag's handle. To change side, select one of *Name tag above, Name tag below, Name tag right, Name tag left* from the shortcut menu. To reposition the tag in the middle of its current side, select *Center name tag* from the shortcut menu.

Resize the group by dragging one of the eight green handles of the group rectangle (as described in more detail for clusters above). Drag the whole group by grabbing the name tag or some point on the group body's circumference other than the green handles, or use the arrow keys (Visio 2000 and later).

A group may also be compressed by selecting *Compressed form* from the shortcut menu.



In the custom properties dialog, you can set individual values for the sharpness of the rounding of the group body and name tag, as well as horizontal and vertical margins for the name inside its tag.

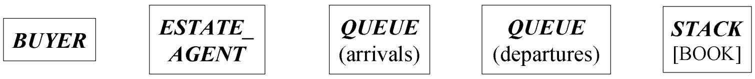
For a compressed group, the body's north and south connection points are moved to the outer border of the name tag, if it is on the same side, to make the connection point accessible for message arrows.

5.2 Object

The object master is a rectangle containing the name of the object's class. To assign the name, double-click on the object and change the text. Some examples are shown below.

The third and forth objects have qualifying identifiers attached to distinguish their roles. Such identifiers should be in lower case not to confuse them with types. To obtain this, the parenthesized line was individually selected while in text input mode and *Format* ▶ *Text* was chosen from the main menu. In the popup dialog, Case was changed from *All Caps* to *Normal* and the styles *Italic* and *Bold* were unset.

The fifth object has a generic parameter attached to specify the type instantiated from the generic class *STACK*. The parameter's point size was decreased from 9 pt to 8 pt to make the class name stand out.



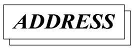
There are seven connection points on each side of an object rectangle, the usual compass point n, s, e, or w, plus six more, evenly distributed along the side. This makes it easy to find proper attachment points for message relations in crowded diagrams.

Objects are moved by dragging them, or by using the arrow keys on selected items. You cannot resize an object directly by dragging the handles of its enclosing rectangle, since it will always follow the size of the enclosed text block. Instead, you may set the top, bottom, left and right margins, and the line spacing of the text block in the Custom properties dialog.

Besides margins, you may set vertical name adjustment, line spacing as well as extra width and height of the object rectangle through the Custom properties.

5.3 Object set

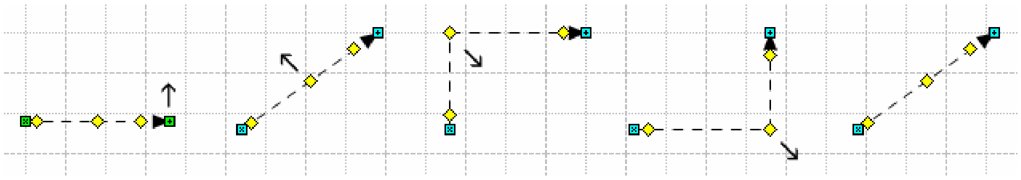
The object set master is similar to the object master, except that its rectangle is doubled to indicate a stack of objects rather than just one.



The connection points of the south and east sides appear on the outer rectangle to make them accessible for gluing message arrows. In all other respects, an object set behaves just like an object.

5.4 Message

The message master is the message relation arrow, which appears dashed to distinguish it from inheritance relations (these two relations should never be drawn in the same diagram, but still). It behaves very similar to the heir master (for more details, see 4.11).



There are two connection points on the arrow line, one for each direction in case the message is bidirectional, to which message labels may be glued. For each connection point there is a corresponding handle to drag attached labels into the desired positions.

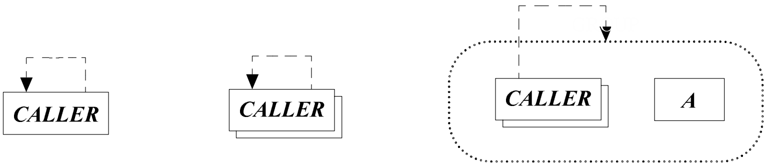
The begin and end points of the arrow may be glued to any connection point on an object, object set, or object group. Forward and backward arrowheads may be shown or hidden through the shortcut menu, and the standard *Adjust beginning* and *Adjust end* (as well as *Adjust beginning rotated* and *Adjust end rotated*) entries may be selected for rotating slanted arrows into horizontal or vertical orientation.

The minimal fold space (see 4.11) for the arrow may be set through the Custom properties dialog.

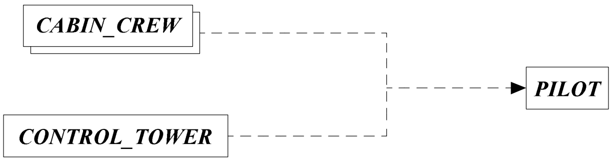
5.5 Message joint

The message joint is specially designed as a transformable building block for creating forked message relations. It is very similar to the heir joint, only that the line is dashed instead of solid. As with the other joint, you can add as many legs in the two directions as you want, and the joint may be rotated into vertical position and flipped along its trunk through selection of the corresponding entries in the shortcut menu (see 4.12 for details).

For example, one message joint with no legs can be used to build recursive messages:



One message joint with one downward leg added, followed by *Center handles* and *Vertical* from the shortcut menu is used to build the following fork:

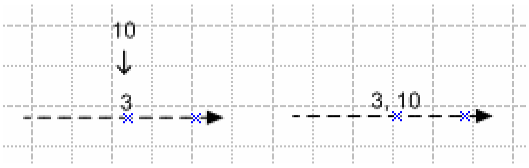


5.6 Message label

The message label master is a text block initially containing the sequence number 1. Each time a new master is dragged onto and dropped at the drawing page, or when a message label master is copied and pasted into the drawing, the sequence number is automatically incremented by one. This facilitates the generation of many message labels.

The observant reader has of course already noted that labels in dynamic diagrams often consist of lists of sequence numbers, rather than just a single number, which would seem to limit the value of the automatic label generation. However, BONsai has a nifty feature to counter this.

When you realize that the message number of the label just dragged onto the drawing page is actually supposed to be part of an earlier label, you just glue the new label on top of the old. This will automatically make BONsai add the number of the new label to the old, separated by comma, and then remove the new label (see below).



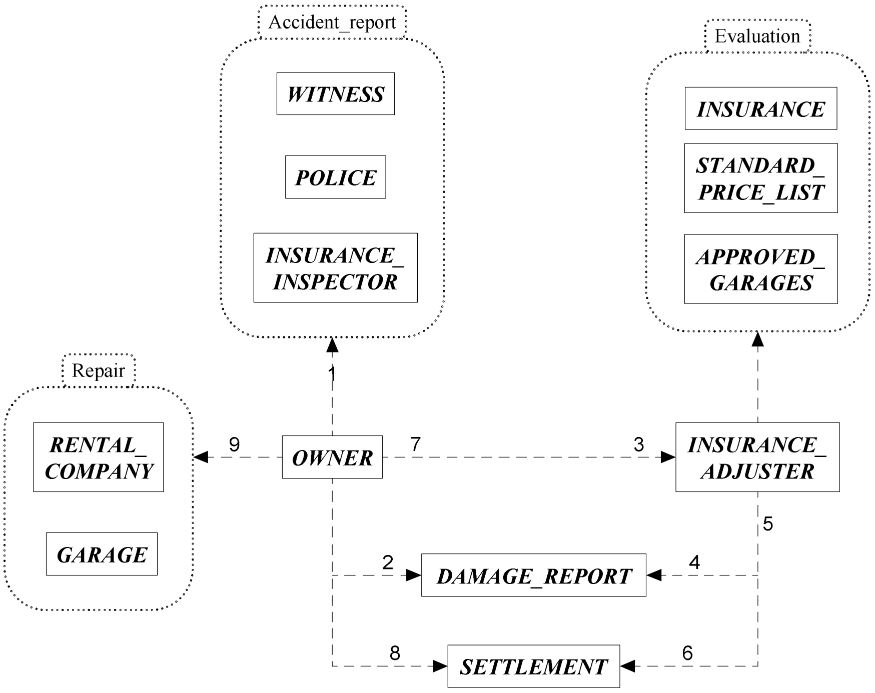
So as long as you do your labeling in the right sequence order, you do not need to edit the labels, just drop new ones on the drawing page and glue them to the corresponding message relations, one by one.

However, when building diagrams on the fly, you may change your mind several times, and then you can change the labels by double-clicking on them and edit the text directly.

The value used as starting point for generation of the next numeric label may be changed by selecting *Initialize sequence counter* from the shortcut menu. If the label text is a plain integer, and not a list; the counter will be re-initilalized to that value, so that the next label dropped onto the drawing will be one greater. If the text is not numeric, the counter is re-initialized to 1.

The labels are initially glueable at the bottom, which means they can be attached to connection points of horizontal message arrows to reside slightly above the arrow line. You may change the glue point by selecting *Glue left, Glue right, Glue top, Glue bottom* from the shortcut menu.

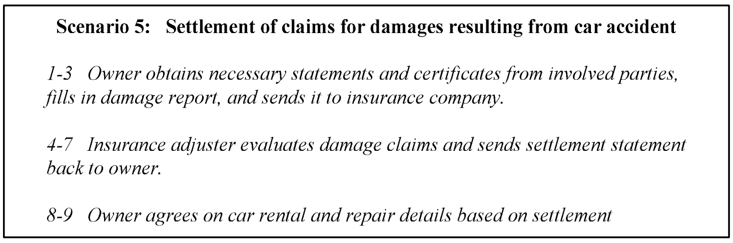
Below is an example from the BON book of a dynamic diagram with object grouping.



5.7 Scenario box

The scenario box master is simply a textblock inside a rectangle that will expand with the text. The scenario title should be entered in boldface, while the description entries should be non-bold italics, according to BON recommendations.

The initial text in the master, which is the title and description of the sequences corresponding to the above dynamic diagram, can be used as a starting point (see below).



Scenario 5: Settlement of claims for damages resulting from car accident

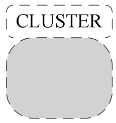
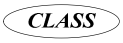
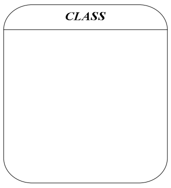
1-3 Owner obtains necessary statements and certificates from involved parties, fills in damage report, and sends it to insurance company.

4-7 Insurance adjuster evaluates damage claims and sends settlement statement back to owner.

8-9 Owner agrees on car rental and repair details based on settlement

6 BONsai master quick reference

Below is a table of all the masters along with their selectable shortcut menu entries and customizable properties with the corresponding default values.

MASTER	SHORTCUT MENU	CUSTOM PROPERTIES	DEFAULT
Cluster 	*Name tag above* *Name tag below* *Name tag right* *Name tag left* *Center name tag* *Compressed form*	Cluster rounding Cluster tag rounding Cluster name horizontal margin Cluster name vertical margin	6 mm 1 mm 4 pt 1,5 pt
Class 	*Persistent* *Deferred* *Effective* *External* *Root class*	Height/width Marker width Marker adjustement Name adjustment Height/width (2 rows) Height/width (3 rows) Extra width Line spacing	0,3 1,35 mm 0 0 0,4 0,5 0 120%
Feature annotation *put+* *get+*	*Glue left side* *Glue right side* *Glue top* *Glue bottom*	Bottom margin Top margin Left margin Right margin Line spacing	4 pt 4 pt 4 pt 4 pt 120%
Class interface 	*Persistent* *Deferred* *Effective* *External* *Show handles*	Marker width Marker adjustement Name adjustment Line spacing Interface rounding	1,35 mm 0 0 120% 7 mm
Invariant section **Invariant**	-	-	-
Restricted export section *A, B, C*	-	-	-
Parent section **Inherits:** *A, B*	-	-	-
Heir 	*Adjust beginning* *Adjust end* *Adjust beginning rotated* *Adjust end rotated* *Forward arrow* *Backward arrow*	Minimal foldspace	0,5 mm

MASTER	SHORTCUT MENU	CUSTOM PROPERTIES	DEFAULT
Heir joint —	*Vertical* *Flipped* *Add upward leg* *Add downward leg* *Center handles*	Link height	2
Client ═▶	*Adjust beginning* *Adjust end* *Adjust beginning rotated* *Adjust end rotated* *Forward arrow* *Backward arrow* *Forward aggregation* *Backward aggregation* *Shared supplier handle* *Flipped shared handle*	Client arrow width Minimal foldspace Extra label space	0,9 mm 0,54 mm 0 pt
Client joint ═	*Vertical* *Flipped* *Add upward leg* *Add downward leg* *Center handles*	Link height	2
Client label (left/right) *label*	*Glue right side*	Bottom margin Top margin Left margin Right margin Line spacing	0 pt 0 pt 2 pt 2 pt 100%
Shared supplier ①	-	-	-
Object group Group	*Name tag above* *Name tag below* *Name tag right* *Name tag left* *Center name tag* *Compressed form* *Name tag hidden*	Group rounding Group tag rounding Group name horizontal margin Group name vertical margin	6 mm 1 mm 4 pt 1,5 pt
Object ***OBJECT***	-	Name adjustment Line spacing Bottom margin Top margin Left margin Right margin Extra width Extra height	0 120% 2,5 pt 4 pt 2 pt 2 pt 0 0
Object set ***OBJECTS***	-	Name adjustment Line spacing Bottom margin Top margin Left margin Right margin Extra width	0 120% 2,5 pt 4 pt 2 pt 2 pt 0

MASTER	SHORTCUT MENU	CUSTOM PROPERTIES	DEFAULT
		Extra height	0
Message - - - ►	*Adjust beginning* *Adjust end* *Adjust beginning rotated* *Adjust end rotated* *Forward arrow* *Backward arrow*	Minimal foldspace	0,5 mm
Message joint - - - - -	*Vertical* *Flipped* *Add upward leg* *Add downward leg* *Center handles*	Link height	2
Message label 1	*Glue left side* *Glue right side* *Glue top* *Glue bottom* *Initialize sequence counter*	-	-
Scenario box **Scenario 1:** *Description*	-	Extra width Extra height Bottom margin Top margin Left margin Right margin Line spacing	0 pt 0 pt 4 pt 4 pt 4 pt 4 pt 120%

References

1. *Seamless Object-Oriented Software Architecture: analsysis and design of reliable systems*, Kim Waldén and Jean-Marc Nerson, Prentice Hall 1994
2. *Developing Microsoft Visio Solutions*, Microsoft Press 2001
3. *Developing Visio Solutions 5.0*, Visio Corporation 1997
4. *Microsoft Visio Version 2002 Inside Out*, Nanette J. Eaton, Microsoft Press 2001
5. *BONUS reference manual*, Kim Waldén 1993
6. *VB & VBA in a Nutshell*, Paul Lomax, O'Reilly 1998