

# Doxology – a Document-Oriented User Interface Model

---

Rick Jelliffe © 2006

Doxology is a model for document-oriented user interfaces for desktop applications for the office and school. It is broadly independent of platform technology and particular looks-and-feels or GUI controls (widgets) and is fairly general. Intended uses are:

A set of design requirements for developers of desktop applications and frameworks.

A benchmark by which platform and standard developers can measure the adequacy of their current specifications or technologies.

An articulation of user interface ideas for discussion in the author's company, Topologi.

Doxology is presented by topic. Example interfaces are presented for clarity and illustration, not as requirements.

The following terms are used:

**Document** – a cohesive set of information, normally storable in a single file or accessed from a single root file. For example, an XMLdocument.

**Program** – the running code

**Application** – a sub-program that handles a particular document type; the program may have multiple applications available, each application may have multiple documents open in them.

**Document window**- a main window feature one or more documents; the program may have multiple document windows, each application may be used for different documents in multiple document windows; typically, however, one document should not appear in more than one document window. When there are multiple document windows, the interface should provide a mechanism for starting and switching between them; for example, a Window menu item. A program may provide different branding for different document windows as if they were separate programs or distinct applications.

**View** – a different way of displaying and interacting with a document

## Screen Modes

The program can have different screen modes. These different screen components and interaction conventions for different use scenarios, users and hardware. Four screen modes are:

**Full screen** – the current document view occupies the full screen. Rationale: full screen mode is used by media players and presentation programs, and by pre-windowing system applications.

Pressing ESC returns to the default mode. Pressing PageUp or PageDn flips between pages. Pressing Space acts suspend and resume; in suspend mode, a *navigation box* may appear. Rationale: these are common controls.

**Wide screen** – the document window contains the document section and other controls tiled left and right. Rationale: larger screen of over 1000 pixels wide are common and tiling makes good use of screen space; wide screen mode suits applications that adopt modern look-and-feels (e.g. Office 2007, Apple , Aerith).

This view is the default and initial view on large screens.

In wide screen, the document window has the following sections, detailed later below:

**Document section** – for selection and display of the document. In the centre.

**Status bar** – for display of warnings, progress and status. At the bottom.

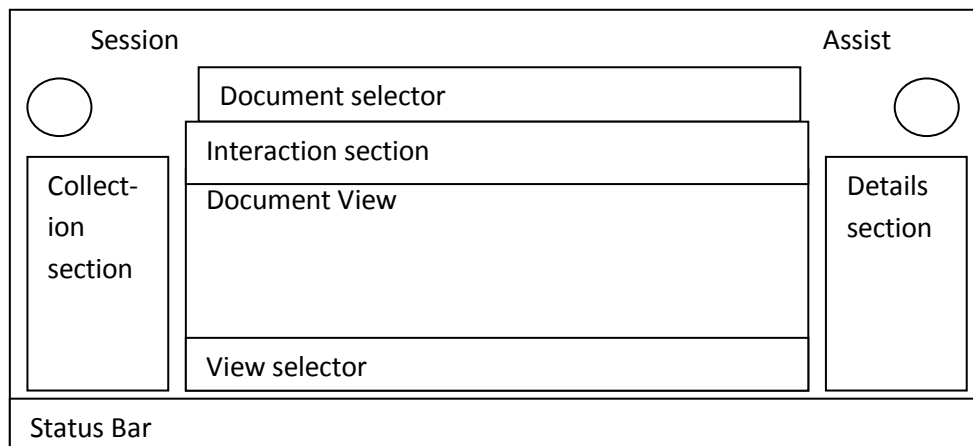
**Session controls** – generic operations for controlling documents, in the absence of menus. Top Left.

**Collection section** – for selection and display of controls that organize the documents. To the left.

**Details section** – for selection and display of controls that act on the currently selected document or object. To the Right.

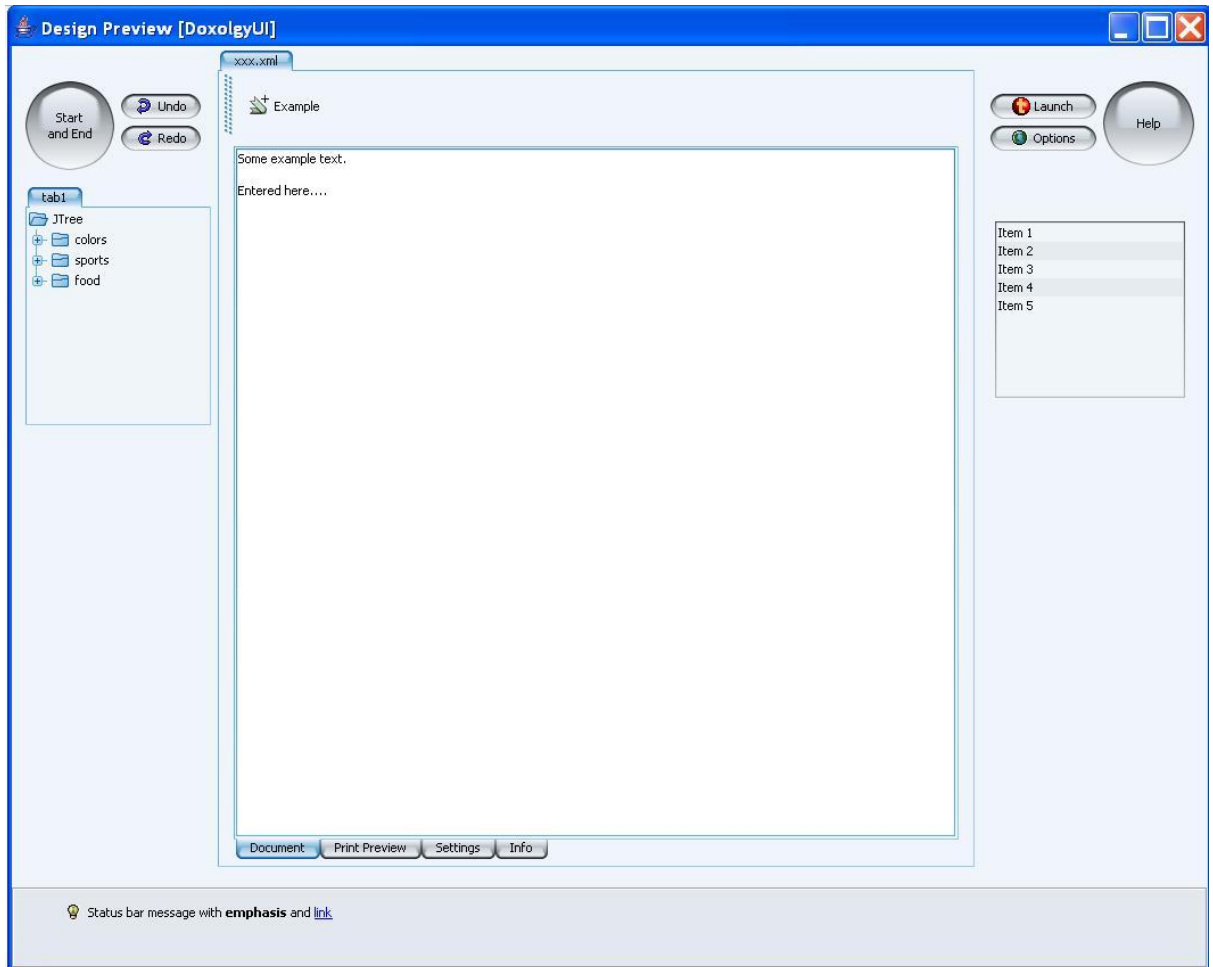
**Assistance controls** – for displaying the help system and user interface settings. Top right.

In wide screen mode, the menus are turned off by default (except for Mac OS X) and are rudimentary. Typically, there will be only one document window.



The view selector provides a mechanism for simplifying the interaction sections: only controls relevant to the current view should be displayed; the session controls and the assistance controls also aid this simplification. This reduces the need for logic to disable controls depending on the currently selected objects and their state. Similarly, the details section reduces the need for controls (e.g. a button) to display editable information about the currently selected object.

An mock-up screen for illustrative purposes follows:



**Small screen** – the document window contains the document section but other controls are floating pop-ups. Rationale: screen space less than 800 pixels makes the document section too small when tiling; the small screen mode suits applications that conform to 1990s looks-and-feels (e.g. Windows XP, Metal, Mac OS 9, KDE).

This view is the default and initial view on small screens.

**Menu** – with items such as File, Edit, View, Window, Options and Help that provide the functionality of the session section and assistance section of the wide-screen mode. There is no requirement that these items provide functionality available from the interaction sections of each document, providing accessibility affordances are available for these to support people with limited dexterity.

**Document section** – for selection and display of the document. In the centre.

**Status bar** – for display of warnings, progress and status. At the bottom.

In small screen mode, the menus are turned on by default and perform the actions performed by the session controls in full-screen mode. Typically, there will be only one document window. Small icons should be used in toolbars, etc.

**High contrast** – this mode is designed for partially sighted people, and features high contrast text, large print, keyboard-based interaction rather than mouse-based, and full advantage of the platform’s accessibility requirements. Where icons are used, they should be large and distinct. Rationale: accessibility is a requirement.

Other screen modes could include multiple document interface (MDI) mode, and tiny mode for PDAs.

## Document Section

The document section contains, from top to bottom

**Document selector** – a control for selecting and closing the current document from a list. For example, a tab control or gallery. At top of the section.

**View selector** – a control for selecting and closing the current view of the current document from a list. For example, a tab control. At the bottom of the section.

When changing view, the old view should be cached as an image. If the document model and other settings are unchanged (not *dirty*), the cached image should be displayed immediately. If the document model or settings have changed, the image should be shown obscured with an indeterminate progress bar running while the new view is established. Rationale: flipping between documents and views is common and sometimes the information sought was on the old page; faster response is important for users; views that are the result of lengthy functions should not be recreated unnecessarily.

Four standard views should be implemented, with others possible:

**Document view** – the normal editing form of the document. This is view that is open by default and whenever the document selector changes the current document.

**Print Preview view** - the print preview, which also include printing controls

**Settings view** – settings and options related to the current document

**Info view** – information about the file, size, permissions, metadata etc.

**Interaction section** - a set of controls for controlling and manipulating the view and document. The controls are application and view-specific, though the layout between each view may be similar. For example, a ribbon control, a tab bar, a gallery, an address bar, a breadcrumb bar. Under the document selector. Rationale: Making the interaction controls the children of the view removes the confusion of controls that are inappropriate for the current view or document being displayed.

**Document view** – the view of the document. For example, a browser pane, a dashboard, a tiled set of webcams, an editor with error messages section.

## Status Bar

The status bar has three modes, selected dynamically as required, either in alternation or in some packed arrangement:

**Progress mode** – a progress bar indicates the progress of some task in action

**Message mode** – a message is displayed for some period. The message typically will have some accompanying icon indicating its severity or source, and the font and background color and notification audio icon as appropriate. The severity types are INFO, TIP, CAUTION, WARNING, ERROR. Where multiple status messages are possible, the message with the higher severity should be displayed by priority.

The document view may have some visual indication of the subject of a message that relates to an overlaid arrow pointing from the message to the relevant part of the document. Provision should be made for the display of longer or formatted messages; for example a popup window or speech balloon in the Details section. Provision should be made for logging and displaying recent status items. Some errors may be blocking: these do not have a time-out but depend on the removal of a condition in the user interface. In general, messages to the status bar should be used in preference to pop-up windows such as alert boxes.

Messages can contain hypertext links to help information or the WWW.

**Status mode** – the bar contains various icons in various states to indicate the status of the current document and view, and of other processes run from the program. For example, an icon to indicate a document is currently being printed. The icons may be clickable to bring up further job-control controls. After the available messages have been displayed for the appropriate time, the status bar switches to status mode.

## Session Controls

Session controls include:

**Document button** – A control such as a large button which is the equivalent of the File menu in menu systems. It provides whatever functionality is not provided by the document selector (and any Window menu or Mac menus) for opening and closing files or documents, for workflow, etc. Note that printing should be a function in the Print Preview view interaction controls.

**Generic edit operations** – Controls such as buttons which invoke actions that can be expected to work in some way in any application and mode. For example, Save, Undo and Redo.

## Assistance Controls

Assistance controls include:

**Help button** – a control such a large Help button which is the equivalent of the Help menu in menu systems.

**Document window selector** – a control to select the window, where there are multiple document windows or an MDI screen mode.

**User interface settings** – controls to adjust aspects of the look and feel or other settings.

## Collection Section

The collection section displays information about the context of the current document. For example, integration into a content management system, as part of a collection, as part of a document set, as a favourite document, for recently used documents, or as the result of a search for tagged document (metadata).

This section may dynamically appear or be collapsable. For screen modes where a menu is available, selecting the Collection section may cause a different set of menus to be displayed and also deselect any objects in the Document and Details sections.

## Details Section

The details section displays such things as further information about the currently selected objects, attributes and properties, the results of searches, palettes, clipboards or task panes that provide sequences of tasks. Note that the interaction section provides the base for user interaction and commands; the details section is used more for affordances that rely on previous steps rather than being consistently available.

This section may dynamically appear or be collapsable. For screen modes where a menu is available, selecting the Details section should not cause a different set of menus to be displayed or change the currently selected objects in any sections.

## Other Controls

Doxology defines several additional controls

**Navigation box** – a set of controls, for example stop, pause, play, fast forward, rewind. Used in full screen mode.

**Title bar** – the title bar of the window will track and display the program the currently selected document, document window name (e.g. the nominal program name) and any type information that influences the major mode that the application will use to process the document. The title bar should not contain information about the view or application per se.