



Tech Info Library

OpenDoc: Overview (7/96)

Revised: 7/25/96
Security: Everyone

OpenDoc: Overview (7/96)

=====
Article Created: 1 August 1995
Article Reviewed/Updated: 25 July 1996

TOPIC -----

This article includes excerpts from "OpenDoc for Macintosh -- An Overview for Developers".

DISCUSSION -----

OpenDoc is a cross-platform software architecture that enables the development of component software. Component software is a term used to describe reusable, compatible building blocks of code that work seamlessly with applications--called "part editors"--from different vendors. OpenDoc consists of five functional layers:

- Compound Document Services
- Component Services
- Automation Services
- Object Management Services
- Interoperability Services

These technologies are being licensed to CI Labs, an organization established to provide open access to OpenDoc source code and specifications. CI Labs is responsible for adopting, licensing, validating, and promoting essential OpenDoc technologies. By providing reference source code for these foundation technologies, CI Labs facilitates the adoption of a common component software architecture across all major industry platforms. CI Labs also provides developers and system vendors with test suites and a conformance validation process, to ensure that software enabled by OpenDoc interoperates properly. OpenDoc will be available for Macintosh, Windows, OS/2, and UNIX systems.

OpenDoc Terminology
=====

Document

A collection of OpenDoc parts, assembled by a user or developer. A part becomes a document if dragged from its document to the desktop. A document becomes a part if dragged from the desktop into an open document.

Part

The fundamental building block of an OpenDoc document. This is the content that users see in their documents. The functionality in the associated part editor or part service allows the user to manipulate the part. Part viewers allow the user to view, but not edit, the part.

Container Application

A monolithic application that has been modified to support embedded OpenDoc part editors and part services.

Part Editor

A part editor displays a part's content, facilitates manipulation of the content, and provides a user interface for modifying that content. This user interface may include menus, controls, tool palettes, rulers, and other modes of interaction. (For example, a text editor is a part editor.)

Part Service

A part service provides "back-ground" functionality for a part, and provides the necessary user interface for manipulating that part's content. For instance, database access functionality would be a part service that could be added to an OpenDoc document. Its user interface could be a menu item or a database query screen.

Part Viewer

A part viewer offers a subset of a part editor's functionality; it allows users to display and print a part's content, but not to edit it. Viewers can be useful in document--sharing situations--for example, when the recipient of a document doesn't hold a license to some of the kinds of parts included in the document, or when the person sending the document doesn't want the recipient to alter it.

How OpenDoc Works

=====

By enabling users to work in a task-centric rather than an application-centric manner, OpenDoc makes it easier for them to create and edit content-rich documents. With OpenDoc, users can purchase or build a "container" document that includes all the functionality needed to complete a task. They no longer have to launch separate applications, worry about incompatible data types, and manually create links to update data. They will have the freedom to buy preassembled OpenDoc software solutions, just as they buy application suites today, or they can mix and match part editors from a variety of vendors to suit their specific

needs. No matter where the part editors are obtained, the user will be able to focus on a single solution that contains all needed functionality.

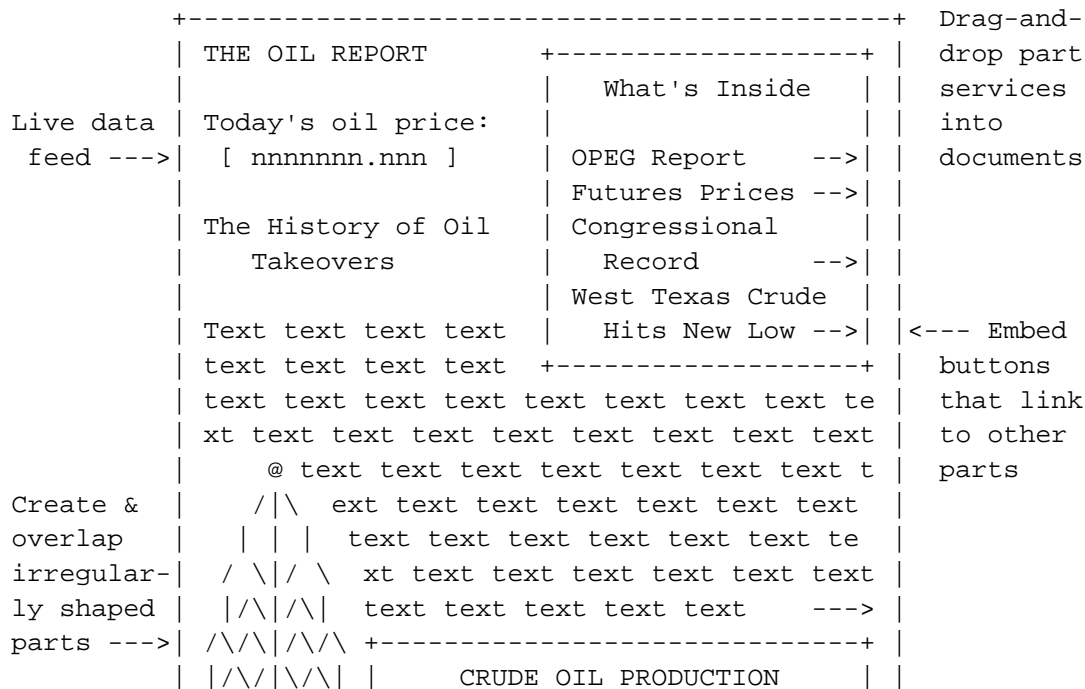
The OpenDoc model also improves upon the current drag-and-drop method of manipulating data. For example, a user can drag a desktop file icon into a document window, and it will automatically open and embed itself within the document. This contrasts sharply with today's way of doing things: opening one application, selecting an item to be edited and moved, copying it onto the Macintosh Clipboard, opening a second application, and then pasting the results into the window of this second application.

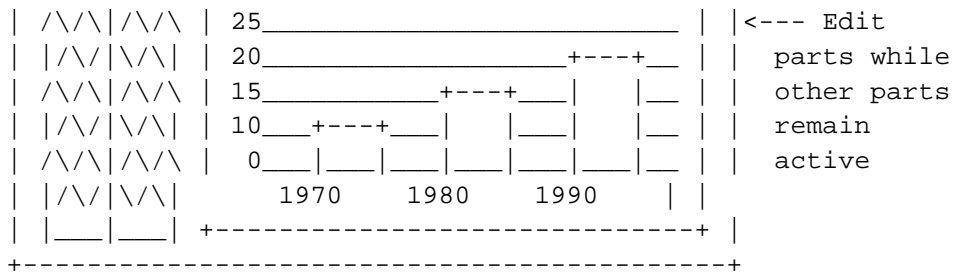
OpenDoc also supports irregularly shaped, overlapping parts, and the ability to have a variety of different parts active concurrently. What's more, between OpenDoc parts, data can be "linked" together. A link is a permanent reference from one part to another. When data in the first part changes, the link notifies the second part to change. Suppose, for example, a company creates an investment portfolio pie chart with a link to stock market data. Each time stock prices change, the chart adjusts accordingly. Before creating software enabled by OpenDoc, programmers will need to become familiar with some new naming conventions.

The Anatomy of an OpenDoc Document
=====

The fictional document below illustrates several unique features of OpenDoc. This oil industry report uses many OpenDoc part editors and services: a text editor, a graphics editor, a page layout editor, live buttons, a grammar checker, and a window that displays live oil commodity prices from the New York Mercantile Exchange:

Begin_Graphic





End_Graphic

You do not need a separate application to work on each part -- just click on an area to edit it. As you click on another object, its own menu bar and tool palette appear. Parts like the live oil market commodity quotes shown in the document stay active even while others are being edited.

The ability of software enabled by OpenDoc to have multiple parts concurrently active allows the author of this newsletter to edit the crude oil production chart while the current oil prices roll across the top of the page. The editor can wrap text around the irregularly shaped oil derrick graphic part. Live buttons within the "What's Inside" field allow readers to click on a button to call up a feature article. And by dragging the "Grammar Checker" icon into the document, the part service can check grammar throughout the entire newsletter.

This article was published in the "Information Alley":
Volume II, Issue 7, Page 9

Article Change History:

- 25 Jul 1996 - Reviewed for accuracy.
- 04 Aug 1995 - Changed formatting, changed parent folder.

Copyright 1995-96, Apple Computer, Inc.

Tech Info Library Article Number:18262