The Living Document
OpenDoc White Paper

Introduction

This paper intends to lay the strategic groundwork concerning the future of OpenDoc in conjunction with WordPerfect. There is a revolution on the horizon that will change the very manner of our work and refocus our energies toward content and away from the monolithic applications of today. Technologies such as OpenDoc and OLE 2.0 promise changes that are both creative and collaborative in nature, forcing strategic decisions upon software vendors that can be both culturally and financially dramatic.

This paper will include the following:

Convergence towards the document
History and definition of the document
OpenDoc...What is it?
OLE 2.0 vs OpenDoc
Strategic implications for WordPerfect

Convergence Toward the Document

Every major software application (spreadsheet, presentation graphics, word processor) has a real life metaphor that gives the user perspective and definition of what the application can do. For example, the real life metaphor for document creation originally included pen, paper and eraser. Over time, typewriters evolved to automate some of the ordinary tasks that were involved in creating the document. Software vendors analyzed the typewriter metaphor and the processes behind the metaphor and invented the word processor. These vendors (WordPerfect, WordStar, Microsoft, Lotus...) have provided the world with an accurate representation of an existing metaphor and its processes. Since the inception of the word processor, vendors have set out to refine functional changes based upon the above identified processes and metaphor.

There are two problems that become inherent with this process. First, these software vendors have not re-examined the actual metaphor since its inception.

The assumption is that although functional changes have taken place over time, metaphorical changes have not (false assumption).

Is it correct to assume that during the past 13 years, people have not changed the way they work? Of course not; automation, organizational structures and behavior, and the evolution of the workgroup have all changed how people perform their daily tasks, changing the very processes upon which the current metaphors are

based. Since the processes have changed, metaphorical changes have certainly occurred. In short, the typewriter metaphor no longer holds true in the description of tomorrow's word processor.

Secondly, the document itself has changed over the past decade. Throughout history, the document was used to communicate static information that had already been gathered and put to rest. The document usually consisted of one input, one output and one author.

Input is defined as the number of information sources represented in the document.
Output is defined as the number of ways the document could be read.
Author is defined as one who originates, creates or changes the document.

The second issue hinges upon the history of the document itself and its evolution into a compound document. In pre-historic times, man wrote on stone to portray history or for story telling purposes. This device was the best available for storage yet retrieval was unrealistically limited. The medium (stone) was very limiting in scope and could only be used once and typically by one individual. The model of communication during this time was fairly limited to one input and one output which proved very difficult to share.

Handwritten manuscripts (on paper or a form of paper) followed, but only offered a minor increase in distribution capabilities. Input was very limited and was largely viewed as static in time. Output was restricted by the time and energies of the scribe(s). As with stone, storage and archival was limited to the physical document itself, severely restricting the flow of information.

The printing press proved to be an invention that changed all the facets of the document and communication as a whole. The printing press allowed for distribution capacities (output) that were previously unattainable, thus improving the speed with which information was disseminated. The printing press made it plausible for multiple authors to be published in one paper, document or periodical, thus increasing the input in size but not scope. The printing press however, removed the writing process from the document, which tends to make it static in nature upon printing.

The typewriter enabled the end-user to personally create a finished document. The authoring process is no longer removed from the document. It is true that, with the typewriter, the document could conceivably have multiple inputs but the inputs are static in nature, meaning the information being displayed isn't dynamic, it's inflexible to change. The typewriter (like handwritten manuscript) limited output to the time and energies of the typist. However, combined with the modern copier, the

typewriter became a powerful device for static document creation and multiple copy distribution.

It is the typewriter that the modern-day word processor mimics most accurately. The documents created in a word processor improved slightly in two areas: input and output. Using the word processor, today's users can pull information from other documents, spreadsheets, databases or presentations using the clipboard function on both Windows and Macintosh based machines. Thus, input becomes slightly more dynamic and easier to implement than what was achievable at the time of the handwritten manuscript. Also, by nature of the medium, the word processor made such input less static. Content could be repeatedly changed to better reflect the dynamic characteristics of information. Improvements to output were gained when the word processor was combined with the network printer and e-mail. The printer allowed for printed reproduction of the document e-mail allowed for mass, instantaneous distribution. However, as with typewriters, multi-author collaboration is laborious and simultaneous collaboration is non-existent.

Today's Document

Today's document, as defined by the above events, is the victim of evolution. Over time, the document has grown to represent the following:

Static information
Limited authorship
Widespread output capabilities

Information has played a static role in each of the events described above. The underlying assumption is that the author knows exactly what they want to present and that the information isn't going to change. While this may work for periodic publications, it has little merit in today's business world. Business plans, financial results, productivity reports, marketing strategies, advertising plans...all are examples of documents that not only change periodically but can change almost daily.

Is it inconceivable that a document can be continually monitored and updated with the most current information available?

A company that bases its future on static information will become history. Information leads to knowledge and knowledge breeds wisdom. Wisdom is used to make profitable business decisions based upon educated risks.

Today's document is commonly written by one individual. This is not because it is undesirable for others to participate in the creation process, it is simply because of the technical difficulty of collaborative writing. The document is a natural metaphor for the collection of ideas and information that may

come from a group of individuals or places (infobases, databases etc). However, today's word processors assume there is only one author/owner and are designed to take advantage of and propagate that very paradigm.

Name an organization where one person controls all the information? Is it inconceivable that more than one person can add value to the creation of a document?

Lastly, output of the document has matured the most over time. It is now possible to distribute multiple copies of the document electronically via e-mail as well as with the new improved electronic publishing tools. Printing, despite the improvements in speed and quality, hasn't really changed since the printing press innovations. In short, throughout time, the major improvements in the "document" have focused on output...distribution of the finished product.

Is it not time to bring similar breakthroughs to the creative aspects (input and authorship) of the document?

The Living Document

The living document is a document that makes exponential improvements on the traditional document in two areas; input and authorship. Input as defined above includes information parts such as text, graphics, spreadsheets, database data, video, sound and any other conceivable form of media. Input in the past has proven to be very static or stagnant in its content. The living document allows for dynamic input that changes or evolves as the information becomes available. The living document is a breathing document that is never put to rest:it is a version oriented document linked with supplemental information. However, the nature of some types of documents requires a stable, never changing document which is also possible.

Consequently, the living document is no longer confined to one application for creation. In fact, a word processor, spreadsheet, database, or presentations package could all be in charge of document creation. The applications will begin to represent document shells that are responsible for the containership of content found in the object parts. Technologies that allow for this include OLE 2.0 and to a greater extent OpenDoc. Intuitively hyperlinking ideas and concepts with alternative informational sources provides for a infinitely expanding knowledge base which resides in a "dataspace".

How to use it

The changes that are outlined are truly revolutionary and are cause for concern for ISVs and end-users alike. However, the implementation of such changes will not disrupt how documents are currently created or distributed. Initially, OpenDoc will be used to encapsulate different data structures in the document,

provides background data which is pertinent to the decision at hand. Information management allows the decision maker to more accurately wade through the dataspace/infobase.

Is it unimaginable to drink from a fire hose?

OpenDoc...What is it?

OpenDoc is a set of technologies being co-developed by Apple, WordPerfect, Borland, IBM, Novell. The goal of OpenDoc is to make the user's computing experience as easy and productive as possible. People are using computers for more complex tasks that include multiple applications and working together on projects. This realization identifies a shift to shared, collaborative computing resources.

The OpenDoc architecture reduces the complexity of computing while simultaneously supporting the flexible and highly customizable applications of the future. OpenDoc is an open architecture, designed to integrate software and enable sharing across multiple computer platforms-providing users with a new level of computing power, flexibility and ease of use. The architecture is an object-based framework for developing applications that are fully integrated and interoperable across platforms and distributed networks, giving the user the ability to capitalize on the true power of the document as a means of communicating.

Benefits to the End-user and to the Developer

The OpenDoc architecture benefits the end-user and developer in different ways. The benefits for the end-user are as follows:

Easy creation of compound documents. OpenDoc is designed to handle current and future types of media. Users can place any kind of media into an OpenDoc document using the familiar cut-and-paste or drag and drop manipulation, thus creating a compound document with "rich" content.

Editing "in place." With OpenDoc, users can edit any type of content within a single document, without having launch the native application. This allows the user to focus on the document and not banal formatting. In addition, this technique saves system resources from unnecessary allocation.

Powerful document management. Rather than manually assembling the various pieces of a document, users can let an OpenDoc document hold all of them. This reduces the task of managing files, and facilitates document exchange and updating. As documents are edited, changes are tracked through drafts, ensuring greater data integrity and allowing users to work on shared documents without content loss from version to version.

Cross-platform support. Because OpenDoc is designed to offer

full interoperability between platforms, OpenDoc users will be able to share and interact with complex documents, regardless of differences in software or hardware, or which platform the document resides on.

Consistency of operation. Because users can specify preferred part editors, they need learn only one way to edit each type of data-for example, using the dame text editor for word processing, entering spreadsheet data, or labeling diagrams.

Uniformity of interface. OpenDoc defines a consistent user interface for embedding and manipulating all kinds of media in documents.

Scalability. The OpenDoc human interface addresses a wide range of users, from novices to experts. No class of user has to understand the additional functionality typically used at the next level-novices can create compound documents easily, while experts can experience nearly unlimited potential.

"Plug-and-play" solutions. With OpenDoc, vendors will be able to assemble collections of parts into solution sets that target specific tasks or work styles. These parts can be shared across documents, news briefs and even networks.

Benefits to the developer include:

Faster more efficient development. Software developers can reuse already developed parts, eliminating the need to start from scratch with each development effort. This ability to reuse existing parts also means that developers need spend less time on parts that are peripheral to their main area of expertise.

Diminished costs of software development. The fact that parts are smaller than applications makes them both quicker and cheaper to write, which reduces the penalties for failure.

Industry-standard object management. Because parts can use a CORBA (Common Object Request Broker Architecture)-compliant object mechanism, they can be written in a wide range of programming languages and development will be supported by many tool vendors. This mechanism gives developers high performance coupled with great flexibility in the use of "plug-and-play" objects.

OpenDoc Technology

The OpenDoc architecture enables the creation of compound, collaborative documents which are interoperable across platforms, other documents and other compound document architectures including OLE 2.0. The architecture is open and source code will be available to vendors who want to support the technology in their products.

The major concepts explored in the architecture include the following:

Documents
Parts

Documents

The document has been described and defined in detail above. In short, OpenDoc propagates fundamental changes in the document. With an OpenDoc implementation, a document is no longer a single application but is instead composed of smaller blocks of content. This content can be interactively edited, altered, updated and contained in as content parts.

The document will act as a shell that houses many parts of content. This shell can take the form of many of today's applications including a word processor, database, spreadsheet or graphics application. In short, nearly all applications will have the ability to create a document.

The definitive applications of today will blend together to form a document in the future.

Parts

Parts can be considered the building blocks in OpenDoc and in the ensuing document structure. These "parts" will replace the monolithic applications of today with smaller units of content dynamically bound with related functionality. OpenDoc parts may be viewed in four ways: content containers, part editors, frames and part handlers.

Content containers can be described as data blocs, each containing information from a myriad of applications including graphics, spreadsheets, databases and text from word processors. Examples of data include pictures, spreadsheet cells, database queries, digitized sound and video etc. The particular type of data that each part contains is known as the part's intrinsic content and is defined by the developer/end-user.

The "mother" of all parts is known as the root part. Every single document has a root part in which all other parts are embedded. The relationship between these parts describe the structure and function of the document. A part's contents include all the data that part needs to implement its specific functionality as well as the information needed to visually display the part object. Additionally, the parts developer determines whether to support the capacity to contain other parts, however, a key characteristic of OpenDoc is that if a part can contain one type of part, it can contain all types of parts.

Part Editors

Part editors are the independent programs that manipulate and display a particular kind of content. OpenDoc part editors serve as the building blocks for solution building as well as document creation. Conceivably these editors would consist of text, graphic, mathematical, database editors as well as others. These part editors will allow for plug and play document creation. In addition, OpenDoc will allow these part editors to be dispersed across a network allowing disparate authors the ability to contribute to the creation of a document (providing the proper rights have been given). OpenDoc parts will allow developers to create new applications in a manner similar to that of constructing a document template in today's world.

Frames

Parts can also be viewed as the boundaries at which one kind of content in a document ends and another begins. A key element of the concept of parts is that each part of a document has its own content modelithe model of objects and operations that is presented to the user. The content model changes at the frame between parts. For example, a compound document could have its root part as a WordPerfect text part editor that provides the template for the document. Included in the document is a pie chart that has been generated from a charting editor. This pie chart is linked back to a spreadsheet editor that contains the raw data and can be updated on demand. The following figure shows this format in a document.

Part Handlers

When a part is being displayed or edited, a part handler is invoked to perform those tasks.
A part handler is responsible for the following things:

Displaying the part both on the screen and for printing purposes.
Editing the part. The part handler must accept events and change the state of the part so that the user can edit and script the part.
Storage management (both persistent and runtime) for the part. The part handler must be able to read the part from a storage device into main memory, manage the runtime storage associated with the part, and write the part back out to the storage device.

Part handlers are dynamically linked into the runtime world of the document, based on the part types that appear in the document. Because any sort of part may appear in a document, the part handler must have the ability to respond to each part, hence the dynamic link to the document. This will provide for a smooth and consistent user experience.

In addition, part handlers can be divided into two types: editors and viewers. When a part is activated an available part editor becomes activated as well, giving the user access to the

tools needed to change or modify the content. Conversely, a viewer provides the user with the ability to display and print the part but does not allow for editing. Viewers primary uses include when the recipient of a document does not have access to a license of a part used in the document or when the person sending the document does not want the recipient to alter it.

Both editors and viewers can interpret the contents of the part and display that content for the user. The idea is that eventually, developers will create both kinds of handler for every part. The editor would be sold, but the viewer would be freely distributed to enable and encourage document interchange.

OpenDoc Versus OLE 2.0

OpenDoc is the result of collaborative efforts from Apple, Novell, WordPerfect, IBM, Borland and other ISVs and industry consortiums. The advantages that OpenDoc has over OLE 2.0 are substantial and include the following:

OpenDoc is an "Open" architecture
The architecture is network ready and intelligent
OpenDoc handles odd shaped parts
OpenDoc allows for multiple moving parts per page

OpenDoc is truly an open architecture. OpenDoc is a cross platform architecture that abides to industry standards and consortiums. Component Integrated Laboratory (CIL is the industry consortium representing the global interests of OpenDoc) is committed to develop OpenDoc for the Macintosh, Windows, OS/2 and UNIX platforms with interoperability being promised for the Taligent platform. Conversely, Microsoft has committed to delivering OLE 2.0 on the Windows platform and is giving lip service to the Macintosh platform. Secondly, the source code for this architecture will be made available to ISVs for implementation purposes. Also, OpenDoc is compliant with the Object Management Group's (OMG) Common Object Request Broker Architecture (CORBA) while OLE 2.0 and Microsoft's Cairo (the company's future object oriented operating system) are not. CIL is also working closely with the Open Software Foundation (OSF), and X Consortium to assure standards compliancy. Lastly, OpenDoc will be interoperatable with OLE 2.0.

The architecture is network ready, allowing for parts to be shipped, transferred and requested from disparate networks and servers. OLE 2.0 doesn't allow for such network intricacies, hence OLE 2.0 links will be severed upon shipping across disparate networks or servers. In short, OLE 2.0 works well in a single server, single network environment but has no capacity to perform in a multi-server, multi-networked environment. Through IBM's System Object Model and the CORBA industry standard, OpenDoc will provide access to distributed object services.

In the OLE 2.0 model, parts can only take the shape of a

rectangle or square. In the real world, parts should be able to assume the best possible shape to adequately display the information on hand. OpenDoc allows for odd shaped parts to be portrayed and handled within a document. Also, in the OLE 2.0 model, documents are forced to comply with one moving part per page. This means that in any one page of a document, OLE 2.0 is only capable of supporting one video, one presentation or one animation. OpenDoc eliminates this limitation and allows for multiple moving parts per page per document. In the future, information will take many forms and the OpenDoc architecture is an attempt to bring the future to the industry...today.

Strategic Implications for WordPerfect

"Your most dangerous competitors are those that are most like you." (Harvard Business Review, 1991)

The decision to write to the OpenDoc specification and design to the compound document paradigm should not be taken lightly. Defining WordPerfect's role in this arena should be a strategy that includes corporate vision (both technology and market vision) and product strategy. A strategy is the summation of the following three elements:

Objectives (not product or technology related)
Vision (product or technology focus)
Message (the communication to the general public)

All strategies start with a list of objectives which one hopes to accomplish. These objectives must be clear in meaning but may be ambiguous when related to a product or technology. Examples of objectives include: protect the existing installed base or change the image of a company from a manufacturer to a distributor. Notice neither objective focused on a specific product or technology.

Product or technology focus takes place at the vision stage. A vision provides an immediate road map of executables that need to be accomplished to help meet the objectives. Examples of vision statements include: refer to object technologies, focus on company's strength in distribution, or focus on the competitors weak development capabilities.

The message is the content used to communicate the vision and objectives to the audience of choice. The message may or may not contain parts of the objectives and vision, and deals with perspective as reality.

By adopting the OpenDoc architecture, the following elements will impact WordPerfect's tactical and strategic decisions in the future:

OpenDoc alleviates dependency on Microsoft for compound document structure for WordPerfect and the industry

Competitive and strategic advantage
Aids in obsolescence
Organizational structure must change

Alleviating One Source Dependency

By incorporating the OpenDoc architecture, the developer is giving the end-user a choice of interoperable compound document structures, thus relieving the one source dependency for future technologies that worries many IS shops. WordPerfect, like many ISVs, are held at the mercy of Microsoft for many operating system services. By fully supporting the OpenDoc initiative, WordPerfect will no longer be held captive to Microsoft and its agenda of propagating its game plan.

Competitive and Strategic Advantage

As outlined above, the OpenDoc architecture provides the developer, end-user with various competitive advantages both tactical and strategic. Support for this architecture will supply with WordPerfect with an overall corporate vision of the future computing environment, and the technology that will make it happen.

Aids in Obsolescence

Products become obsolete at an increasing rate in the technology marketplace. Managing one's product obsolescence is crucial to the success of the product and the company. In the past, Microsoft has excelled at this and is currently in the process of doing it again with Windows, NT and Cairo. The theory is that in an industry that has very short version life cycles (version 1.0 vs version 2.0), management of that version cycle and the subsequent version cycle is crucial for the products success. For example, WordPerfect for Windows 5.2 had a version cycle of only 11 months. Management of the version, how and when to discard it, as well as the management of the upgrade are issues that don't just happen on a development schedule. The word processor is expected to fade from prominence sometime in the next three years. OLE 2.0 and Cairo will secure this prophecy and enable Microsoft to manage the process.

In short, the OpenDoc technology helps WordPerfect make the word processor obsolete. This technology allows WordPerfect to make the decisions as to when an how we choose to change word processing instead of playing by someone else's rules.

Organizational Structure

Technology will not drive cultural change.

As products become smaller and more modular, business models will have to adjust. the typical business model of major software vendors today is composed of an organization based on a product

model and dotted line to specific functions (marketing, development, advertising, and PR). During the next five years, the industry will see two compound document architectures evolve (OpenDocs and OLE 2.0), two object-oriented operating systems introduced (Cairo and PowerOpen), and the applications and categories losing their tight definitions of today.

The business model of today enables the vendor to sell today's technology as tomorrow's applications, all packaged in new shrink wrap. Part of the problem during the 1980s was that the personal computer revolution was so young that users had their hands full learning <<CTR-ALT-DEL>> and other codes. The market was so young that there was not time for innovation...only time to market. It simply would not be successful to employ a business model that exploited innovation and forced the end users to change and adjust constantly.

Business models of the future will be focused on delivering functionality innovations and no simply propagating the current product line. In fact, the current product lines will slowly dissolve into functionality sets. The organization of the future will no longer be organized around products but around functional sets of technology. For example, every word processor has a director of product marketing; tomorrow there will be a director of document rendering.

This modular approach to the organization structure maps exceedingly well to the inevitable market transformation. This organizational structure is vastly different than the one that exists today at WordPerfect and forces change upon many groups and individuals. By adopting OpenDoc, WordPerfect must be aware that organizational changes are on the horizon and a plan must be initiated to manage the change ahead.