

From: Office Product Unit
Subject: Office Product Unit 1995 Three Year Plan
Date: March 1995
Key Authors: andrewk chrisgr jimco jonde peggyst rwolf samh stevebr tonyw
Contact: stevesi

1. Introduction

This memo outlines the technology plans for the next three releases of the Office product: Office 96, Office 97, and Office 98, and perhaps beyond. Office 96 is well underway and this memo will describe the Office feature set that will ship in June of 1996, for more information please refer to the Office 96 product specification.¹ We describe below the features that are common across the Office applications being designed and implemented by the Office Product Unit. This document is not a complete description of the Office product and should be considered in conjunction with the three year plans and specifications for Word, Excel, PowerPoint, Access, and Ren.

Office 97 and Office 98 are described as a set of features that will evolve into two products. The Office 97 release will be a *polish and preview* release based on Office 96. Our goal in Office 97 is to provide visible features that will drive upgrade to this release and polish off the loose ends from the previous major Office 96 release. We plan on previewing some portions of the Office 98 release in the Office 97 release. We will not invest in major architectural changes for the Office 97 release. Using Office 95 as a guide, we expect the Office 97 release, as "12" release in the 12/24 methodology, to use approximately 15-20% of the total development resources, and the Office 98 release to be a major release using 80-85% of the team. At this point in our planning we do not have the information required to commit to the specific contents of these subsequent two releases.

Office is currently in the midst of Office 96 and as such this memo represents a directional statement more than a product plan. The primary goal of this memo is to consolidate our thoughts on where the 1997 and 1998 (primarily 1998) release of Office will head. There are portions that are vague and lacking in detail and merely represent our intent to investigate the area for possible features. Portions of this document just serve to raise important issues and problems that we wish to consider, but do not have proposed solutions. We are using the three year plan process to document the ideas that we currently have for beyond Office 96, so in that spirit this document invites discussion and debate.

2. Mission Statement

The role of the Office Product unit is:

To make Microsoft Office the most popular suite of desktop productivity tools in the world by

- ◆ designing and developing all core features
- ◆ innovating in cross-application technology and user-interface
- ◆ leveraging strategic Microsoft technologies
- ◆ leading the definition of product and marketing strategy

¹ Contact AndrewK

MS6 5007098
CONFIDENTIAL

Microsoft Confidential

-1-

FL AG 0043480
CONFIDENTIAL

March 1995

HIGHLY CONFIDENTIAL

Office Product Unit 1995 Three Year Plan

We will achieve this mission by evolving the Office product across several major axes. This broadly defines the major architecture areas of Office. Our intent is to use these axes as broadly defining the work of the Office product unit.

2.1 User Interface and IntelliSense

Mission: Office applications will be the simplest, most approachable, and most fun productivity applications to use. Office applications *just do the right thing* all the time.

We use the IntelliSense umbrella to include features that make Office the easiest and most fun to use suite of productivity applications on the market. In this category we also include our enhancements to basic user-interface elements.

We will continue to evolve user-interface techniques and underlying technologies to make features in Office either more accessible or accessible for the first time to both new and upgrade users. In Office 96 we will take a big step with the introduction of the social assistant as a mechanism for accessing user-assistance (e.g. help, tips, alerts), and integrating Wizards. We will refine this in the Office 97 release by introducing more richness to the assistant's interactions with the user. In terms of user interface, we will be creating unified menus and toolbars which will lower the barrier to end-user customization and reduce the number of unique user-interface elements for the user. We will also enhance context menus with graphical icons and context indicators. For the 97 release we will look more closely at the use of sound, animation, color palette, and alternate input devices which would make some features more accessible and easy to use (for example the tablet or mouse with thumb gauge).

Document management will be a key feature in accessibility and ease of use since Office 95 represents the first really accessible (i.e. transparent) use of a document store based on the file system. We will integrate this into Ren further in Office 96. Progressing through Office 98 we will make document management more full-featured and leverage the key operating system storage technologies.

The role of IntelliSense features will be of primary importance to the Office product because of their visibility and the excitement that these features generate. We are challenged by our competition since these features are often easy to copy, though hard to duplicate. By leveraging the work in Consumer and AT we are working to gain a longer term technology based lead.

2.2 Communicating Between Users and Workgroups

Mission: Office applications are the first choice for groups of people that need to work together on the authoring of documents and sharing of information. Office applications will be the first to leverage the expanding communication infrastructure.

Leveraging the significant messaging infrastructures is a key component of our workgroup strategy. We will start with Office 95 with WordMail, which uses the DocObject technology developed for Binders, and evolving to rich views on FAT and document management in Office 96. Office 96 introduces Ren, which is a sophisticated viewer for MAPI based stores as well as a front end for document management. Our applications will be integrated with Ren by supporting journaling of key user events along the lines of *Project X*.

In the past we have had trouble arriving at a definition of workgroup features within the Office product. Generally we have included any feature that involves the shared editing, viewing, or markup of information. With Office 95 we have a clearer understanding of where we want to take features in this category, such as the slide show conferencing in PowerPoint and multi-user workbooks previewing in Excel 7. We will evolve these document conferencing scenarios in Office 97 and 98 by leveraging the more general document/application sharing infrastructure acquired by systems. Building on the Escher drawing layer, applications will add support for annotations in Office 96.

MS6 5007099
CONFIDENTIAL

Microsoft Confidential

-2-

FL AG 0043481
CONFIDENTIAL

March 1995

HIGHLY CONFIDENTIAL

We will be enhancing the workgroup markup and viewing of documents across all of the Office applications. Office Binders will evolve to be a more strategic part of this infrastructure by providing a rich hierarchical view of the contents of a group of related documents. We will build on the multi-user reconciliation support in Office 95 Binders in the next releases. We will also improve OLE integration, with additions such as object pagination.

2.3 Business Process Automation (BPA)

Mission: Office documents will be the preferred mechanism for viewing, analyzing, and manipulating information in a business environment. Through the Office application object models and VBA, Office will be the most flexible, customizable, and best performing suite of productivity applications.

A traditional strength of Office applications has been the built in customization and programmability. We will continue to advance the state of the art and to invest heavily in these features. There are two key approaches we will take.

Near term we will invest in VBA inclusion in each application. This means we will improve the development environment and remove it from the applications so as to more closely align with our development tools strategy. Programming an Office solution will be a subset of the full client-server development tools from the Developer Division. Office will also coordinate a consistent object model across the applications.

Long term we will be investing in making Office applications better steps in the business cycle. There are a number of hard technical issues to deal with, but in terms of features the top of the list includes cross application macro recording, which is a current competitive deficit, and some form of process diagramming/visual programming currently under investigation.

BPA also includes the issues associated with configuration and distribution of Office applications and documents. We will be enhancing setup to make it more suitable to a corporate customization environment, including features such as run from CD and components on demand. Office applications will be the best network environment citizens by respecting things like the directory service and registry storage of profiles.

Longer term we will gain advantages in BPA by leveraging a consistent document architecture across the Office applications. This will enable both the Operating System and tools to generically manipulate certain aspects of Office documents. We will also be working towards a better template model so the distribution of Office documents with code behind them can be made simpler.

2.4 Leveraging Key Operating System Functionality

Mission: Office will be the best client of key Microsoft OS functionality and as a customer of BackOffice, Office will exploit the advantages of our platform synergy.

There are a number of systems features on the horizon that are going to be very important for us to leverage and adopt. The real key will be for us to drive the feature set of these as much as possible as early as possible. We are currently working very closely with the post-Windows 95 planning in order to maximize the possible synergy. Over the next months we will work as closely with the Windows NT team and BackOffice to drive this cooperation as well. A summary of BackOffice synergy issues is given on page 50. The key systems technologies include the following.

- Exchange will be the company's answer to a number of important competitive issues (higher end document management and Notes). We will do significant work to exploit this. We will have several features in Office 95 for leveraging Exchange. In Office 96 Ren will serve as the advanced front end to Exchange.

- Forms³ will serve as the basis for custom dialogs in applications in Office 96. This also includes DataDocs, which will be live reports on structured data available with Access. We will also be looking at Forms³ as a way of improving our end-user forms story for Ren, viz Notes competition.
- OFS will be a key technology for the document management group. In order for us to exploit it more directly in our applications systems will need to have an easily upgradeable OFS for Windows 95 machines. It will also be critical that we have a consistent and credible API for accessing both MAPI-based and OFS stores in this time frame.
- OLE 2.x and 3.0 continue to be important to applications. OLE 2.x needs to address primarily performance concerns. In terms of in-place editing and compound files, size and boot time are most critical. For BPA we need OLE to address the cross process performance issues. OLE 3 will introduce transparent objects and irregular shaped objects, though the timing of this needs to be worked out
- Communication infrastructure is being acquired and developed in PSD and this will be an integral part of the conferencing strategy for future releases.
- Hardware devices such as the new tablet or mouse will also be areas we will leverage, in the 1997 and 1998 releases.
- New user interface infrastructure such as speech, 3-D APIs and improved pen support will be leveraged, hopefully with adequate support from the system. We will be investigating these over the next year to determine the viability for Office applications and will work with systems during this time

2.5 Content and On-line

Mission: Office will be the preferred platform for authoring, viewing, and distributing information. Office will be the cornerstone in *corporate publishing* scenarios and in general distribution of document-based information on all on-line services.

There's been a lot of discussion about an impending "sea change" regarding the computer as a communication and information appliance. Long term, information will be the driving force behind software usage (we define information as not just raw data, but also analysis, expertise, consulting, and any type of knowledge transfer). The risk for Office is that consumers will trade better viewers for better access to information. For instance, there is a large volume of information being put into Notes (everything from brainstorm discussions and corporate publishing to newswire feeds) despite its inferior document editing, viewing, and printing capabilities.

However, people won't put their information into Office formats unless there is a very good reason since it's clearly not the least common denominator (i.e., most spreadsheets including Excel can read WK1 format, so why not just put it in that format.) To create the incentive, we need to embark on a strategy that includes:

- **Easy on-line publishing.** Chances are the user is already authoring using our tools. Publishing on-line should require little advance planning.
- **Links everywhere.** We plan on incorporating a general link mechanism (based on OLE) across all of Office applications. This will allow a user to link arbitrary Office documents together, along with Operating System objects, in a web of content.
- **Great viewers for free.** We need to remove the concern over a recipient's ability to read a document. Most people will want to pay more to get authoring and advanced viewing capability.

- **More content in Office format.** The deciding factor could easily be content. Since Office is the market leader, more information is being authored in Office format than any other format. However, Office is not yet a center for distributed information, consulting, support, or reference.

Implicit in our strategy is the fact that we believe that as the number of users increases so does the number of content authors. Although there will be a large amount of professional content, the wide-spread availability of easy to use and accessible authoring tools will make everyone an author. The Web today is almost entirely a read-only transport, though with some write once forms and very limited pseudo-interactivity. There is no reason for one to assume that content is either static or read-only. By providing integrated on-line viewers and authoring tools, users will seamlessly switch between readers and contributors.

2.6 Document Architecture

Mission: Office Development will design and implement the leading technologies for shared application and document architecture.

Starting with Office 96 the groundwork for implementing a shared document and application architecture will begin. The best way to think of the work being done is as a common application framework for Office applications, or a "base" application. With each subsequent release of Office the amount of work done in this base application will expand. The goal of doing this work is to enable sharing of code (efficiency of development and performance), a higher level of architectural investment (longer term investment and harder to clone), as well as an improved level of integration and synergy across our Office applications.

2.7 Category Innovation

Mission: Each Office application should be the best in its category. As a suite, Office will be the best integrated and most consistent.

Underlying each of these areas is a very conscious effort to track what Lotus, WordPerfect/Novell, and Borland are doing in their individual applications, in addition to the suite category. Each release will include significant time to respond directly to competitive issues. The priority of this will always be high since we always need to insure that we win reviews, however, as our competitors become increasingly desperate we need to be wary of defocusing and more importantly alienating our installed base where upgrades are critical.

3. Architectural Release Themes

The Office 96, 97, and 99 releases will have the following major technical/architectural themes:

Office Product Unit 1995 Three Year Plan

Product	Themes
Office 96	Sharing Office Infrastructure 1.0 Develop first shared infrastructure technologies
Office 97	Polish and Preview Polish Office 96 infrastructure, Office Infrastructure 1.1 Preview Office 98 technologies Sustain user interest and excitement
Office 98	Document Architecture Office Infrastructure 2.0 Design new infrastructures for sharing, communication, and user-interface techniques

Table 1. Office Product Unit architectural release themes..

4. Release Timeline

Our schedule for post Office 96 is based on the 12/24 methodology. The following table summarizes the release timeline as best we know it today.

Product	Time	Comments
Office 95	June 1995	15% of resources, tied to Windows 95
Office 96	June 1996	85% of resources
Office 96/Mac	Office 96 + 30 days	Relies on simultaneous development. See Macintosh plan.
Office 97	Office 96 + 12 months	"12" release
Office 97/Mac	Office 97 + 30 days	Very dependent upon marketplace and Macintosh System
Office 98	Office 96 + 24 months	"24" release, tied to major Windows release

Table 2. Timeline of Office releases over the next 3 years.

5. Windows Operating System and Hardware Requirements

The most obvious assumption we are making in our post Office 95 plans is that systems will align the Win32 API, including the documented and public additions for Windows 95, on both the client and the server. It is critical to our planning and ability to deliver timely release that we avoid separate releases for Windows NT. If it becomes necessary, we will support Windows NT as a subset of our Windows 9x applications.

We are actively pursuing joint requirements analysis and design of features with both the PSD and BSD divisions. In particular we are working to minimize duplicate design efforts and work towards our applications being first class users of system services, rather than duplicate them. In order for this to succeed we will need to arrive at designs that meet the feature richness required by the applications as well as the API requirements of the operating system.

PSD areas we are focusing on include both implementation (memory use, performance, use of thread and processes, OLE) as well as user-interface features (rich views on FAT, command bars, social interface, SDF). Ideally we would like to arrive at a subset/superset relationship for functionality in the system, where the application ships first with the feature followed by the system. Subsequent system releases then inherit the previous application level functionality.

MS6 5007103
CONFIDENTIAL

Microsoft Confidential

-6-

FL AG 0043485
CONFIDENTIAL

March 1995

HIGHLY CONFIDENTIAL

Office Product Unit 1995 Three Year Plan

At the BSD level we are focusing on SMS, Exchange, OFS, and DS service integration. Our requirements on these features are more geared towards our ability to integrate them into our applications easily, than towards meeting the direct needs of our customers. This document contains a summary of our synergy with BSD in page 50.

Office program management and development will be coordinating efforts with the operating systems groups. The following table illustrates the Windows operating system requirements for the forthcoming releases of Office.

Office Release	Requires Operating System	Leverages Additional Client Features	Server Platform and Features
Office 95	Windows 95	N/A	Runs on NT 3.51 with conditional runtime checks. Exchange 1.0
Office 96	Windows 95	Incremental features in update to Windows 95	Requires post NT 3.51 (3.51 + Windows 95 shell and APIs).OFS supported as standard store (i.e. FAT)
Office 97	Windows 95	Incremental features in upgrade to Windows 95	Runs on post NT 3.51
Office 98	Next major release of Windows client	Key features of next major Windows client	OFS as a document library store

Table 3. Office product Windows version requirements.

Performance requirements for subsequent Office releases will become extremely important as we move to a market that is composed primarily of upgrade customers. A key issue for us is the overhead introduced by components that Office applications consumer, and notably the code provided by Office falls into this same category. A major concern is the fixed overhead that infrastructure pieces impose upon Office applications, especially if common usage scenarios do not include the code, yet initialization and startup invoke substantial portions of the component.

Our hardware platform for Office 95 and 96 is a 486 DX/25 with 8MB of RAM and VGA resolution. This platform will support running two Office applications (primarily Word and Excel), including running VBA macros. For Office 96 we will support Ren and one additional Office application on this same platform. For Office 96 we may require Super VGA for some applications or scenarios.

When Office is used as a development platform, that is the creation, editing, and debugging of VBA macros, we anticipate a larger memory footprint. We are in the process of evaluating the cost of the shared VBA development environment, including Forms³. We are working to make the record and play macro scenario require less memory than the full development platform.

For the Office 97 release we do not plan on requiring additional hardware. However, at this point we anticipate doing substantial work to leverage high color monitors, enhanced graphics adapters (i.e. hardware support for 3D operations), and high speed conferencing connections for video and/or data.

It is too early in our planning cycle to determine the hardware requirements for the Office 98 release.

6. Focus Areas

The following sections are the key focus areas for the Office Product Unit. These focus areas are designed to concentrate our effort on the above missions. Some of these focus areas are related to the process of producing Office and not a specific feature area. The focus areas are as follows:

Microsoft Confidential

-7-

FL AG 0043486
CONFIDENTIAL

MS6 5007104
CONFIDENTIAL

March 1995

HIGHLY CONFIDENTIAL

Office Product Unit 1995 Three Year Plan

- Application and Document Architecture
- Office Infrastructure
- Performance
- Development Tool Issues
- Far East
- Localization
- Macintosh
- IntelliSense
- User-Interface and User-Assistance
- Content and On-line
- Programmability
- Business Solutions
- Upgradability and Administration
- Using Exchange/Notes and Workgroup
- Conferencing
- Document Management

6.1 Application/Document Architecture (jonde, tonyw)

Full SDI

Although we were forced to cut SDI from Office 96 we learned a great deal about the large number of issues that we need to think through in order to make SDI work properly for our applications. We also have a much better feel for the type of support we would need from the system and therefore we can progress in parallel with the system so that we do not end up with competing windowing models as we have today with MDI. Finally, the work that PowerPoint did in restructuring PowerPoint 95 to be an SDI application (though it is not enabled by default) will be very helpful as we make this transition.

The following outline some of the issues we will address as we move to SDI.

Process model

Component re-entrancy will allow operations in progress in one (compound) document not to preclude operations in another. Also Office applications need to manage concurrent user and programmatic actions in the same (compound) document. PowerPoint has implemented one solution to this in '95.

Windowing model

Support for multiple SDI windows associated with a single task (or workspace, or project) requires window group activation behavior. Cairo USER previously had some support for this in a system-standard way. Projects and workspaces are topics that repeatedly come up in Contextual Inquiry research, which we feel Binders are the first step in solving these user problems. There is significant work to be done in this area before we deploy this solution. We need to consider the general classes of applications: productivity (small number of windows), development tools (larger number of relatively static windows), mail (large number of dynamic windows).

UI model thought out

For the user-interface model some issues include the behavior of Window menu commands such as Window New Window. How does that window behave? We will also need to introduce a formalization of

"companion windows." Toolbars are a form of companion windows in that when you minimize the host document, the toolbars go away. However, floating toolbars that are shared between documents should not go away but should be re-wired to work on the document that becomes active after the minimize. There are components like the VBE IDE that are also companion windows. The notion of grouping a workspace is also important to figure out. There will be the notion of truly global properties that are used to initialize new blank documents (think about auto-calculation state in Excel but there are dozens of these). This introduces a notion of templates that inherit the "default" setting from the current session instead of being a hard-saved setting. In general I think most properties will be saved with a template, but there will be a few that should be dynamic.

Some UI issues related to SDI include

- providing ways for UI customization to be made available to all windows of a given class of documents, or for UI for a given type of document component object to be shared, say in a floating palette, between windows even if their top-level class is different.
- providing a place for currently app-global commands like document creation and global settings. Places could include the Shortcut Bar (or what it eventually becomes in the system), or the social assistant.
- providing a way of grouping windows into workspaces where this served a legitimate purpose in MDI. For example, there will be cases where a group of windows will be created as part of a custom application, and they will require their own private global settings. There may also need to be activated or managed as a group. An answer may be binders, but we can't assume that the single-window model will be adequate for all workspaces.

Global versus document properties

Application and document state and properties will be partitioned among document, workspace/project, template/class, and per-user scopes (e.g. preferences). This will help applications to manage contention for access to global state, as well as providing a consistent user model.

No save

Persistent change logs will support implicit persistence of documents, with the ability to checkpoint and roll back changes (undo). Idle-time robust updating of saved state will be necessary, and will further add some degree of fault tolerance.

Going to no-save will also require a better understanding of version management. We will need to support the version-management uses of Save and Save As, giving the user control of when their current changes are committed to storage, and when this results in a new version being created. Doing this without confusing users as they migrate will be a challenge.

The ability for us to execute a user-model that does not require save will require us to think through the relationship of document management to the average user. The Save command serves as a lay-person's document management system, where users use ad-hoc conventions to delineate checkpoints and versions of documents. Our ability to provide document management in a seamless and obvious manner to the average user will be a major step we will start in the Office 96 release.

Full versioning & transaction support

Tracking version history, with branching and merging of changes will help in groupware scenarios. Cooperating with a transaction manager (e.g. Viper) will allow Office applications to participate in robust enterprise solution applications.

**MS6 5007106
CONFIDENTIAL**

Standard Document Attributes

Office document properties and methods will be further unified and standardized. Critical to this effort will be having a consistent interface across all the Office applications. We also think it is critical that the system support and evangelize the use of these standard properties. Following is a representative list:

- OLE Properties
 - More standard properties
 - EMail related properties
 - Notes related properties
- Page setup properties
 - Generic page that document content flows into
 - Paper
 - Slide
 - On-line Page
 - Margins
 - Header/Footer
 - Print settings
 - Watermark(s)
- Standard font
- Color scheme
- Styles in general
- Standard Methods
 - Save
 - Print
 - Version checkpoint methods

Template Model

We will need to support inheritance from multiple templates, and sharing user-interface elements between templates. Generic add-in commands can be stored in templates, as can document customizations (which act like subclasses, e.g. an expense report as a specialization of a spreadsheet).

Generic add-in commands can be stored in templates, as can document customizations (like subclasses, e.g. expense report). In general, commands are add-ins to views, and could apply to many customizations.

I think we need to introduce a notion of where exactly UI is getting stored for templates. Once you have templates based on other templates, you will see command bars which are stored in one template or another. We probably allow editing command bars in the active document but not in a referenced template. We probably also allow the template creator to just copy the command bars into a new template instead of refer to command bars in another template.

We need to support template versioning, with support for reconciliation, and conversion of instances of older classes to newer versions, by supplying default values, conversion rules etc.

Document as a form

Office documents as excellent front ends for groupware and enterprise applications require first class support for data binding, where data primarily resides outside the document, usually in a transacted store. We will want to support transparent transition between the cases of a document with (cached) data in it, for transfer (e.g. to take it home and work on it off-line), and a form-like viewer of external data. We have a weak form of this planned today, with OLE property exchange, and OLE DB client support. Object customization will allow for the same kind of form customization that VB and Forms³ provide. Customization can be used to add properties, and interact with external services such as MAPI/Exchange.

MS6 5007107
CONFIDENTIAL

6.2 Office Infrastructure (jonde, tonyw)

Office 96 will be the first release of the Office Infrastructure. We will continue to build upon this in subsequent releases. We view the Office infrastructure as a key competitive advantage and as such do not plan on taking on the burden of releasing subsets to other groups at Microsoft. Our plan for migrating features to a broader audience is through Windows, where features will be released to developers after first appearing in Office applications.

Move to Forms³

This is not a whole hog replacement of the SDM user-interface, but the goal will be to use Forms³ where appropriate. Some issues we are looking into include does Forms³ replace the implementation of command bars. We also need to evaluate Escher versus Forms³.

Event loop architecture

Overall this architecture will allow the host application to share the core event loop in a generic manner. This will include the sharing of idle time, event processing, accelerator translation, etc. The IDE integration work being done in Office 96 is the pilot for this architecture.

Command dispatch architecture

We will provide a common way for add-ins to access commands and command lists, and a common way for add-ins to add their commands to application user interfaces.

Selection architecture

A standard way for add-ins to write application-independent macros requires a standard object model with respect to active.document etc., but further, a standard way to reference the selection is required. Some commands will require "ref-edit" style modal selection, which needs to be part of the selection architecture.

Fully implemented text architecture

Word is implementing a shared text architecture that will evolve starting in the 1996 releases of our products. Office will be clients of this architecture through the Escher drawing layer's use of text services (Quill).

Property architecture

Standardization of properties across applications will allow us to implement common structures for passing properties around, such as Font, Fill, Line, etc., and to consolidate properties on similar objects, and common support for styles.

We will investigate shared and efficient storage of properties.

Leveraging the Office Architecture in Office 97

We have a great opportunity to build on the Office 96 shared features for our 97 "polish and preview" release. This will begin to deliver on the promised leverage from shared code, i.e. one group can write new features that all of the products get.

Examples of high visibility 97 features that would leverage 96 Office work include:

Social Assistant

- More user profile and context information make assistance smarter
- Social Assistant takes more of an active, automation role (e.g., filtering agent)

Shared Menus/Toolbars

- Begin evolution to a simpler user interface where rather than all functions available at once (on large menus and multiple toolbars), we move more toward an interface that adapts to the task at hand, and allows the user to feel more in control. In 97, we experiment with toolbars, menus, and the Social Assistant, to learn about larger directions we might want to take Office 98.
- Social Assistant can highlight menus, toolbars, and shortcut menus when giving guidance
- Copeland UI support for toolbars and menus

VBA Everywhere

- Improved development environment and solution-building tools

Shared Drawing

- Attractive and exciting graphics and effects
- Programmable graphics

6.3 Performance (jonde)

The following is a brief list of the known areas for performance improvements. Obviously over time we will be investigating which of these will show the greatest benefit. The most critical need for us in improving performance is the need for better performance measurement tools. We have an ongoing dialog with both Windows 95 and Windows NT on these issues.

Since the Office group provided shared components whose use will sometimes be required, even at startup, substantial effort will be put into insuring that the boot time cost of Office components is as minimal as possible. We will make sure that Office's services can be loaded at the discretion of the application wherever possible and that loading will be delayed until the service is actually required.

Some of the areas we will investigate include:

- Properties user-interface and style storage
- Dialog manager
- More caching of objects used across applications
- System API for discretionary memory
- Better compound files from systems
- Consolidate some subsystems
- Still better performance tools
- More and better lazy initialization

6.4 Development Tool Requests (jonde)

Not available at time of printing.

**MS6 5007109
CONFIDENTIAL**

6.5 Far East Development (stevesi, jonde)

Process

We are currently working on finalizing the Far East development strategy with MSKK. The goal of this strategy is to streamline the FE development of Office components. More details on this process will be available soon. At this time, applications are not impacted by the changes to the FE development process for Office.

Office 96

There are numerous opportunities to use the sharing in Office as a model for sharing the Far East technology. Office will be driving this effort working with experts in MSKK and the applications. The General strategy:

- All applications use UNICODE strings. Office APIs that use strings will expect UNICODE.
- Mso96.dll is either worldwide or at minimum single FE executable.
- Mso96.dll has an international DLL
- All applications use *Espresso* tools to localize

We are currently considering a large list of potential shared Far East features to be implemented by the Office group. Some of the possibilities include:

- Word breaking code. There are many flavors of this as there are different cases such as word selection and word wrap to take into account. Also, IME provides word breaking functionality but nobody uses it.
- Text normalization (all caps \leftrightarrow no caps)
- Narrow to Wide ASCII translation
- Number formatting. Many flavors here ranging from full featured general number formatting using the Excel code as a base through just providing date formatting. VBA needs to be brought in on whatever we decide to do here
- Number parsing raises many infrastructure issues.. Word uses their AutoFormat engine to sense cases.
- String manipulation. NLS API has problems in performance, bugs and Mac availability. Need hirigana \leftrightarrow katakana translation, accenting, and circle number translation capability.
- Character set conversions. UNICODE \leftrightarrow many codepages including Mac.
- Determining *lucky* days
- Text comparison. We have to handle collating and exact match separately, plus phonetic comparison. Sort key generation is also a required element.
- Shared text edit control as being developed as part of the text architecture..
- Shared Tables. We will investigate shared tables, or perhaps one implementation of text only tables (i.e. leave excel tables out of this investigation)
- IME integration. We want "level 3" IME integration with the shared text control (murrays). Interim steps may involve help dressing up the system IME windows.

Office Product Unit 1995 Three Year Plan

- Localization tools (franzr). We need to provide a translator to move our DBCS tokens to UNICODE and make sure all the localization tools are ported to UNICODE. Interim may be have localizers work in DBCS and we write a translator, but that breaks the no compile paradigm.

6.6 Localization and International (stevesi)

The Office localization plan is essentially as it has been. The improvements we will make in the release process starting in 1995 (automating the generation of setup INF files) are key to making our releases more timely.

The primary technical changes will be in the area of improved support for Far East and BiDi languages in the core products. Our goal is one world-wide executable. The near term goal, starting in 1996, is to have three executables: single byte, double byte/wide, and bi-directional. Starting in 1996 we are UNICODE/wide only.

We are moving towards Redmond based development when possible, for pilot projects in each of one tier. We are also moving to include more world-wide features in the core product.

Category	Languages	Timing
Tier 1	English-US, English-International, German, French, Japanese, Italian, Spanish, Swedish	Within 60 days
Tier 2	Portuguese, Dutch, Danish, Norwegian, Korean, Chinese, Finnish	Within 90 days
Tier 3	Based on business case and bi-directional	Within 120-180 days

Table 4. International Release Plan for Windows Office 96.

6.7 Macintosh (andrewk)

We recently held a BOOP review on our Macintosh strategy, led by Lewis Levin. This summarizes the strategy and provides some additional details on Office features. For the Office 96 release, this plan is still being finalized as of this writing. Subsequent Macintosh releases are highly dependent on the Apple Mac OS evolution and subject to change based on additional information and more concrete data from Apple.

Supporting the Macintosh for the minor release on our 12/24 methodology is proving difficult. Macintosh releases take about 20% overall of our resources for a release, which is a huge impact when we are working in a team of 5-7 developers. We will have a high level of efficiency in place in our 96 infrastructure, but this is still a risk area.

This plan assumes that the Macintosh marketplace and development efforts required by us remain much as they are today. Should the marketplace change significantly or should Apple make developing Macintosh applications substantially different than Windows, we will need to revisit this plan. Our intent is to work diligently on the Office 96 release for the Mac and re-evaluate the 1997 and 1998 plans as we have more information from Apple. Thus the plans here are tentative at best.

Office 95

No Mac as per plan.

Office 96

Mission: Great PowerMac applications that customers really like, built in a sensible way. Great personal productivity applications.

Microsoft Confidential

-14-

FL AG 0043493
CONFIDENTIAL

MS6 5007111
CONFIDENTIAL

March 1995

HIGHLY CONFIDENTIAL

Office Product Unit 1995 Three Year Plan

- PowerMac only
- Focus on end-user productivity features, not enterprise or data access features
- Everything we had in Mac Office 94; careful not to add new dependencies that would put us at risk.

Using above criteria, features that won't make it into the Mac Office 96:

- Access (including Datadocs and Designers), MS-Dev, Jet, DAO, OLE-DB
- Ren and Ren integration
- Binders
- Support for ODMA in document management and rich views on the file system

Features that will make it:

- VBA including Forms³, OCX, and the VBA IDE
- Will maintain ODBC 2.0 and Query 1.0 (potentially lots of work for Excel team)
- OLE 2 with specific enhancements
- Great backwards file compatibility (since many companies will also have non-PowerMac's running Office 94.)
- Targeted Mac-specific features to be a better Mac UI citizen; currently considering Apple Drag and Drop, QuickDraw 3D (perhaps via Escher), and other UI clean-up work
- Content indexing on local stores

For the Office 96 release we will rework shared memory code to make sure we will run under Copeland (schedule to ship around the same time as Office 96, not certified backwards compatible) We may need to make other fixes as well. True Copeland support will have to wait until Office 97 since we probably will not get code from Apple until June 95 the earliest. Adding new operating system features so late in our schedule is risky—we're assuming Copeland will slip. Just making sure we will run under it with specific Copeland features will be enough of a development and testing burden.

Plan for delivery on schedule:

- Keep Mac builds and testing in synch (ship Win+30 days)
- Same languages as Mac Office 94, but may consider dropping versions that haven't been successful. Will not give apps to Apple to localize, except if we drop an existing language.

Office 97

Mission: Same "polish and preview" goals as Win Office 97 + Copeland-savvy. This release depends greatly on the marketplace and on the direction Apple takes with the Copeland system.

We have to do a Mac Office 97 because:

- No major Windows upgrade excuse for doing Win-only
- Will be a new major Mac platform (Copeland) that might have already been out for almost a year
- Waiting until 98 might really hurt our Mac position.

Mac Office 97 has the same polish and preview features as Win Office 97, plus Copeland support:

Microsoft Confidential

-15-

FL AG 0043494
CONFIDENTIAL

MS6 5007112
CONFIDENTIAL
March 1995

HIGHLY CONFIDENTIAL

Office Product Unit 1995 Three Year Plan

- Copeland's new UI (so can support their visuals and user customization). This involves calling Copeland to draw interface controls. [Note: Our shared menu and toolbar code should make it so Office team can do menu and toolbar part of the work for everyone.]
- No OpenDoc, though will have to make sure OLE/OpenDoc interoperability addresses system needs
- Backwards file compatibility is still an important Mac issue
- Lots of open issues with Copeland. Will we be forced to require Copeland since Apple is pushing ISV's to develop for Copeland-only and Copeland won't be backwards compatible?
- Polish and preview features may have specific Mac adaptations—e.g., any voice input work on Windows gets adapted for Mac PlainTalk, any Social Assistant ghosting work will investigate supporting the look of Apple Guide.

Office 98

Mission: Same as Win Office 98 plus more Mac-specific, Copeland features based on user feedback, Copeland reaction, etc.

We have low confidence in any Mac Office 98 plans due to the many variables, including:

- Copeland performance—will it be fast and high quality and be a success?
- Gershwin—no idea what it will be, but Apple says this follow-up to Copeland will be ready in 97. We are especially concerned about the statements that Gershwin will not run existing Macintosh software.
- Mac market: will Mac Office 96 be a success? what will the Mac share be? ClarisWorks share?
- General divergence of Mac and Win operating systems However, our plan of record is:
 - Target Copeland machines (assumes Gershwin won't be until 99)
 - Hard Copeland work that couldn't make it into Office 97

6.8 IntelliSense Overview (samh)

IntelliSense is a loosely-defined set of design objectives rather than a specific technology or feature area. For the purposes of this three-year plan, IntelliSense is defined as "anticipates what I want, makes my work faster, and helps me discover the product." Given that loose definition, "IntelliSense features" span the range from simple UI changes to true intelligence, in the form of understanding and learning. Our approach to IntelliSense for the next three years ties to our product plans:

- For Office 97, IntelliSense features exploit the foundation built in 96. These features focus on UI changes and adding content
- For Office 98, IntelliSense features require new architecture and new technology from MS Research and elsewhere

Office 96 Highlights

- AnswerWizard 2.0 (aka "IntellAssist"): add context information; "guess" help needs; disambiguate terms with context; help topic hierarchy
- Office Social Assistant UI: intelligent window management; centralized/categorized/prioritized tip architecture; "look-at" animations; integration with wizards

Microsoft Confidential

-16-

FL AG 0043495
CONFIDENTIAL

MS6 5007113
CONFIDENTIAL

March 1995

HIGHLY CONFIDENTIAL

Information Resources

IntelliSense features with real "intelligence" require: (1) a source of information about the customer's needs, (2) analysis of that information, and (3) useful results based on the analysis. The following diagram provides an overview, with information sources at left and ways to use those information sources at right:

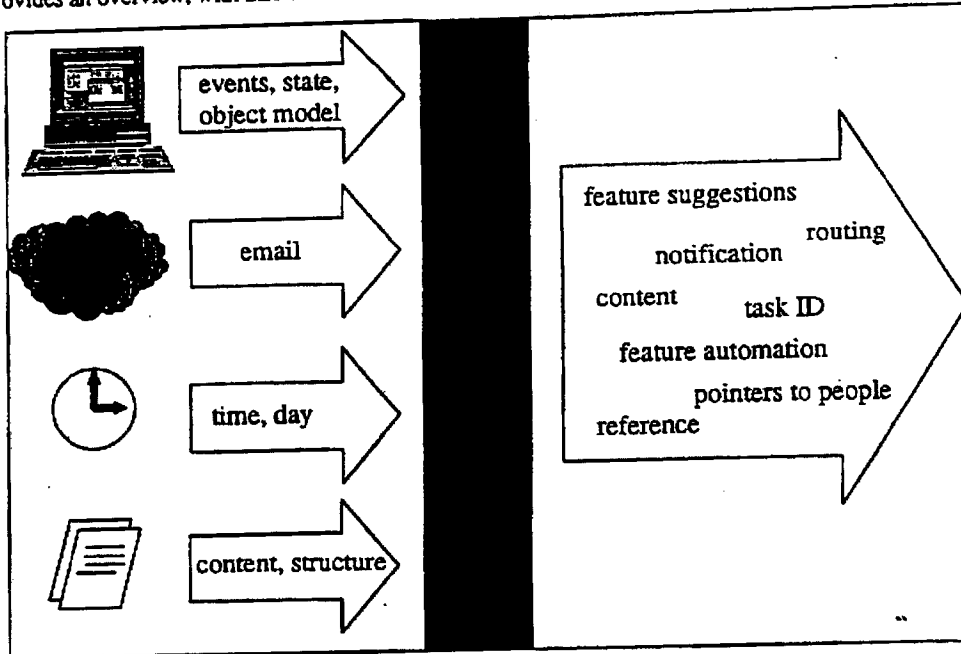


Figure 1. Information resources available and their uses

Sources of information include:

- events: the stream of commands, keystrokes, and mouse actions
- state: the state of the document, application, and focus of attention
- document content: the semantic content of the text and other document objects
- document structure: the semantic content of the document structure (e.g. salutation implies a letter)
- stated intentions: explicit goals stated by the customer (e.g. Answer Wizard query)
- EMail: the content and implicit meaning of email received and sent
- time: time of day, day of week/month

Useful results based on analysis and processing of this information include:

- features: application or recommendation of product features
- content: creation/insertion of document content
- tasks: identification and/or automation of customer tasks
- reference: connections to related information and resources, including help
- notification: reminders triggered by relevant events
- people: connection to or communication with others
- devices: intelligent routing to devices (e.g. portable PC, home PC)

MS6 5007114
CONFIDENTIAL

Thinking about the connection between each information source and each result type yield a number of interesting and productive scenarios. A couple examples:

- Based on observing a salutation, personal word choice (e.g. "Your pal."), and a short document length, it can be inferred that the task at hand is creating a personal letter. This information can be used to trigger the letterhead to use and a recommendation for Word's envelope feature. The document "type" (i.e. personal letter) can prompt intelligent routing to the recipient's home address by another application.
- Based on analysis of the document structure and content, an Excel worksheet is identified as a market projection for PC sales. Supporting forecast-data content from Dataquest's MSN service can be offered. Intelligent auto-formatting can be applied given the document's structure and purpose is known.

Key Technologies

To enable these scenarios, a number of heavy-duty technologies are required. Key among those technologies are:

- **Deterministic reasoning:** an architecture to monitor streams of information in real-time (as an interruptable background process), identify pre-written rules that are triggered, and trigger the actions attached to those rules. We have made steps in this direction with the Tip Wizard architecture in Excel and the Monitor architecture in Word.
- **Reasoning under uncertainty:** an architecture to manage uncertainty of non-deterministic triggers [We'll rely on MSR's Decision Theory Group to guide us in developing the architecture and doing the modeling. Need to reduce current modeling costs to make this cost-effective]
- **Pattern recognition:** an architecture to scan streams of information and documents, identify known or new patterns, and represent those patterns in a useful form [MSR's assistance is required to evaluate appropriate technologies]
- **NLP:** understanding customer input and the semantic meaning of text information [Will rely on MSR's NLP team for this technology]
- **Information retrieval:** this is key to the success of the Office Character. NLP may provide a solution here. Otherwise, we need to continue improving and lowering authoring costs for Answer Wizard.
- **Online:** seamless connection to supplemental information and resources as well as remote control of a connected computer

Having these technologies (or architectures enabling these technologies) provided as Windows system services is desirable. For example, an event model that could feed into the reasoning systems would be nice.

Architecture Thoughts

The architecture to implement these technologies also has a number of important requirements:

- *A common event model and representation* among the Office applications is necessary to share cross-product knowledge and rules. For example, an "Edit Copy" event in Access has meaning for when the customer switches to PowerPoint. Shipping the instrumented version might serve as the basis for this architecture as well as provide a large pool of research candidates and data.
- The rules and pattern-recognition capabilities must be implemented as *shared services*.

- A *customer profile* needs to be created and "attached" to the customer. The profile contains information about the preferences and expertise of the customer. For example: commands used, profile of work pace, help topics viewed, proficiency assessments and usage frequency for larger activity areas, profile of work versus time of day and day of week, email priorities and habits, etc. The profile is continuously updated and refined by all Office-compatible applications.
- The *Office object model* is a resource to be leveraged. The built-in relationships in the object model can be used as a reference for determining customer needs and refining the IntelliSense we offer. For example, answers to help requests can be gracefully targeted to topics related to the parts of the object model being used by the customer. Similarly, feature recommendations can be gleaned by exploring unused properties/methods on commonly-used objects.
- Allowing ISVs to *customize and extend* our IntelliSense architecture is a big benefit. In addition to being able to add new rules, for example, ISVs should be able to insert a new pattern recognition module.
- The success of "quicktips" in our 95 releases increases our desire to extend the "self-documenting interface." We may decide to tie the assistance text more closely to our applications than relying on Win/Mac Help as the containers for assistance text. This would require both architectural changes as well as process changes, but the benefits are more control and better performance.

UI Implementation Techniques

There are a number of UI and implementation techniques we can use for a better IntelliSense feature set:

- *Invisible data collection.* The monitoring and processing of customer information should be done in the background, invisible to the customer. The CPU of most computers sits idle most of the day; this is an opportunity for Office to be continuously monitoring and analyzing data for patterns and opportunities.
- *Customer control.* Given that IntelliSense features can be "wrong", it is important that customers have a way to explicitly control how much IntelliSense they get. In addition, it is critical that customers be able to understand, edit, and control the profile information being gathered about them.
- *Conservative exposure.* Given the uncertain nature of IntelliSense, we must be conservative in exposing the results. The UI to signal the availability of an IntelliSense feature should be recognizable, but unobtrusive. Undoing IntelliSense should be likewise. IntelliSense should be based on probability to allow the setting of a confidence threshold below which an IntelliSense feature is not exposed. Finally, we should be smart about the timing of IntelliSense exposure. For example, accumulating non-time-critical feature recommendations (i.e. "tips") not only reduces the number of times the customer's work is interrupted, but allows useful categorization and prioritization.
- *"Multiple-choice intelligence"*. For situations involving uncertainty, presenting the customer with multiple possibilities increases the chances we'll be "right" and also increases the perceived intelligence. Word 95's background spell-check UI is a good example.
- *Social UI.* The Office Social Assistant is a useful place for exposing IntelliSense. Multiple Assistants might be used, with each Assistant representing one "type" of IntelliSense (e.g. an artist for formatting IntelliSense.) Customers are able to identify the category of IntelliSense being offered simply by seeing the Assistant "advertising" that information. Customers can choose which Assistants they want to hear from.

Office 97 Features

- *Fuzzy UI.* Fuzzy UI provides pseudo-IntelliSense in an inexpensive way. We take a look across the Office products to find applications for fuzzy UI (e.g. "about a month ago" for find UI.)
- *Shared rules engine.* A shared rules engine is added to MSO9x.DLL for processing of tips and other rule-based features (e.g. AutoCorrect.)
- *IntelliSense UI.* A standard and unobtrusive UI convention is established for indicating availability/application of IntelliSense.
- *Focused troubleshooting.* The top-20 PSS problems are addressed with Assistant-based troubleshooting UI.
- *Online help extensions.* Online help is seamlessly extended with connections to the Knowledgebase and other PSS support offerings.
- *AutoX improvements.* Our existing AutoX family of features (especially Word's Auto writing/editing/formatting) are improved to the next level.
- *IntelliSense control center.* A central UI for controlling IntelliSense is established.
- *IntelliSearch authoring improvements.* Automate or partially automate authoring of databases.

Office 98 Features

- *NLP.* NL understanding of customer requests via the Assistant, and NL understanding of document contents.
- *Shared pattern recognition.* A shared pattern recognizer is added to MSO9x.DLL for analysis of events, document structure, and time.
- *Shared inference engine.* A shared inference engine is added to MSO9x.DLL for processing of non-deterministic triggers.
- *AutoX improvements.* Our existing AutoX family of features continues to be improved.
- *Online remote control.* Remote control of the customer's PC by PSS engineers allows the ultimate in IntelliSense and provides a test ground for "assistance for hire" services.
- *Office as a gateway.* Office 98 is the connection point to resources on the net: (1) to people with similar interests & backgrounds; (2) to data relevant to the Office customer's work; and (3) to events of interest to the Office customer.

Related Features

There are some features that don't fit into the IntelliSense discussion, but are thematically related:

- In Office 97, the Assistant's UI capabilities are extended to be able to provide step-by-step (i.e. "cuecard-like") guidance through the product.

In Office 97, more on-line help is converted to use the Assistant. We intend to invest in the implementation of the assistant to provide richer visualizations and interactions. For example, we would like to use bitmap as well as metafile assistants, regional assistants, and more sprites. We will work to obtain additional system support as well as richer assistant creation tools.

6.9 User Interface and User Assistance (chrisgr/DAD Research)

Overview

Office 97 will be a 12 month release. Therefore we should avoid major UI disruptions and avoid features requiring expensive new infrastructure. In addition, we should base many Office 97 features on infrastructure work done in Office 96, to maximize the productivity of our development effort. This could include the Escher shared drawing layer, VBA, Command bar, Shortcut Bar, and Ren/Exchange, as well as Windows 95 extensibility mechanisms.

Office 98 is a 24 month release, probably near a major system release, so larger infrastructure and user interface changes can be considered.

This section highlights the problems we need to solve in all of our applications. Office will be working in concert with the applications to prioritize these along with areas that are specific to applications. When possible we intend to leverage advances in user-interface across all applications using the office document architecture or at the very least shared code across applications.

As a process note, the design of user-interface features that are common to all of our applications is not necessarily owned by office program management and/or development. We will continue to have features that are Office-wide, yet they are owned in applications.

User Interface for Commands, Properties and Tasks: As we add more and more capabilities, and capabilities become increasingly dynamic, our static menu based user interface is reaching the breaking point. We will need to rethink the command and property user interface. There are several objectives that we need to try to support:

Maximize interaction efficiency: Users need to be able to perform frequently needed actions quickly and efficiently. A potentially useful way of characterizing user performance is along a spectrum of skill based, rule based and knowledge based interaction, which comes from research on cognitive psychology. We should try to move as much interaction as possible toward skill based, or rule based styles to lower the mental load placed on the user by the interface.

In general we can say that *skill based interaction* is enhanced by the having UI objects in reasonably accessible, predictable locations when they are needed, but avoiding the presentation of too many similar objects, which would make it difficult to find the required objects. *Skill based interaction* also requires avoiding complex side effects from UI actions, avoiding the need for too much advance planning, avoiding assuming that the user has deep a mental model of the interactions between objects being worked on, allowing the user to perform actions directly rather than indirectly through secondary user interface objects, and providing multi-step undo for as many actions as possible.

Rule-based interaction is an efficient way of performing *multi-step-tasks* that can't be done in a single step, or of performing tasks where the user may not know a single step shortcut. Efficient learning and use of rule-based interaction requires that tasks be able to be performed by a sequence of steps that is stable and predictable across a wide range of context without too much thought or advance planning to adapt to a given situation. We should think about the types of actions we expect users to be able to perform using a sequence of steps and think through ways to keeps the steps the same across a wide range of situations. In addition, we need to find ways to allow users to identify situations where a given series of steps can be used. Some of the answers to these questions are could be an outgrowth of today's cue cards, migrated into the social interface.

Another way of performing *multi-step-tasks* is with a Wizard. Unlike rule-based performance of tasks, where the rules are in the user's head, Wizards package the steps in a canned sequence, which may branch based on user decisions. For very well-defined tasks, where the user would be overwhelmed by having to make all necessary decisions in a single step, Wizards will continue to be useful, but all other

things being equal, it is probably preferable to use a cue-card like approach to teach the user the primary user interface, because the user will be more readily able to generalize and adapt this knowledge to other situations. This will be especially true where applications may be upgraded one component at a time, which would be more likely to break a Wizard than a learned sequence of steps.

Maximize discoverability and recallability: We want users to become aware of new capabilities of their software that could benefit them, and we want users to be able to remember them in situations where they could be useful. We also want users to be able to perform these actions quickly and efficiently if they find they need to do them often.

Discoverability is enabled by some combination of making capabilities visible at the right times, and telling user about capabilities at a time when they could benefit from them. However, if too many capabilities are simultaneously visible they won't be found, and if users are told about capabilities when they don't need them, they will either be annoyed or at least not remember them. Either way, we'll have to know more about what the user is doing, either by the user indicating this through some explicit choice, or by recognizing patterns of actions. The social interface will be an important way we will educate users about capabilities, but we will need to have a better infrastructure for detecting what a user is doing and what capabilities are likely to be useful, and for the user move to more efficient interaction styles for actions that are done often.

Provide optimal "places" to do certain types of work: The contextual inquiry research suggested the idea of "places" where the UI is optimally adapted to the task at hand (like rooms in a house). Users perceive places as the physical manifestation of a task (i.e. a kitchen is for the cooking task). As we understand this better, it could be a promising direction to evolve the UI in the future for well-defined tasks. Examples could be delivery/printing, and collection. The result could be an interface that doesn't overwhelm, but encourages users to try new things out. In 97 we could experiment with command bar, menu and assistant combos. In 98 we could experiment with more visual change, for example a direct-interaction diagrammatic interface.

Improve techniques for direct interaction with objects: As the user interface increasingly supports a wide range of types of objects, all of which may come and go, and be visible simultaneously at varying degrees of activation, it seems inevitable that we should move as much interaction as possible to be performed locally on objects themselves. In the real world users interact with most objects by going closer to them and examining them, rather than indirectly. Context menus, property sheets and drag/drop are examples in today's user interfaces.

However, there have been obvious problems of discoverability, browsability, understanding of scope, awkwardness of using the right mouse button (especially with the pen), providing a simple interface for the huge number of things that can be done to some objects, providing a quickly accessible interface for frequently performed actions, etc. There are also issues of where to put commands and properties that do not have obvious objects to be attached to, and issues of performing actions across mixed selections of objects.

In spite of these issues, directly interacting with objects seems so natural that there ought to be a solution to these problems, possibly helped by some new user assistance and interaction technologies that will be available. We should not let limitations of today's implementations cause us to overlook this. We should set as a goal (subject to the above assertions being proving to be correct) that by the time we get to Office 98, the primary way of performing actions on objects be directly interacting with them. The social interface will be the way of discovering and browsing capabilities, and it should teach the user direct interaction techniques as the primary means of interaction. The social interface could also be a way of teaching or performing global actions that don't have natural objects to hang on. Discoverability can be enhanced by providing visible affordances, possibly that appear only at the "right time", that will remind

the user of where to go and what to do to interact with any object. Of course, for consistency, we would also want the same interaction techniques to be used by the system.

Better support for user assistance in the user interface: In support of the social interface, we need to add better capabilities for ghosting and indicating all parts of the user interface. We should do all of this we can for Office 97. Maybe the system could add some of these capabilities in their minor release after Windows 95 and before Memphis. If not, we would want extensive capabilities of this type from Memphis, as well as the ability to detect all aspects of the state of Windows and its user interface.

We also need to make available to the user assistance system much more information about the state of the application and the user interface. This has been a problem in developing the intelligent user assistance for Office 95 and 96. Anything that the user would see as relevant state should be available to the user assistance system to enable it to make intelligent inferences about what the user is doing, and to offer intelligent assistance that is exactly relevant to a given situation.

UI Infrastructure: The infrastructure for all common UI should be built from VBA in 98. There should be an eventual convergence between Office command bars, dialogs, and Forms³.

Documents

Creating and viewing documents is the reason for existence of most desktop applications. Even traditionally non-document-centric application like Access are moving towards a document-centric way of representing themselves.

Today, documents can be thought of as objects that are used to organize and exchange information. They contain clearly bounded and coherent bundles of information, with content, structure and properties bound statically, because that is what the technology of paper documents permits. Our machine readable documents, which emulate paper documents, are somewhat more flexible, but not nearly enough. Now that we have the potential to supply, assemble and format information dynamically, the tight binding between content, structure and properties can be broken-down. By rethinking the underlying purposes documents, we can better support the new capabilities that ubiquitous connectivity will make available.

The role of documents as information objects will still have a role in terms of managing scalability of access to information based on one's current connectivity to the knowledge base. i.e. If one is less connected due to being on a slow or non-existent link, we want to maintain as much capability to do useful work as possible, maintaining access to useful information. In this respect, documents will be seen as bundles of information whose availability and currency should be maximized by the replication and synchronization system.

Document viewers should remain much the same across the full range of degrees of connectivity. However, an issue will be how useful document viewers can be provided across devices of varying form factors, while maintaining as much as possible the user's mental model of the content, structure and formatting of the information. For example, what is the most useful representation for a user to consult a large text document from a palmtop device, or over the phone?

Links

We will need a broaden concept of links to support polymorphic objects, and to be independent of storage location. They need to be able to find not only an exact instance of the object originally linked to, but the most accessible and/or up to date instance of that class of objects each time the link is resolved. For example, suppose a user creates a link in a document to a piece of information inside Encarta, while Encarta is in a local CD ROM drive. Later, suppose someone is reading the document we would like to be able to resolve the link to find a version of Encarta that is most accessible at that time, even if it is over

Office Product Unit 1995 Three Year Plan

MSN or the Internet. Another alternative if the link cannot be resolved, would be to prompt the user to insert the CD ROM or connect to a network, etc. Note that these links are a lot like queries.

For Office 96 we will be pursuing an office-based implementation of monikers that will enable office documents to be better citizens in link scenarios. Applications will also be doing work to become better servers of Monikers and to integrate better with the Windows 95 shell shortcuts, starting in Office 95.

SDI

We need to move to SDI as soon as possible. Because this is not planned for 96, and 97 is a 12 month release, we should target SDI for Office 98. Key issues we will have to resolve include

- We need better UI techniques for providing shared user interface across a class of compound document objects. We don't want to duplicate UI elements in every SDI window. We want settings like command bar customization to be available globally.
- We will need better ways of managing related groups of SDI windows. Some concept of workspaces seems to be necessary.
- Allowing per-document settings for global state in today's apps.
- How can we provide global settings that affect the behavior of a given class of objects? A possible answer is to store these settings in templates. Changing the settings in templates the user can change the behavior of all future documents created from the template. However, this only affects future documents. Is there a way to make settings that work retroactively on a class of documents? This gets complicated, as certain existing documents may require their existing settings. Another issue is how we ensure that all appropriate documents or viewers have access to changed settings, since they may not have access to where global settings are stored.

Office Shortcut Bar (MOM)

It is assumed that we will keep the Shortcut Bar in Office 96. After having provided MOM in Office 94 and the Shortcut Bar in Office 95 we should not remove it until the system provides capabilities that users would consider to be reasonable substitutes for the Shortcut Bar's capabilities.

Since Office 97 is a 12 months release and is unlikely to be synchronized with a major release of the system, it is assumed that in Office 97 the shortcut bar will be much the same as in 96. It would only be enhanced if this moves it in a direction that is compatible with the unified MOM/Task Bar that we would like to see in the next major system release. For example, if the System were going to adopt Office command bar technology to replace menus and toolbars, it would be good for MOM to be based on command bar technology either in Office 96 or 97.

The key capabilities of the Shortcut Bar we would like to see in the system shell include:

- Elimination of the distinction between launching an application, and switching to it when it is running
- User customizable, fast access to most frequently used objects and locations.
- Place for cross-app Office functionality to be located, for example, document creation. The Shortcut Bar is the first instance of a (rather simplified) shared toolbar.
- Quick access to objects on the desktop without major disrupting the user's working environment
- A related issue is where the social assistant is located for cross-app, app-system and system-only scenarios. The assistant is another object that needs to be available all the time.

MS6 5007121
CONFIDENTIAL

Microsoft Confidential

-24-

FL AG 0043503
CONFIDENTIAL

March 1995

HIGHLY CONFIDENTIAL

Office Product Unit 1995 Three Year Plan

- If the system really does replace MOM, then what will provide the Office identity? (I think the answer could be the social assistant.)
- An interesting solution to both unification and identity could be to unify MOM with the Task Bar, and the social assistant with the Start Menu. The system could ship with the unified Task Bar/Shortcut Bar and Start Menu. When Office was installed, it would replace the Start Menu with the Assistant.
- We need to clearly define the distinction between in-place switching (as in Excel Workbooks, in Office Binders, or in Ren), and cross-window activation and switching as in the Task Bar and MOM. The Ren main window for '96 looks a lot like an Office 95 Binder, and serves to allow related views to share the same window. Therefore, maybe they should be unified into a more general type of viewer for '98.

Subject to resolution of these issues, we would like to see the Shortcut Bar be dropped from Office in the first version of Office designed for Memphis, to be replaced with a merged version of the system Task Bar and the Office Shortcut Bar.

Natural User Interface

As we have had more computing power and interaction technologies available to us, we have in part used it to make the user interface more natural. That is, to make it match human sensory, mental and manipulative skills better. The move to GUI, drove a "sea change" that revolutionized computing and caused the leaders to change in almost every category of computing.

Although GUI with direct manipulation is more natural than the character based interfaces that it replaced, it is still unnatural in many ways. In the next 3 to 5 years, much more additional computing power will be available, and important new interaction technologies will be built into the operating system, presenting many opportunities for making the user interface more natural. After the connectivity revolution, the natural interface revolution will probably be the next "sea change" that we should plan to exploit to maintain our competitive edge.

These are some technologies we will have available:

Speech: Commands, Dictation, Output, Phone

Speech over the phone: We should look into providing the ability for commands and limited information retrieval over the phone as one of our higher priorities related to speech. This could be especially useful for a personal information app like Ren. It would be great if we could find the design and development resources to do this in '96. However, given development resource constraints, Office 97 is probably the release to plan on this.

Hand and eye-occupied interaction: Speech can be a useful side-channel for interaction with the computer in situations where the user's hands and eyes are to some extent occupied on another task. Speech or sound output can be used for confirmation where the user's eyes are also on other work. Examples could be system control or information capture for people doing things like hands-on technical work. These will likely often be custom made solutions, so we would need support for these capabilities in our customization tools. With research, we also may find some SMORG applications that we could support directly that could be a win for us.

Dictation: We should consider supporting discrete speech dictation for Office 96. We should definitely support it no later than '97. This is a key feature for Far East markets, but could also be a compelling feature for domestic markets. Our speech technology can allow the user to switch between continuous speech and discrete speech, depending on the level of accuracy required. Accuracy will be excellent for discrete speech. As we expand computer usage to less expert less-frequent computer users, fewer of our users will be proficient typist. We shouldn't underestimate the potential importance of dictation to these

users. This would be a good feature to add for 97 since it probably does not involve infrastructure or major UI changes.

Commands: Voice recognition for commands is certainly technically easier than dictation because of the benefits of context in narrowing the possible vocabulary. The ability to give voice commands to applications is necessary for all the above scenarios. In addition, speech could be a convenient side-channel for commands where data is being entered through a primary channel, such as the keyboard or by pointing with a pen. For example, a keyboard users may benefit from having the option of being able to issue voice commands rather than removing their hands from the keyboard to use the mouse, or having to remember keyboard shortcuts. Similarly, a pen users on a small form factor devices may benefit from being able to issue voice commands since the display is too small to make many UI elements simultaneously visible. There is an opportunity to model the user's current perception of the context to make available a more natural vocabulary, and as a basis for better recognition. For example, the user should be able to describe text in terms of words, sentences and paragraphs.

Deferred dictation: One of the reasons for lower accuracy in dictation systems is that real time response is required. It might be interesting for some applications to allow the recording of speech at a given time that would be fed into the dictation engine later, even overnight, which would allow much better recognition. For applications like field data capture, one might even envision the user just carrying a relatively low powered palmtop computer, or even a tape recorder, the output of which would later be fed into a more powerful computer for recognition. For applications like this, it would be useful to retain and be able to play back the speech transcript while viewing the recognized text for validation or correction.

Interacting with invisible objects: Speech is a natural way to interact with or summoning unseen objects. Thus it could be an interesting way to unclutter the screen. An issue would be recallability.

Speech output: Computer generated speech would of course be necessary for the phone-access scenario described above. Speech could also be useful in scenarios such as reading back a list of data for validation. In addition, speech could be a natural way to give provide feedback to the user on system status or events. However, we need to think about why some uses of synthesized speech have been unsuccessful, or even annoying, such as talking instrument panels in cars. I think this is because it's not natural for a person to interrupt another person with minor information without first requesting attention and waiting to be acknowledged. We can probably develop an equivalent capability in the computer, possibly with the social assistant. Another issue is that if we have the assistant talk, users may expect more intelligence than we are prepared to deliver, resulting in disappointment.

Visualization and Navigation

Today's user interfaces underutilize humans' ability for spatial processing, including the ability to navigate spatially, to reason and see relationships spatially, and to recognize and recall spatial information. It is true that today's user interfaces lay out objects that are on the screen spatially in two dimensions, allowing them to be manipulated. This is a key reason for the power of GUIs. However, in today's GUIs viewing and navigation to objects that are not visible is usually by a wide range of inconsistent techniques, including from scroll bars, task switching buttons, running an application, hierarchical navigation, opening folders, zooming, opening property sheets, opening menus, closing windows to access the desktop, etc.

The result is that there is no consistent spatial model that the user can use to reason about the behavior of the system. Furthermore, interaction is limited to the finite size of the screen, and anything that is not visible on the screen must be reached by any of the above inconsistent, discontinuous, and not directly reversible techniques. A side effect is that we've cluttered the screen with controls because the user's field of view was restricted by tunnel vision of screen. Unlike the real world, users can't maintain a mental spatial image of what they can't see, and get to it by, smooth, continuous panning and navigation. Instead, on today's GUI systems, users must maintain a symbolic image of what they can't see, and navigate to it symbolically, and which is much harder.

MS6 5007123
CONFIDENTIAL

March 1995

Microsoft Confidential

-26-

FL AG 0043505
CONFIDENTIAL

3-D can make interaction more natural. It's not clear that we'll have the computing power or that we would be able to migrate user to a new interaction style quickly enough to base the entire UI on 3-D in 1998. However, we could use 3-D techniques to solve one or more important and strategically significant UI issues for which 2-D solutions have eluded us. Example could be window management, task switching, workplace switching, hierarchical/network navigation, and navigating in documents. This will also require some thought about where we'd ultimately like to go with 3-D so that our first steps don't cause problems of backward compatibility.

Another ideal use of 3-D would be in information visualization. There has been a lot of research into 3-D structures for information visualization and navigation. In Office 97, and 98 we should start using 3-D techniques to help users work with complex information. For example, we could use graphics including perspective to show the relationship between documents in a document library. Another example could be to provide a time-visualization wall as in the video produced by the graphics group. There are many possibilities to investigate.

To make full use of 3-D will require new interaction techniques, and more degrees of freedom from the pointing device. We need to develop a pointing/navigation device, and the supporting software techniques, to provide manipulation as good as a mouse, but to also to provide smooth intuitive navigation through space.

Multi Modal Interaction

People interact with each other and with the world through multiple input and output channels simultaneously, and one channel tends to compensate for weaknesses in others. For example, speech is enhanced through the visual channel, through gesture, eye contact indicating attention, etc. Speech communication on certain topics is greatly enhanced by looking at and pointing at images, illustrations or diagrams. In addition, speech is enhanced through reference to shared knowledge, history, or context. It is possible that through the clever use of complementary interaction techniques, and decision theory, we can make more natural interaction feasible sooner than if we waited until these techniques were individually perfected.

Decision Theory

Decision theory can play a role in many areas of the user interface. As we depend more on decision theory, we will need increasingly detailed data on user behavior, both in advance to test and calibrate our models, and in real time for the models to detect user actions. We'll need data both at a higher level, such as inferred user intents, and at a lower level, such as mouse movements, hesitations, etc. If in the future we could detect things like the presence of the user, eye movement, etc., our models could be even better.

We need to develop an enhanced infrastructure for instrumentation of our software to detect user actions. Preferably this should be in the infrastructure of the main product so that we can collect information quickly and whenever we need it. We should start planning for this as soon as possible, because we need this information in advance to develop and test user models.

The Microsoft Network could be valuable as a means for collecting this information. In addition to basic usage statistics, it would be very interesting if somehow over MSN we could detect user requirements, even for capabilities that we do not yet provide, as an aid to discovering the need for new features. As we move to a serviced-based business model from a product-based one, this could be a great way of researching and marketing specialized capabilities or services. A simple way we could detect additional requirements would be to capture unsatisfied requests typed by the user into the user assistance system.

Non Rectangular Windows and/or Sprites

To support the social interface and other types of graphical interaction, it would be very helpful for the system to support sprites that can be moved without interfering with other UI elements. We would want the ability to animate these and control of the z-order. A more powerful alternative would be non-

rectangular windows that would support all the current APIs and events that Windows supports for programming into rectangular windows.

Interaction and Communication Techniques

As the complexity of software increases, the need for more diagrammatic user interfaces increases. Current user interfaces make little use of diagrams to aid in the explanation of the system. Current examples of diagrams include: Previews in dialogs, quicktips, and large cards. Some user-interface controls also act as diagrams. Hierarchy navigation trees, for example, are diagrams of containment. While these have proven to be nicely scaleable and somewhat efficient for a wide variety of containment scenarios, they tend to be so generic that they lack visual cues that provide landmarks that are proven to aid in navigation and recall. File folder icons, for example, are predominantly the same size, shape and color even though they may contain a diverse collection of objects.

As we move to a component object world, this problem becomes increasingly complex. Using one tree visualization of file storage, Word and Excel outlines, Binder containment, VBA projects, PowerPoint slides is inadequate. In order for users to shop for objects, navigate containers and manipulate objects, we need better ways of categorization and better ways of diagramming complex object containment and inheritance.

Following are proposals for alternative interaction and communication techniques:

Z Order Visualization

Z-Ordered graphical objects are prevalent throughout the system. Windows, graphic objects and controls, text frames and spreadsheets are examples of objects that can exist in z-order yet we have no efficient way of diagramming that containment. One popular technique for visualizing containment or arrangement of objects is an 'exploded' view. This view is consistently used when diagramming a complex automobile part or electronic device. This view is also useful for displaying z-order and for selecting hard to select objects stacked in z-order.

Task Flow Diagrams

Common in the programming world for charting the flow of programs, this technique is also useful for showing the flow of a user's tasks, tools and objects. In general terms, this is a great technique for diagramming the taxonomy of business processes or task flow.

Because task flow is not inherently obvious or always available, diagramming task flow is somewhat challenging. There isn't one generic task flow in Office usage, so it's apparent there would be many task flow diagrams. One way of identifying task flow is to generate it from user data. In fact, we may find vertical markets possess a standard set of tasks that could easily be illustrated in a common way. Users would use the task flow diagrams as means of exploring and navigating the tools and objects available to them based on the context of their current task. This would also act as a way of filtering menu structures and dialog boxes. (How's this different than visual programming? Current task flow/visual programming tools usually require the user to build the task flow themselves when the flow of the task may not be apparent to them - thus the need to build flows for them or ship default flows)

The "Getting Results" books for '95 are a glimpse of such task structuring. If we applied this approach to the user interface, we may find that experienced users find new and interesting functionality while providing an easy way for new users to browse the potential of the system and learn the tools at the same time.

Concept Diagrams

Large Cards in '95 approach this technique, but they're not coupled with the UI. Large cards are a part of help that diagram a particular concept, but show 'snapshots' of the UI and the user is forced to take what they learned in the help context and apply to their work context. Often times, the lessons learned are

FL AG 0043507
CONFIDENTIAL

forgotten because the contexts are too disjoint. If, on the other hand, large cards actually used the users content instead of made up content and the popup quicktips contained actual controls for manipulating that content, the user would not be forced to leave their context. This is basically taking the notion of diagrammatic help (like Large Cards) and merging them with dialog boxes and wizards..

Event History Diagrams

The Ren Timeline is an example of this diagram that will ship with Ren in 96. As it's currently designed, the time scale granularity goes from 2 hours to 2 months. This may work for time management purposes, but if the granularity went down to minutes or seconds, we could use that recorded data for:

- **undo/redo:** The user could step back in time and 'see' the events that have taken place over the course of their work session. Where appropriate, we could also supply a snapshot of the screen or object for reference.
- **programming by example:** When faced with a repetitive task, the user could go back in time, select a series of actions and based on some rules and parameters build a macro out of that sequence.
- **revision merging:** When the user is faced with merging revision of their work, they could stack the history of the artifact on the time line and visually see how the artifact has changed over time with a track for the resulting merge.

Other Charts, Diagrams and Maps

- **Cluster diagrams:** Cluster diagrams for plotting file find and other search results. Each cluster could represent different search criteria.
- **Network Maps:** Mapping printers, servers, web sites, computer locations, colleagues and other on-line information is a useful way of recalling and retrieving information.
- **Data Clouds and cone trees:** Another form of clustering, data clouds and cone trees make use of the third dimension for displaying and filtering information.
- **Fish Eye Lenses and other 'local' filtering:** Most of the above mentioned views act on a global set of data. There are other techniques, like fish eye lenses, that act on a subset of data. The user can 'fence pick' a region of information and have only that area be affected by the filter. Because the use is acting on a smaller set of data, this could result in more refined filtering and increased performance. Other examples of local filtering include x-ray like lenses that only see certain information. Selecting a range of cells with a formula lens reveals only those cells that contain formulas. Selecting a range of text with a 'styles' lens to show a particular style or set of styles. Selecting a group of filled polygons with a 'border only' lens so you can select hidden polygons.

Audio and Visual Techniques

In addition to diagram graphics, there are other graphic techniques that could be useful and compelling in a software environment. Many of these techniques are used in TV and movies today to establish mood and convey information..

- **Animation:** More animation could be used to establish context and origin. For example, status dialogs could animate out of the status bar region or a dialog box could animate out of a menu item. Win95 allows for this animation of windows, but apps could do similar things with dialogs, toolbars and tool palettes. Many of the PowerPoint transitions and build could be applied to page turning, tabbing displaying of dialogs and forms. (Form1.Visible(cut)=0/1
Form1.Visible(dissolve)=0/1). Third party OCX controls exist that do this today.

MS6 5007126
CONFIDENTIAL

March 1995

Microsoft Confidential

-29-

FL AG 0043508
CONFIDENTIAL

- **Translucency:** We need Raster Operations for variable translucency. Win95's 'screen door' transparency for icon dragging is a glimpse of the possibility. Adjusting the translucency of a given window would allow the use to see through a floating toolbar or dialog to their content while still being able to interact with the user-interface controls. This technique is becoming increasingly popular in TV graphics during sporting events. Player's statistics are displayed 'over' the game without too much image degradation. (this technique is used in some of the ITV demos). Variable translucency would also be useful in Escher for translucent fonts and graphics.
- **Lighting and Shadows:** Sometimes lost in the fervor of 3D geometry, lighting techniques can yield compelling effects for creating mood and for highlighting elements on the screen. In today's windowing system everything has equal ambient lighting properties with a subtle illusion of an 'upper left corner' light source for 2.5D UI controls. Imagine that light source traveling with the arc of the sun throughout the course of the day. All the shadows on the icons would shift as well as the button and window highlights and shadows. Even windows and dialogs could cast shadows and vary in length and projection based on the season and time of day. With lighting techniques like these, we could allude to a 3D environment with the primitive 2D geometry used in windows today. Spotlights could also be used by the help system to highlight elements of the screen.
- **3D Assistants:** Current social assistants are flat and dated. With accelerated 3D APIs we could construct assistants with 3D geometry instead of 2D metafiles. This could result in much more realistic animation and more stylized modern assistants.
- **True Z-ordering of windows:** In addition to translucency for creating depth, traditional window frames could also be push back in Z. Depth is a perceptual cue that is underutilized in today's windowing system. Proximity, as in the real world, could be used to establish priority or importance.
- **3D Space:** While immersive 3D space may be allusive in the three year time frame, the use of perspective to establish 'space' could be useful for collaborative work spaces within a given window. Artifacts such as calendars and agendas could be posted on walls, proximity of objects and people could suggest association and lighting and 3D assistants could become more useful and relevant.
- **Sound:** *Earcons* or small auditory feedback coupled with user-interface elements when done with reserve could provide addition feedback to the user. For example, using the frequency of a sound to give feedback while scrolling. As the scroll elevator approaches the top of the scroll bar the frequency increases. Like Win95 scroll elevators scale to the content so could the sound. Large scroll spaces get deep sounds while smaller ones are higher. The help system could speak cue card like instructions while you're working so that your eyes don't have to leave their context. It's like having an instructor over your shoulder. Sound could also be a big win for visually challenged. On command the cursor could 'emit' a sound and based on its location on the screen could send a different sound. Each side of the screen, window or dialog could have a different sound. As the user moves to the edge of the volume of that sound increases. This resembles the auditory cues we all receive as we navigate through spaces. As 3D sound technology gets better, we could give better spatial cues as well. When the phone rings the sound could come from the phone icon. So, if the phone icon is in the lower right corner, the sound could come from the lower right corner. With full 3D sound, sounds could even come from behind you. These techniques could also be coupled with the animation cues outlined above. To further establish the origin of a window or dialog, a sound could accompany the animation. This technique is especially useful in the animation of assistants.

MS6 5007127
CONFIDENTIAL

Support for Additional User Activities

- **Projects.** We need to do more CI work to understand this feature which was cut from Office 96. We will be investigating much lower technology solutions, such as more formal use of OLE properties to tag documents as project related.
- **Workspaces.** Related to projects in the sense that they are also folder-like containers, but CI clearly shows that users have a notion of physical grouping of "rooms" in addition to grouping all related tasks together in a project like environment.
- **Vertical markets:** Possibly we could provide add-on kits to customize Office user-interface and command sets for key vertical groups, such as home, small business, SOHO.

6.10 Content and Online (andrewk)

Office 95

Multiple efforts in our products can teach us more about this area, including:

- Word Viewer
- Word Internet Assistant
- Word Mail
- MSN Connection (mostly for support)
- Office/Bookshelf bundle

Given our very large market share, we should start convincing information providers that users will prefer the convenience of data that is "ready to use." We hope to:

- Set up an Office Compatible Information Providers program. Provide tools (e.g. Wizard builder?) and necessary expertise to encourage information providers to use the richer Office formats, which might in turn allow them to profit more from the "ready to use" information they publish.
- Start to better understand the potential of MSN-provided information, and how it could take advantage of the richness of Office formats. We should aggressively start to publish templates, data (with preset wizards, pivot tables, etc.), services (similar to our Deluxe printing feature in 95) and whatever might enhance the use of Office applications.
- Need to invest in building "on-line communities" for Office. Not just general discussion areas that tend to be dominated by a minority of users; but, rather forums for special interests (e.g. screenwriters that use Word). Office formats should begin to show up in non-software-devoted forums (e.g. Office business templates available from the small business discussion areas.) Investments in this now will teach us for the future.

Office 96

In 96, Word is leading the way with their Word Everywhere effort. We need to turn this into "Office Everywhere". Plans to create an Excel viewer and an enhanced PowerPoint viewer will bring us part of the way.²

VBA everywhere is a key component of the strategy. Delivering on the sea of objects means providers of expertise (e.g. performance review software, accounting packages, and all typical MIS projects) should just program on top of Office. The Office Compatible and Solution Provider programs should convince them that the time savings and enhanced power outweighs the concern over requiring Office.

² A related memo on Office documents on-line and on the Web is available from SteveSi.

**MS6 5007128
CONFIDENTIAL**

Also, crucial is Ren and the Office documents as forms strategy. Basically, our user research shows that user's don't distinguish information by its source (i.e. external vs. internal), but rather by its need/value. So, a salesman might have a dictionary (static, external information), vendor catalogs (dynamic external information), and a services price sheet (dynamic internal information) all within arm's reach if they are all frequently accessed. (Just like we have dictionaries, news, and feature lists all "close by".) Users will want to access and search information consistently. We're also investigating links that would allow a user set up the equivalent of a Word "home page" on their network which would allow browsing to other Word documents, or other Office document types.

Since it's crucial that we start developing expertise in this area, we should continue to push information in Office formats over the Internet and MSN. We should set up a special MSN Solution Provider program where you can order a custom template. We should (perhaps through the Office Small Business Pack) contract a number of compelling examples and learn where our limitations are. And, we should certainly start investing in International Content to learn more about the localization challenges of an information strategy.

Office 97

Polish: expect lots of fixing of viewers, VBA, forms, Ren, and Binders based on scenario feedback. Remote links to MSN (e.g. Excel sheets that recalculate based on current currency conversion rates or stock prices.)

Preview: Combine social UI with information access to get simple agents that assist you in searching for information and analyzing information across applications. Keep building sense of on-line community with features that promote information exchange between users.

Office 98

Office 98 must clearly be the best platform for authoring and viewing information. Additional reasons include:

- Advanced viewing and browsing taking advantage of new technology (e.g. multimedia, 3-D) that users will come to expect of computers as they begin to consume richer information (like Encarta and Cinemania) more regularly.
- Agents automate searches and analysis (AutoExecutiveSummary)
- More Microsoft information easily lands in Office format (everything from consumer titles to pivot tables analyzing the files on your hard drive)
- Binder Everywhere (the binder in Word Everywhere demo) is a reality, and is the same as a folder in Windows
- Close ties to OFS; unified access to all sources of information (i.e. single search UI--probably agent-based "fuzzy search")

6.11 Programmability (rwolf, tonyw)

Office 96 - Enabling Office as a Platform

The theme of 1996 is enabling Office as a development platform. In Office 96 all applications (except Ren) will provide an OLE automation object model and host a common Interactive Development Environment (IDE) including VBA and Forms³. These technologies provide the base level support for customizing applications and integrating the applications to form business solutions.

Here is a table of when various enabling technologies are first available:

MS6 5007129
CONFIDENTIAL

Product	OLE automation object model	VBA	IDE with Forms ³
Excel	94	94	96
Access	95	95	96
PowerPoint	95	96	96
Word	96	96	96
Ren	96	97	97

Table 5. VBA Enabling Technologies.

Common Integrated Development Environment

Office 96 will provide an Integrated Development Environment (IDE) that is common to all of the Office applications. Elements of the IDE include:

- code editor
- object browser
- forms design environment based upon Forms³
- project window
- properties sheet

An important goal of this IDE, other than that it is common to Office, is that it is similar to VB. This similarity leverages the popularity of the VB development environment and meets user requests for an IDE that works like VB. In fact, one way to think of the Office IDE is that it is "VB in-process". We are very excited about delivering on the vision that programming Office applications is truly a subset of programming in an enterprise client/server toolset. In particular, we view the scalability and reuse of the DD technologies for IDEs as critical.

The various elements of the IDE are contained within an MDI window that provides the IDE workspace. This MDI window is separate from the MDI window that hosts each application in Office 96. There is one such workspace per Office document.

The separation of Office applications from the IDE workspace is a loose integration model. Contrast this to the integration model of VBA and applications (Excel and Project) in the Office 4.0 time frame, where the VBA elements were integrated into the workspace of each application. The loose integration model has several benefits. It allows the IDE design to mimic and leverage VB. The bandwidth of the API between the apps and the IDE is less complex, making it easier to integrate the IDE with the apps. Since the IDE is decoupled from the apps it will be easier to upgrade the IDE in the future without disturbing the apps.

Office intends that the Office IDE share a code base with the next generation IDE provided by the Developer Division. First, our customers want a common IDE spanning Office to standalone development. Second, as an efficiency issue we believe that a single IDE code base will best allow Microsoft to deliver a common IDE. The Office IDE will include elements of the Office look and feel such as command bars.

Forms³

Office 96 will use Forms³ technology in the IDE forms designer and in the runtime for forms created by the IDE. Compared to the dialog editors incorporated in today's Office, Forms³ provides three major benefits. First, Forms³ is an advanced forms package with numerous additional features. Second, Forms³

allows us to provide a single dialog editor for all of Office. Finally, Forms³ is Microsoft's strategic forms technology, so Office will benefit from a continued stream of future enhancements and synergy with other products.

The Forms³ forms in the IDE live within the IDE workspace window. At runtime the Forms³ forms live within the workspace (MDI window) of the application.

VB style user model - code behind objects

Office 96 will adopt a common user model for the relationship between VBA customization and documents. Office will use the *code behind objects* model of VB.

The easiest way to explain this is with an example. Suppose the user selects a control on an Excel worksheet. The user then chooses the *show code* command from the Excel *Tools* menu and the IDE pops up showing the code that handles the events raised by the control. The user illusion is that the code is "behind" the control. Contrast this to the current Excel model where the code lives on another ply of the Excel workbook.

In reality, the entire VB project, including code modules and forms, is stored with the document. This storage model is much like Excel's current design, but is a change for Word with respect to Word Basic.

Common Object Model - qualified

Office 95 will provide a common object model that spans many aspects of the Office applications. The principal user benefit is that the user can transfer their knowledge of programming from one application to another application. Ideally, programming in Office should feel like programming one large application.

The VB language, its syntax and keywords, provides a beneficial degree of commonality, but it only covers a fraction of the knowledge that a user must employ to successfully program with Office. So standardizing the language is only a small part of the solution. Much more of a user's interaction with Office VB programming is with the object model than the language, so it is important to standardize the object model as well.

Since 3 of the 5 (including Ren) Office applications will have shipped an object model before Office 96, it will be difficult to achieve total commonality of object model within Office. Although we possess the technology to change the object model from year to year and still run user code written against an earlier object model, users will weigh the learning transfer across Office apps with the loss of learning transfer from one version of the individual products to another. Therefore, we will strive for the greatest possible consistency among Office apps balanced against the desirability of compatibility with previous versions of the application object models.

"IntelliBuilder"

IntelliBuilder is the only programmability component in Office 96 that is directed toward end user programming. The target for IntelliBuilder is the power user who thoroughly understands the product functionality and has a need to script portions of a solution. This user is about the level of an Excel or Access macro programmer, although they need not be familiar with a particular language syntax.

IntelliBuilder allows the user to string together a sequence of actions that are at a higher level than VBA language statements or object model references. This sequence makes up the IntelliBuilder "macro". IntelliBuilder presents the user with a choice of approximately 40 actions for each application, so the user need not browse a large space of possibilities.

IntelliBuilder then generates VBA, which is compiled and executes. If the user is familiar with VBA, they have access to the generated VBA, which the user can modify.

MS6 5007131
CONFIDENTIAL

Office Product Unit 1995 Three Year Plan

IntelliBuilder will be included within Access. Inclusion in the rest of the Office apps is probable, but is not the current plan of record.

Issues

The following issues are the major issues facing programmability for Office 96:

- What technology will the Developer Division provided IDE be based on, VB or VC++?. There are two separate groups in DD, each working on their own technology. Additionally, there is a task force in DD charged with evaluating the technology base upon which DD will build its future IDE. The Office requirements are clear - *Office would like to use the IDE that shares the code base with the future Ide of the Developer Division. We do not want to become dependent upon an orphaned technology.*
- The IDE is off to a late start and there are questions if it will be done in time. We will manage the dependencies and requirements we place on the DD team providing the IDE so it cuts features to meet the schedule.
- We are concerned about the delivery of the IDE, Forms³, and VBA on the Mac.
- The object models of most of the apps are partially complete or entirely complete and were designed independently. This will make unification of the object models difficult for the participants and imperfect in the final consistency we can achieve.
- The inclusion of Office components, such as Command Bars, in the IDE complicates the issue of compatibility and code sharing with the VB IDE. VB developers expect the ability to redistribute elements found in the VB IDE in their apps, and we will not allow redistribution of Office components without an Office license.
- We do not intend on making the distribution of the Office DLL(s) a commonplace occurrence. The user will obtain this code by purchasing Office. For the needs of some select internal customers, for example Bookshelf which ships in the Office box and stand-alone, we will provide alternate solutions. We do not want the Office DLL(s) to become another VBRUNx.DLL.

Office 97

In 1997 Office will continue the work of enabling solutions development and will begin to add more end user programming capability. As 1997 is a "minor" release we will not attempt major architectural work in this area, constraining the possible areas of work.

Building Solutions

Some of what we do in Office 97 will be polishing, or perhaps more appropriately, finishing the work of 1996. In Office 97 the Ren viewer will host VBA, completing the hosting of VBA in all the Office applications. We will add features cut from the Office 96 IDE, which quite probably will be a great many features as the development of the IDE is off to a late start.

We will pick up features added to the VB IDE, since we plan to leverage a common code base with the VB product. Of course, we may choose not to include all the features added to the VB IDE if they are not appropriate for the development of Office based applications. Two areas where we may expand our support for building solutions are the addition of cross app debugging and support for workgroup agents (programs that run on behalf of the user on a server).

End Users

In Office 97 we may add some support for end user programming. Working with DD, we will look at the development or acquisition of simple visual programming technology, such as Prodea Synergy.. We may add cross application recording, as recording is one of the only entrees to programming for end users.

Office 98

In Office 98 we will build upon our base of VBA enabled applications and begin to directly support building solutions. Much of this work is described in the *Business Solutions* section of this document . Here we describe additional programmability features for solution building. Office 98 is also the first year we may make an effort to genuinely support end user programming. We will also examine the packaging of Office as a development environment in a separate SKU.

Building Solutions

Possible programmability features for building solutions include:

- Task oriented code generation wizard for developers, much like the wizards in MFC.
- Cross machine debugging
- Break out more components of the Office applications. The idea is to follow the Excel Graph model, where rather than componentizing an entire app we break out a significant chunk as a component.
- Since we plan to move Office to the SDI model in 1998, the object model will change to reflect this. For example, the application object, a prominent fixture of our current object model, will not be appropriate or will radically change in an SDI world. Note that the Office 96 IDE design accommodates SDI, despite the fact that the IDE is an MDI.
- An important user issue for running solutions build with Office is the configuration management to insure that the users machine is appropriately configured to run the solution. Mismatched or missing Office components, differences in path setup or other personal options can prevent a solution from deployment. We will work with the configuration management team to make sure that our solution in that area is sufficiently robust.

Much of the work planned for Office 98 architecture area directly benefits the construction of solutions. In particular, the factoring or properties among the workspace, class, document, and user, the selection and command architecture, and in general the SDI work, help solutions development.

In Office 98 we may look at the issue of Office apps on servers, where the scalability of the applications is an issue.

End Users

In '98 we will make progress in making Office a more hospitable environment for end user programming. Possible programmability features for end users include:

- Visual programming
- Assistant integration
- Intentional programming

FL AG 0043515
CONFIDENTIAL

MS6 5007133
CONFIDENTIAL

6.12 Business Solutions (rwolf, tonyw)

Office 96 Enabling Office as a platform for solutions

Applications will host VBA, Forms², and the IDE. This provides the basic support for customizing applications and integrating them into business solutions. Beyond '96 we need to address more of the problems that customers face. Hard technical and logistical problems that solution providers encounter include:

- Performance of cross-application calls
- Resource consumption in multi-application scenarios, especially when a small part of the application is being used
- Bullet-proofing the solution: error handling, control of access to application UI and functionality
- Concurrency management, when multiple applications and/or multiple storage systems are used
- Configuration management for solutions: COGS, setup, version management of Office components
- End user training, when the solution contains custom UI and Office application UI

Integration of Office development environment with enterprise development environment: source code version management, coordinated debugging, build and deployment.

Office 97

We need to remove the purely logistical issues that are barriers to use of our applications in business solutions. These are in the areas of configuration, setup (for the solution app) and version management (for the Office components used in those applications).

Office 98

As described in the architecture sections above, there is much work to do around SDI issues and concurrency. We need to support apps running under programmatic control and user control at same time, managing reentrancy in apps and concurrent access to the data. For some Enterprise applications, we need to cooperate with transaction managers, using the applications as a front end to enterprise data. This means providing great support for binding to external data sources, and manipulating that data as easily as if it were local to the application.

In solution scenarios, particularly enterprise applications, the Office document types will be heavily customized. This needs to work well and efficiently. Office must allow for evolution of the solution apps, which encompasses schema management, integration with enterprise development environments etc. Most customizations of Office applications will be as part of a larger development that the customer is undertaking, and must be manageable under the same tools (e.g. source code version management system). Since Office will not be providing programming environments for (e.g.) AS/400, Office customizations must be controllable by other programming environments.

6.13 Upgradability and Administration (peggyst)

MS6 5007134
CONFIDENTIAL

Overview

Our work on Setup '96 is just beginning and a good deal of the work that we are doing today and planning for the next two years is reactive rather than proactive. In '96 we will re-architect the Setup engine in order to provide a solid structure for the many configuration changes that we expect in the coming years. For this reason Setup may be one cycle behind the rest of Office in terms of new features. The assistant

will be introduced in Office in '96 but will probably not play a roll in installation before '97. Making decisions for the user is going to be visible in Office '97 and may not be a design goal in Setup until the '98 time frame.

Our goal is first to make our application Setup the easiest, most flexible installation program available. When we have assured ourselves that this goal has been met we will concentrate on making Setup, hands down, the coolest feature of Office.

Areas of Concentration

Easy to Upgrade

The Setup feature will play a key role in the sales of the Office products in the next three years. As our market moves from new customers to the maintenance of existing accounts, the ease of migration is critical to justifying a software update. Often the cost involved in upgrading a large account is greater than the cost of the software itself. Upgrading existing applications will become practically invisible by Office '98.

Piracy

Currently almost half of our software is stolen. As we move away from the use of floppy disks and towards the use of on-line and CD technologies, we have increased opportunities to safeguard our products.

We have no specific answers to the piracy problem. In some ways these new technologies lend themselves to the prevention of casual software piracy and in some ways they make it more difficult than ever to implement control mechanisms. As we put more and more information on a CD and encourage users to run files off of the CD the user will be reluctant or unable to give away the CD for other people to use. However, unless we required the user to have the CD mounted at all times we would not be doing anything to prevent the CD from being passed around. Requiring the CD to run our apps would limit users ability to use other CD titles in conjunction with Office and would prevent mobility for laptop users. On-line services provide built in licensing/auditing particularly if the user does all installation and maintenance from the on-line service.

In '95 we have taken very small steps towards the reduction of piracy. In '96-'98 we can use the technologies provided in the system and Back Office to achieve much tighter control over distribution of our applications. The Configuration Management group will continue to explore creative solutions to the piracy issue in the next three years and beyond.

Administration

Over the next three years we need to examine the procedures used to make the deployment of Office software across large numbers of workstations easy and flexible. SMS will play an important roll in helping us to push installations down to workstations, but we have a large responsibility in making it easy for administrators to control the look of the final install. A Setup SDK will be key to providing the tools and documentation necessary for network administrators to quickly and safely deploy Office applications.

Customer Issues

As our products become larger and more complex we need to concentrate on making Setup simple for the end user, yet flexible for network administrators. Some of the major customer complaints/requests that we will be addressing in the next three years are included below:

- The ability to completely customize an installations, specifically when rolling out to a large number of workstations. This includes installing already customized apps or user-specific components. ('96)

Office Product Unit 1995 Three Year Plan

- An easy method for customizing the install process (editing the STF). ('96)
- Better reporting of Setup process (logging). ('95)
- The ability to choose between Drive letter and UNC. ('95/96)
- Set the default Program Manager group for application icons. ('96)
- The ability to update admin. installations in place and propagate to workstations. ('96/'97)
- One button installs: user is asked minimal questions and information is gathered from previous installs, available resources, etc. ('96/'97)
- Control over what format people save files in so administrators can auto set Save to old version during transition period. (Apps groups '96)

Office 96

Mission: Lower the cost of administering Office for both network and single user installations. Cover both ends of the installation spectrum: the administrator who wants complete flexibility and control over the installation of each component; and the user who wants the installation to happen quickly and without a lot of user interaction.

Design Goals

Modular Installations: Each component within Setup will be treated as an independent module. This enables the administrator to have more control over the end user configuration. This plug and play idea means components can be installed on their own, or as a part of a larger installation.

Flexibility: Give users the freedom to install any component to any location. This goal can only be achieved through cooperation with the Apps groups in the design of the components themselves. Our job will be to evangelize the importance of this functionality.

Admin. Tools: Provide tools and documentation to take advantage of the new flexibility. This will happen via the Setup SDK.

Core Code: Create core code environment for development of the Macintosh installations. Match Windows functionality as far as the Macintosh system supports its.

Reduce Size: Reduce the size of Setup in proportion to the rest of Office. Make Setup appropriate for smaller kits and Consumer products without reducing Office-related functionality.

Increase Speed: Reduce amount of time it takes for typical installation. Where speed cannot be enhanced, use creative methods to "pass the time" or make the time used less obvious to the user.

Software Distribution: Outside of just working with SMS we need to provide solutions for distribution of our software for the multiple workstation shops. Designing flexibility into our Applications, as well as Setup, is key to making this happen.

Key Technologies

- Work well with SMS: better integration/rollout
- Systems technology: there are a huge class of installation issues that could be handled by improved system support. Most importantly component registration and management are need to be independent of Setup and controlled/maintained by application working with the system.

MS6 5007136
CONFIDENTIAL

Office Product Unit 1995 Three Year Plan

- License management: Provide metering functionality (SMS dependent)
- Component customization/flexibility - push use of registration of components onto apps groups
- Auditing: in conjunction with licensing provide better feedback of product installation and use in corporate installations.
- User level profiling: Machine independent application customization based on user login, via user registration information (Windows/Cairo)

User Scenario

Otto is a network administrator at a pharmaceutical firm, Drugs 'R Us. He has a total of 1,000 workstations to setup with the Office software. Of those workstations he has four workgroup types: Finance, Marketing, Administration, and Product Development. Otto wants to customize the installation for each workgroup and there are some standard company Word templates, Access forms, and Excel macros that need to be installed for some of the workgroups. The Finance Group does not want Access on their machines since the majority of their work is done using Excel and they have several standard Excel macros that they use for generating financial reports.

First, Otto authors a small Setup table for the installation of the Excel macros using one of a variety of install templates provided in the SETUP SDK. Then he writes a short script, again based off of a supplied a template, which lists the components to install and the destination, on the server or on the local workstations, for each component. He includes a pointer to his Excel macro script so that the names and location of those files will be written to the local workstation registry. For the Marketing group Otto alters the install script to include some custom templates for sales reports. Otto uses SMS to distribute the software to all of the workstations using his customized scripts.

Several months later there is an upgrade to the Office components. Otto receives three disks containing all of the changes. When Otto installs to the Finance server Setup seamlessly updates the server in place and provides a log that outlines the changes that were made. Otto passes the update install script to SMS for distribution to the workstations.

Office 97

Mission: In the '97 time frame we need to continue our quest for easy upgrades and enhanced software deployment strategies. MSN will begin to play a key role as a distribution mechanism for our product installation and maintenance.

The design goals for meeting this mission include:

Code On Demand: Code on demand is the ability to have access to all features whether or not the necessary files are installed (registered) locally. The components can be installed *on-the-fly* as they are requested for use, rather than strictly at install time. Similar to the functionality that Windows has today, the files are copied locally only when the user chooses to use the given feature. The distinction between installed and uninstalled components becomes blurred.

Advertised Features: Components that are not installed locally can be advertised on the user's machine through an icon, dummy file, or pre-constructed list. This enables the code on demand idea to work.

Upgradability: Push forward with the idea of plug and play components and allow for some type of "drag and drop" upgrade. Setup may run as a background application, or all components could be self-registering.

Intelligent Installs: Make install decisions for the user based on existing profile and machine configuration, as well as previous installations and current software layout.

MS6 5007137
CONFIDENTIAL

Office Product Unit 1995 Three Year Plan

Piracy: Provide a mechanism for safeguarding the free distribution of our software. Concentrate on CD and On-line as primary media on which to develop blockades for copying applications.

Auditing: We need to provide our corporate accounts with solutions for metering the installation of software. We would like to use SMS as the solution for software auditing, but we may need to provide added support in Setup.

Social Assistance: Investigate the possibility of intelligent use of the assistant within Setup. This may be particularly appropriate for the installation of Consumer titles.

UI Customization: Provide an easy interface for providing user level UI controls. This would be used by internal Setup authors and network administrators.

Localization: Provide a core engine that will support easy localization specifically for BiDi and DBCS products. Provide an environment which could support multiple language installations in one package.

Key Technologies

- Microsoft Network
- SMS - transparent replication of software, automatic synchronization

User Scenario

Drugs R' Us purchases the Office '97 through Microsoft Network. Otto drags and drops the software from the Microsoft Network Server to his local server. The next day users are alerted to the updated software and Setup updates the local machines.

On her local machine, Patsy doesn't have a lot of free disk space. Setup intelligently determines the software that should get installed locally, and those items that should remain uninstalled, but advertised. For example, the 400 additional PowerPoint AVI files do not get installed automatically although PowerPoint retains a pointer to those files. When Patsy goes to use or look at any of those items, the file is dynamically installed on her machine. Patsy looks at several of the installed features and each time the component is installed locally.

During the year both Access and PowerPoint make some enhancements for running on the *septium* chip. As soon as Otto hears about this he investigates the availability of the software via MSN and sees that it will be available at the end of the month. As per the schedule, the software is updated on the Microsoft product server at the end of the month along with information as to the changes that occurred. Otto drags the new files down to his server and Setup automatically registers the file version information. Using SMS the file is updated to all licensed users.

Office 98

Mission: Subscription by taking full advantage of the Microsoft Network for installation and maintenance. Use the functionality within the network to do more than just enable easy, "on demand" installations and upgrades.

Design Goals

On-line everything: Use the network as major distribution mechanism for our software release and upgrades. Make it more attractive - faster, cheaper, easier - for users to install and maintain using on-line services.

Selling through Setup: Use the Microsoft Network technology to provide a medium for selling and advertising add-ons, new components, features, upgrades or other products. Rather than

waiting for the user to come get updated products, use MSN to prompt registered users when file/product changes are made available.

Multimedia: Use sound, video, and assistant animation in Setup where appropriate.

Discoverability: Provide a way for the user to learn about features at installation time. Interactive tutorials could be used to help the user make decisions about which features to install, or as a mechanism to demonstrate functionality as files are being copied in the background.

Component self-installation: Investigate the possibility of having each component include its own installation information. The onus of installation information could be pushed onto the component itself, in conjunction with the appropriate system environment (Cairo). This could only be achieved through the design of the applications themselves along with support by the system

User Scenario

Otto has quit his job at Drugs 'R Us and is now consulting out of his home. He has just purchased a new machine so he logs onto Microsoft Network to see what software is available and sees Office '98 with the Library of Congress reference volume. Otto chooses to install the executables locally but decides to leave all of the Library of Congress information along with the sample multimedia templates up on the MSN server to save some space on his machine.

Since Otto is a registered MSN users, Setup tailors the installation to his personal profile. During the installation a multimedia demonstration shows Otto the new features in Office '98 and how to use them. In addition he is presented with demos of several add-ons and other Microsoft products which are useful for small businesses.

Six months later, Word completes some performance enhancements on Word 9.0a. After the new version has gone through the normal Testing and QA cycle, Sheri drops the new files on the MSN product server. Otto receives a message alerting him that there is a new version of Word available along with a detailed description of the changes that have been made to the product. Otto elects to receive the updated version, which is copied down to his local machine, and his account is billed for the upgrade.

6.14 Using Exchange/Notes/Workgroup (rWolf)

Office 96

In Office 96 we will build upon the strategic Exchange integration work we have done in Office 95, especially WordMail and OLE property promotion. We will continue to exploit Notes integration as soon as possible as a defensive measure. Office 96 will be the first time that we ship PIM capabilities (Ren) as part of Office and we will integrate our document based applications with the Ren.

We are doing work in the collaborative arena that is not directly connected to groupware stores like Exchange Server or Notes. In Office 96 our apps will support multi-user editing of documents via reconciliation technology, which we will leverage for Exchange and Briefcase scenarios as well. Office Binders will be enhanced. Office will provide a consistent annotation facility, based primarily upon the shared Escher draw layer. Additionally, many of the document management features support workgroup scenarios.

Exchange integration

WordMail in Office 95 is based upon an OLE extension called DocObject that enables the replacement of the rich text editing capability in the mail send note with Word, providing all of the editing and formatting capability of Word. This feature is optional since Word together with mail exceed the minimal memory requirements. However, for users with additional memory this is an exciting feature.

Moreover, WordMail is highly strategic in the groupware competitive landscape. Lotus has declared their intention to base their desktop strategy on integration with the groupware backend, in their case Notes. WordMail puts us on the offensive by making the case that we understand integration of our Office apps with a groupware infrastructure better than Lotus and that our desktop apps will be integrated in a compelling manner when we ship Exchange.

OLE property promotion in Office 95 promotes document properties, such as author or keywords, to the Exchange store where they are available in views. This work was a joint effort of the Office and Exchange groups. Office writes the properties into a compound file stream and Exchange promotes those properties out of the stream into MAPI properties, where they are then available in Exchange client views. The properties are also available in Office File Open, which indexes files based upon these properties (as well as content), and in the Win 95 shell, which displays the properties on an individual file basis in a property sheet.

OLE property promotion serves a similar purpose to Notes F/X, but it has several advantages. Unlike Notes F/X, it is a passive technique that does not require running the application to promote the properties. Additionally, since OLE compound files and their accompanying property streams are likely to be supported by many apps in the Win 95 time frame, whereas Notes F/X requires support of the Notes API, it is likely that OLE property promotion will serve as the standard means of providing app property promotion, placing our apps on a more strategic long term path than the Lotus apps. Even Notes 4.0 will support OLE property promotion for apps that publish their properties in a compound file stream.

In Office 96 we will leverage the DocObject work in WordMail and the apps, especially Word, to provide Office documents as simple forms. Ren will provide an enhanced send note, called the Super Note, that can contain any DocObject server as a form. OLE property promotion will be used to communicate form fields in the documents to the views. This feature will enable the use of Office documents in certain important groupware scenarios, which leverages the strengths of Office documents and the market share of Office. It is important to note that Office documents in conjunction with this feature are not the forms strategy for Exchange. Word will look at easy adds to their current forms capability to work better in this manner.

We will also look carefully at any Notes Task Force recommendations regarding the role of Office.

Notes

NotesFlow allows a document that is part of a Notes form to be informed of state dependent commands the user can execute (i.e. "Route to Manager", "Approve", "Resolve"). The document just puts these commands on a menu and then delegates to Notes. We will support NotesFlow in Office 95 if we have the Notes 4.0 beta and sufficient technical details in time. If we do not add NotesFlow in Office 95, then we will add it in Office 96.

We should plan to add to Office 96 any other Notes 4.0 integration features that we do not yet know about. Lotus's intention to focus their app development on tighter integration of their apps with Notes implies that there will be more extensive integration technology forthcoming. Since we do not know what that is, we can not plan for it in detail.

Apps PIM integration

The presence of the Ren PIM in Office 96 provides us with an opportunity to integrate our documents with PIM functionality. Customer visits have shown us that customers view their documents as part of a larger continuum involved in managing their daily work life. We plan two features to integrate documents and the Ren PIM - *events in the Ren journal* and *annotations as tasks*

Events in the Ren Journal

The Ren Journal records time based information that would be of interest to the user, for example, phone conversations, meetings, appointments, as events. These events are then displayed in the Timeline view, allowing the user to examine the events which took place during a particular period of time.

Many of the recorded events are Ren specific, for example, completing a task or composing an email, and are recorded automatically by Ren. Other events, for example, making a phone call or having a conversation, require explicit user interaction because they are not automatically tracked by the computer.

Some of the most important events belong to the world of documents. Office applications will record events for: Open, New, Save, Save As, Print, Close, Send and Route.

Annotations as tasks

Annotations as tasks is an easy means for the user to associate a document, or parts of it, with a task that appears on the central PIM task list and also provides a quick access to the part of the document the task points to. This is important since many of the tasks and to-dos managed in a PIM are related to a document or parts of it.

Collaboration features

Multi-user editing and reconciliation

Word, Excel, and PowerPoint will provide multi-user editing, allowing multiple users to do away with file locking and work on the same document as seamlessly as possible. Multi-user editing will enhance the process of collaborative authoring by enabling concurrent editing. The underlying technology allows the individual apps to properly merge shared documents. The same technology also enables reconciliation for documents stored in Exchange. It also enables versioning and rollback features.

Office binders

Office binders will be enhanced to add additional printing features (pending milestone review). These features include the possible addition of print preview and consistent headers and footers across the binder sections.

In Office 96 Binders become the container for multiple documents on line (as seen in the Word Everywhere demo). More specifically, this means that the Binder adds the ability to display hierarchy in its scope pane and the Office documents export their sub document structure so that can appear in the Binder scope pane.

Annotations

Commenting documents is a popular workgroup activity, often done by paper and pen today, but an electronic means will become increasingly important. Excel has Cell Notes, and Word has Annotations and Revision marks, and PowerPoint has the Notes view. Office plans a consistent, workgroup aware facility for commenting documents for 96.

Office 97

Office 97 will focus on incremental improvements to the existing feature set. We should improve Office documents as forms by improving the enhanced send note and by making incremental opportunistic improvements to Word to make it a better forms editor. Office will continue to support Notes by matching Lotus's desktop applications in Notes interoperability, provided we have sufficient and timely information from Lotus. Additionally, we will make incremental improvements to our PIM integration. The document management team may leverage their indexing technology to provide rich views on FAT as plug in to the Windows Explorer.

Office 98

In Office 98 we have the opportunity to integrate our documents with groupware stores in a more seamless fashion. There are several challenges in this area.

Making our Office documents into finer grained documents is one of the most important steps we can take to better fit into a groupware store. We want to expose what are currently document sections to the store, so the sections can be displayed in the views. For example, consider a folder of PowerPoint presentations. It would be extremely useful to create views that view the folder by slide properties, for example, all the slides on a given topic, regardless of which presentation they are part of.

Along the lines of exposing finer granularity, Office documents should expose other objects such as annotations, so they can appear in views as well. This would allow annotations to appear both in place in the context of the document and in the view. It will be necessary to assign properties to these fine grained sections and objects, so they can properly be categorized in the views.

Going in the opposite direction from finer granularity, it would be useful to allow document sections to participate in a multi-document rollup for printing, much like the way in which documents participate in Binder printing in Office 95. This would allow users to print a single document by selecting sets of documents in a view.

We should continue to improve our apps, especially Word, so they are better forms and email editors. This could involve factoring them so the unneeded components can be left out and the remaining app is smaller. As previously mentioned, this work improves our solutions building story as well.

We should continue to expand the role of the Binder as a browser of multiple sets of documents by adding a scope pane that allows the browsing of documents stored in Exchange or OFS. At this point the Binder becomes a document with the additional ability to view its storage context in an optional scope pane. If Exchange adds a property that allows documents to retain their order by participating in an ordered set, we can support the Office 95 Binder scenarios entirely in Exchange.

We will need to fix our link support for documents stored in MAPI stores. First, MAPI stores need to support a suitable linking mechanism, such as monikers. Then we need to support that linking mechanism throughout the apps. If we intend to unify our links via monikers as described in the architecture section, then we need to make sure that our apps support monikers everywhere they use references, such as Excel cross workbook references.

Finally, it is worth noting that the SDI changes discussed in the architecture section improve the scenarios that depend upon the DocObject technology, such as WordMail and document forms. SDI eliminates the modality issues we currently encounter while trying to run an app as a DocObject server at the same time the same instance of the app is managing an open document window.

6.15 Conferencing (lucyp, rwolf)

Document Conferencing is the interaction of two or more users over a remote link with each focused on viewing and editing the same document in real time, possibly in conjunction with a separate audio or even video conference. PowerPoint is currently the primary driver of this functionality.

The top user scenarios for document conferencing are:

Multi-user editing

Document conferencing speeds up the process of reaching agreement on documents. Rather than repeatedly revising and communicating documents back and forth, users can reach an accord during a conference call.

Multi-user Presentation

The presenter delivers a presentation to the other conference members possibly leveraging a lightweight Office viewer. There is no editing of the actual document, but the users may interact with the runtime instance. This interaction could, for example, be as simple as allowing each person to use the John Madden drawing tool in slide show.

Whiteboard

Users share an electronic whiteboard upon which they type and sketch.

Helpdesk

The helpdesk guides users by stepping them through problems. This includes remote control of the PC.

Office 96

Document Conferencing in Office '96 will be comprised of the Win 95 provided app-sharing component and the integration of that component in the apps. The app-sharing component will be distributed via a Windows frosting pack release.

The app-sharing model involves two machines, one running an instance of an Office application and the app-sharing component and the other running only the app-sharing component. The app-sharing component provides a clone of the view of the application being shared, including toolbars, menus, borders, and, of course, document content. The app-sharing component works by sending graphical information in the form of bitmaps and metafiles from the machine hosting the app to all remote machines. The app-sharing component on the remote machine sends user input events, such as mouse movement, to the machine hosting the app.

This basic level of document conferencing is available to any application since it is provided Win 95. Office '96 will invest a small amount of effort to provide integration of the app-sharing component directly into Office to close some holes which exist with complete reliance on the component as well as make Office a little more integrated with Windows.

While each Office component will implement the app-sharing model, PowerPoint will additionally provide the native app conferencing which is present in PowerPoint '95. The app-sharing model presents particular problems to a presentation app. Complex graphics and complex slide backgrounds (such as graduated fills) must be sent as bitmaps or metafiles, which is inefficient. Builds and animation must be sent as continuous updates in the form of bitmaps or metafiles, adding enormously to the inefficiency. These elements are more and more a part of modern presentations.

The PowerPoint native app implementation leverages RPC and OLE Automation to manage communication channels and drive remote instances of PowerPoint, each of which manipulates a local copy of the document. It accommodates presentation sharing and unique views of a presentation on different machines; it does not provide app sharing among those machines. PowerPoint '96 will clean up the shortcomings in the previous release, integrate the T.120 compliant DataBeam stuff just acquired for Windows, and add some minor new functionality.

Office 97

Office '96 document conferencing features minimal integration of Windows's conferencing. Office '97/'98 will enhance the minimal integration of Win 95 conferencing capabilities and possibly provide native application conferencing.

The most visible enhancement to the use of the app-sharing form of conferencing in Office 95 would be to fix remote printing. The problem is that since the app runs on a host machine, it is not possible to print the document from a remote machine.

Native app conferencing is a different model than the app-sharing model. Native app conferencing, like the PowerPoint implementation in Office 95, leverages RPC and OLE Automation to drive remote instances of the app, each of which manipulates a local copy of the document. It has important performance advantages and is the only form of document conferencing feasible for remote presentations.

Configuration management is a pre-requisite for attempting this form of document conferencing, as it is essential to be able to guarantee a known configuration of the app on the remote machines. If the configuration management work is not far enough along, we may have to attempt this in Office 98. Other than configuration management, this feature does not appear to depend upon infrastructure changes, so it could be provided in Office 97. Of course, the architectural changes contemplated for Office 98, such as SDI, will help document conferencing by decoupling individual document instances from the app.

Additionally, it would be useful to explore the use of PowerPoint as an advanced white board tool. Much of the native functionality of PowerPoint, such as outlining, drawing, and organization charts, would be useful in an advanced whiteboard facility.

Office 98

Office 98 will focus on enhancing and integrating conferencing into the groupware structure. If we do not implement the native app form of conferencing in Office 97 then we should do native app conferencing in Office 98.

Since the Exchange client (Ren) will be seen as the central groupware facility, it will be important to integrate conferencing into the Exchange client. Ren should be the control center for conferencing. Conferences should start and end in Ren and we should unify meeting setup in Ren with conference setup. If we do not implement native app conferencing in Office 97, we may choose to do the Ren integration work in Office 97 instead.

We should facilitate transitions into and out of conferences from other scenarios. For example, we should make it easy to set up a conference with the participants of a mail or public folder thread or with the editors of a multi-user document. We should make it easy to save a conference, including the participants, documents, and other state.

The app-sharing form of conferencing suffers from an the problem of requiring the initial download of the document before the conference can begin. We should attack this problem, possibly by changing the manner in which we load documents in the apps or by changing the granularity of documents. It may be possible to brute force this problem by restricting the domain of collaboration within a document.

Other areas to explore include:

- Adding a human dimension to conferencing, for example, displaying a scanned photograph of the conference participants;
- Adding addressability to the all apps, so it would be easy to talk about line numbers in Word, for example;
- Adding the ability to break out sub-conferences.

6.16 Document Management (stevebr)

There are two strategic areas this group will drive in the next three years: document management and corporate publishing:

Document Management

At one time a niche market, "document management" is rapidly evolving into an important component in the office computing environment. Customers were once satisfied (although not happy) using the products

MS6 5007144
CONFIDENTIAL

Office Product Unit 1995 Three Year Plan

of small ISVs who integrated their solutions into the mainstream office products using macros and other general extensibility techniques. But with the new perceived importance of document management in the corporate environment, the success of office software is becoming dependent on how well it is integrated with document management solutions.

With the Word Perfect/Soft Solutions merger, we can expect document management to be integrated into the Perfect Suite line more so than it already is. Lotus is in a position to exploit Notes as it's document management store, and has already taken steps to do so. Microsoft is behind in this category, only now responding to WordPerfect's Quick Finder technology. Integrated document management will be increasingly more important in corporate evaluations, and Office could lose sales as a result. This has already occurred in the legal sector (although document management, is not the only missing feature).

The immediate need is to enable tighter integration of third-party document management systems into Office, and to provide simple document management functionality for mainstream users without installation and maintenance hassles. We will do this in '96.

Once established, we can leverage BackOffice technology to provide more robust and scalable document management capability in the Office box. Also, we can leverage Forms³ capability to add user-defined property sheets and query-by-form. And we can tighten our VBA integration to enable codifying business process including mail-integrated workflow. We should also keep tabs on the acceptance of industry document management APIs to avoid being left behind as we were with ODMA. This is our '98 plan.

Online and Corporate Publishing

With the recent explosion of networking, both internal networks and the Internet, corporations are publishing increasingly large and complex document sets on networks rather than distributing them on paper. Often authored collaboratively, and often rich with links between them, creating and maintaining these document sets is currently a difficult task. We use the term corporate publishing to include any person(s) that wish to publish on-line information.

Office can provide tools to aid corporate publishing. Already, the applications groups are thinking about small-document publishing, with features such as viewers, hyper-links, and document loading and display optimized for the network.

This group can help by providing the tools for large-document publishing. Authors need collaborative tools such as already planned for document management. But they also need store-wide versioning, publishing, and history, support for the creation and maintenance of links, and document set copying and mailing services. This is an area we know less about, and so there will be other features that will need to be considered.

Some support for small-document publishing is going into the applications in '96. In '97 (or possibly '96), we will add better support for links, leveraging off of our content-index technology. Finally, in '98, we will add all the store-wide support for versioning, history, and copying/mailing.

Technology

This 3-year plan leverages existing technology, some of which was built within this group. Here is a list:

- **Content-indexing:** Our content-index is the basis for our FAT-based document management store. In '95, it is simply a content-index. In '96, it will also include whole-property-indexing, enabling fast property display, categorization, and ordering on FAT. We will then use it as the basis for our link-tracking in '97 (possibly adding some rudimentary link-tracking in '96). We will also supply our content-index to JET in '96.
- **OFS/Exchange:** Our scalable document management and corporate publishing solutions will, of course, be based off of Exchange in '98. We assume Exchange will be ported to OFS by that time. Server-side code will ensure a fully secure and robust document management system.

FL AG 0043527
CONFIDENTIAL

Office Product Unit 1995 Three Year Plan

- **OLE-DB:** OLE-DB is the common interface through which we will access browsing features on FAT, MAPI, and OFS. OLE-DB provides dynamic view (cursor) support including categorization, and specifies a query language flexible enough to handle these different environments. Although this group will not be in the business of defining APIs, we plan to extend OLE-DB to include document management verbs. OLE-DB will also enable easier plug-in of third-party APIs and document management stores such as DEN or Shamrock.
- **Forms³:** In '98, the document management system will enhance support for properties by storing property pick lists (and "required" lists) and user-defined property sheets in stores, and by supporting query-by-form. The Forms³ technology will enable this.
- **OLE:** Properties and content, of course, are stored in OLE compound files. In addition, hyperlinks will be OLE objects (monikers).
- **Common User Interface:** Document Management provides the common user-interface for the File Open and File New dialogs. These dialogs are a rich extension to the Windows common dialogs and represent a competitive area for productivity applications. This technology relies on Windows 95shell APIs.
- **Source Safe and Repository:** Although not currently fully developed in this plan, it is our intent to explore integration and sharing with the work being done by Source Safe and the Repository teams in the Developer Division.

Office 96

Mission: To integrate and make accessible simple document management functionality into the Office product line. In addition, we will supply core technology we develop to other groups in Microsoft.

- Provide a simple, easy-to-install, file sharing-based document management solution in the Office box. It will be based on FAT and compatible file systems, will work either locally or on file servers, and will be completely integrated in the Office product line. We will include some rudimentary support for routing.
- Provide access to third-party document management products through an industry standard API (ODMA, Open Document Management API). This API should replace macros as the mechanism used by third parties for integration. The API is a small set of abstractions that cover the checkin/checkout metaphor.
- Enable rich browsing of documents by property, including fast property categorization and ordering. This capability will be built off of our content-index technology.
- Make content-based document searching more useful through more advanced and interactive search techniques.
- Advance the File Open/Save dialog, adding miscellaneous usability features and filling in missing Windows '95 functionality.
- Provide access to MAPI and OFS file systems from the Office products including the Office Explorer (Ren) and File Open.
- Provide core technology (content-indexing and viewing functionality), which we develop to meet our other goals, to Access (Jet) and perhaps other groups at Microsoft. This includes packaging and support (not just throwing code over the wall).

Microsoft Confidential

-49-

FL AG 0043528
CONFIDENTIAL

MS6 5007146
CONFIDENTIAL

March 1995

HIGHLY CONFIDENTIAL

Office 97

Mission: To enhance in simple ways the document publishing scenario, giving users a taste of document publishing features we will be implementing in '98. We will also use this year as an opportunity to plug any obvious holes in our document management functionality and respond competitively.

- If needed, implement an OLE-based hyper-link moniker.
- Include link-tracking in our content-index. Support links as a query term (which can be mixed with other terms). Also include simple support for link replacement. Aid the applications in including a simple search-based link-creation tool.
- Depending on the Windows NT development schedule, we may ship our support for FAT browsing in Windows in '97 rather than '96.
- If DEN support becomes important, we can do this in '97. DEN, Document Enabled Networking, is an API being developed by Novell, X-Soft, Documentum, and others. It is a more grounded version of Shamrock (another consortium API consisting of Microsoft, IBM and others), but is still an abstraction that covers a wide array of Microsoft APIs such as storage, configuration management, etc.

Office 98

Mission: Expand our in-house document management functionality to scale to an enterprise solution based on BackOffice. Support fully large-document publishing creation and maintenance. Enable automation of business process including workflow.

- Using server-side code with additional client-side UI support, build a robust and secure document management system based on OFS. We will include a front-end to NT security, and support for archival and replication.
- Add support in a shell (Windows or Office?) for administration of document stores.
- Enhance property support by storing property pick/required lists in stores, and supporting creation and use of property sheets and query-by-form. Integrate Forms³ technology.
- Support store-wide versioning and history on both our FAT- and OFS-based document management stores. Support store- publishing, and browsing previous store versions (including hyper-link traversal to past versions). To the extent possible, enable this in the face of replication.
- Support links more thoroughly, including copying and routing documents with links, and store-wide copying based on links.
- Add support for client- and server-based document search agents.
- Add thorough VBA support to enable business process automation.

7. Summary of BackOffice Synergy

This section contains a summary of the support that the Office product will have for BackOffice.

7.1 SMS

We're actively pursuing possible integration work. Office is ahead of the competition in many respects, though because of the competitive needs of the SMS group there are some potential liabilities (e.g., SMS wanted to provide even richer administrator setup than we supported in Office 4.3 so SMS reverse

engineers the setup script, rather than having us address these needs in ACME directly.) There are some things we are not currently doing in 1.0 such as having SNMP-level agents (Standard Network Management Protocol—a proposed industry standard). There are disagreements over how well defined these are for apps.

We use and document a large number of registry settings so that things can be remotely administered. We actively pursue this and include this in our standard specification reviews. For 96, the big goal will be to provide a whole range of features we call "run from CD" which really means the ability to locate any part of Office code anywhere and we'll find it, see page 39. In particular, we want split local/network setups and the ability to tolerate dynamic server locations (i.e. \products1 and \products2) for load balancing.

7.2 SQL Server

Excel is doing the most here by moving from ODBC to OLEDB. Excel is also hoping to be an OLEDB provider which means that an OLE DB viewer (i.e. Access) could look at spreadsheets.

Access will have replication from local MDB files to SQL server data this summer, after 95 ships though. Many Notes replication scenarios can be covered by having structured replication from the laptop to the business running SQLServer. This is, arguably, the #1 missing in Exchange.

7.3 OFS

We are hoping to use OFS as the backend for our server based document management. One thing that is hurting our performance today is the use of multiple streams in compound files. We avoid this at all costs, yet if we used this we could get some very nice OFS features such as per-stream security, which translates into Notes features like security on a per-section of a document. As we get performance gains from OLE in compound file performance we will investigate this implementation, or if for Office 98 we can assume an OFS client.

7.4 DS

Directory Service is going to be important for us. Our goal here will be to use this for admin. features for maintaining per-user profile information so that a user can move between workstations and get the Office settings that they like. We get a lot of this already with NT, and it should just get better.

7.5 Cairo

We aren't doing anything here except assuming that it is a true superset of Windows 95, though the above use of OFS, DS, and the Windows 95 shell work are really the core components of Cairo until we learn more. As far as DCOM and distributed OLE things, we do not have anything planned. OLE Automation over a distributed net should just work.

7.6 Exchange 95

- **WordMail/DocMail:** We worked with Exchange client to get the performance for this killer feature done so it is reasonable on a 12MB machine (note this is minimum, and that the Exchange client is an 8MB machine app). This replaces the normal Send/Read note in Exchange client with a word document as per demo.
- **Post to Exchange:** This feature saves you the trouble of saving a document and then dragging it into a public folder. Instead of File Save you do File Post and you pick from a list of public folders. We then put the document in what is called a "free doc" (or transparent envelope) which is really just a blank message that holds the document and is transparent to the user. At this time all the properties are promoted (see below). This is an Office 95 feature. Doing this work for

MS6 5007148
CONFIDENTIAL

Notes is hard since Notes does not really have a straight forward idea of a table like MAPI does and instead of folders (MAPI tables) has views and forms, neither of which makes total sense for this feature, since in order to add a record you need to invoke the code (@ formulae or LotusScript) behind that view or form.

- **Compose:** This is a small delta off of WordMail since it lets you send a message to a BBS-like folder in Exchange. The note is a little different since the subject gets decided by the conversation thread and there aren't to/from/cc fields really.
- **OLE properties:** We view OLE properties as one of the key enabling workgroup technologies in Office documents. The only way to come up with interesting views without using Electronic Forms Designer in Exchange is to have a public folder with documents that promote interesting properties automatically. We put OLE props in a special compound file stream that Exchange picks up when you drag a document into a folder (see above). This is really a Notes FX on steroids, which is something we support.
- **Schedule+:** We will put buttons on MOM to create appointments/To-Dos/contacts. Word also did implemented mailmerge the S+ contact manager and Personal Address Book (PAB). We really need systems to come up with a PAB that will be the one place to store these things or a better architecture that allows us to use multiple PABs.
- **NotesFlow:** This allows a document that is part of a Notes form to be informed of state dependent commands the user can execute (i.e. "Route to Manager", "Approve", "Resolve"). The document just puts these commands on a menu and then delegates to Notes. This is something similar to what we did with DocObjects. We want to support this in 95 but if we don't get the beta very soon it will be impossible. We think this is an interesting model of developing a feature for the server that is somewhat easy for the client applications to implement. Exchange should review this and propose something that is similarly useful and easy to implement.

7.7 Exchange Post-95

- **OfficeForms:** This represents the work of rwolf, mikeang, mikemat, billbl, and a few other. OfficeForms (aka the Ren SuperNote) can be thought of as an improved WordMail. Basically, abstractly, we will create a generic notion of a container that will hold an Office document. The apps will then do work to do a good job being contained as a message. For example, Word will expose events like OnSend, OnOpen, OnReply, etc. so that special code can run at this time. The scenarios that this is interesting for are for some loosely structured exchange of information (brainstorming) or using Excel for complex highly structured things (expense report). This does not make Word compete with the Notes note., but makes Office as a suite of cooperating applications very compelling on higher end client machines.
- We will probably need to come up, working with BSD, with something like NotesFlow. This shouldn't be hard, but without having a language for driving forms on the sever this could be really tough to do something meaningful.

7.8 Client-Server Scenario

MS6 5007149
CONFIDENTIAL

The following is a brief scenario that we will aim to implement in the Office 96 time frame that will leverage BackOffice technologies. This is still under development.

The scenario is an automatic roll-up of a loosely structured status report. Each week automatic reminders go out via email to the department managers (a server based app runs on schedule using information in the directory service or SQL server HR database). In the message is a word wizard/template that steps through the goals from the previous status report (stored in Exchange) and asks the status. The user then just uses word to add comments. The message is sent back to a public folder. A server based app runs on

schedule to grab all the reports out of exchange and runs a Word macro that glues the reports together and AutoFormats them without intervention. The report then gets routed through Exchange for approval, using annotations/revision marks. After the last stop it is put in a public folder and an announcement is mailed out. Pretty high tech and very custom/SP oriented, but that's how workflow is.

This leverages the fact that we can run our apps on the server unattended and our programmability. On the Exchange side this forces the need for VBA access in a simple way to the message store. It also assumes the Cairo scheduler for doing tasks (something in SQL 95).

As another example, it is easy to see how we could also produce an Access report/pivot table on some SQL data on schedule, though this is super simple since it just requires the scheduler. For example, this is the weekly sales results out to the field.

8. Office Compatible (jimco)

8.1 Overview

During the next three years, Office Compatible's strategic goal is to help entrench Office everywhere. We will use four primary tactics to achieve this goal.

- First, Office Compatible will establish a large family of complementary, products that work as Office does, so customers will see that Office is actually a Super Suite which includes many of their most valued third-party products.
- Second, Office Compatible will establish Office as the indispensable technology hub of the desktop which adds real value to many other products (including mission critical products) via the technology it shares with them. Customers will say, "I live and die by my accounting software, so I buy Office, because it makes that software better."
- Third, Office Compatible will help leverage Office's success to increase greater success for BackOffice. We will work with Office and BackOffice program management to identify strategies by which Office's functionality can be enhanced when BackOffice is present, and we will share these enhancements via Office Compatible with third-party vendors. Customers will say, (1) the Office UI is the natural front end for good BackOffice clients, (2) Office is an even better product when BackOffice is present, (3) my mission critical third-party products are better when both Office and BackOffice are present.

The ultimate Office Compatible message will be that Office enriches the desktop in a way that goes way beyond words, numbers, and pictures. Indeed, Office is indispensable.

8.2 Sharing Future Office UI Innovations

To implement its first tactic, Office UI innovations must be shared in a timely fashion with complementary products. In most cases, this sharing should occur during the same development cycle when Office introduces them. Our goal should be to "sim ship" 25 great Office Compatible products that share Office's most exciting new features. But of course, it almost goes without saying that these features must be carefully selected so that sharing them adds value to Office. For example, many of Office's standard UI elements will evolve in the future: new menu entries will become standard, current buttons may be relocated on the toolbar and new buttons will be added, accelerator keys which do not exist today may become central to the customers work in just a few years. These basic UI elements must be shared via Office Compatible to ensure that Office is the natural and seamless core for cross-application tasks.

FL AG 0043532
CONFIDENTIAL

MS6 5007150
CONFIDENTIAL

8.3 Sharing Future Office Technology

Sharing Office's technology with complementary products is an essential ingredient in the goal to entrench Office on the desktop. Note that this is not a charity program: we will not give our technology to other vendors in some kind-hearted gesture. On the contrary, shared Office technology can be used by Office Compatible products if and only if Office is present on the desktop. So Office's enriches products that use its technology. The technology which is shared via Office Compatible must be carefully selected to assure that an Office win results. We should not reflexively share future technology just because it is in Office's DLL. However, we should not hesitate to share technology when that will significantly increase Office's appeal. For example, Office 96's new Command Bars should be shared in '96 with Office Compatible products, because this actually reinforces the appeal of Office. From one perspective, sharing this technology establishes Office as a revolutionary force that alters the way work is done on the desktop.....not just within Office but among all of the customer's most important products. From another perspective, failing to share this technology could actually undermine Office 96. We know that customers demand cross application consistency, because inconsistency slows them down and reduces their productivity. So by significantly altering the way work is done (as Command Bars will), we actually expose Office to risk by making it harder to use with other important products which continue to use prior-generation features. So introducing Command Bars in Office makes it easier to use alone, but sharing them with complementary products makes it easier to use with other apps. And to an increasing extent, ease of use with other apps is a crucial factor affecting the customers software preferences.

8.4 Creating Office - BackOffice Synergy

In the future, the Office Compatible staff must become a source for practical and innovative strategies for creating synergy between Office and BackOffice. For example, many customers use Excel to maintain long lists, but if BackOffice is present, SQL Server is a more natural repository for this kind of information. So, anticipating the needs of Office customers who later buy BackOffice, Office Compatible should encourage the development of macros (etc.) which easily export Excel lists to SQL Server. In other words, once BackOffice is present, SQL Server should be used as the data repository, and Excel should be used as the client-side front end and as the analytical engine. In addition, these "points of synergy" between Office and BackOffice should generally be shared with Office Compatible to add value to them, too. Thus, customers who are considering buying BackOffice should understand that it increases the power of Office and of their other Office Compatible products, too.

8.5 Office Compatible Implications for Office Program Management

Office Program Management should actively look for opportunities to introduce or refine features in a manner that creates synergy with third party products, and they should actively avoid designs which are difficult to share with others. Every Office (and Office component application) specification should contain an Office Compatible section where the author addresses the potential implications for Office Compatible. For example, if a program manager proposes that the location of the File New toolbar button should be moved, the spec should address the fact that this will "break" consistency with Office Compatible products, and a note should be made to include the revisions in the Office Compatible program. Similarly, when program managers design revolutionary features such as the Answer Wizard, they should analyze the potential benefits (and costs) of sharing the feature via Office Compatible. Whenever appropriate, features should be designed so that sharing is easier rather than harder.

8.6 Future Office Compatible Implications for Office Development

When Office developers write code, they should be mindful that it may be used by Office Compatible products (via DLL sharing, etc.), so they should write their code in a robust and highly maintainable fashion. In many cases, writing sharable code does not cost much more...as long as the need to share the

FL AG 0043533
CONFIDENTIAL

code is understood at the outset. All that is needed is to set developers' frame of reference so they understand that in some cases they are not just writing for Office, they are writing for the world.

8.7 Future Office Compatible Implications for Office Testing

We have already seen that the Office Compatible program can benefit Office testing. For example, to test the new Office Compatible DLL (MSOC95.DLL), testers created scripts which found no bugs in the DLL itself...but they did find several, previously undetected serious bugs in Office. In addition to this kind of fortuitous benefit, Office testing should place a greater emphasis on "interoperability testing" in the future. We must recognize that on average our customers use Office with nine other products from third-party vendors, and if these products do not work well together, then Office is perceived to be part of the problem. For example, if interoperability between Office and PeachTree Accounting's software package is crucial to the customers business, then the next version of Office must not break this interoperability. And when Office testing designs its interoperability tests, which products matter most: Office Compatible products, of course.

8.8 Plans & Deliverables

In 1995, Office Compatible will deliver a detailed style guide that shows vendors how to implement many basic Office's features, including menus, accelerator keys, toolbars, Wizards, Tip of the Day, Scrollbar Thumbtips, etc. In addition, we will spec the API's in MSOC95.DLL, which will let Office Compatible products access Office's document properties technology (including the UT). We will supplement this information with: an Office graphics library, extensive information and tools related to Office's UA strategy, VC++ code, on-line documentation, and sample apps, and a VB 4.0 add-in Office Compatible Wizard (plus sample code and documentation).

8.9 Plans & Deliverables

The 1996 Office Compatible deliverables will parallel those of '95: a basic UI style guide, API's for calling selected Office technology (via MSOC96.DLL), and development tools as appropriate. However, we will persuade the MSVC and VB teams to build the basic facilities for creating Office Compatible applications within their products. So our tools offerings will be restricted to making shared Office technology more accessible.

9. Competition

This section is not a complete competitive analysis of our competitors, but an overview of the major threats for the Office 96 release. Additional information is available from SteveBu.

9.1 Lotus

Lotus will release a major upgrade to SmartSuite this summer. The major concern we have is in the categories this release has the potential to be very major. As shown in the following table, Lotus has not released a major upgrade for most of the applications in quite some time.

Product	Released	Contents	Notes
SmartSuite 2.0	6/93	1-2-3 Rel. 4.0 Ami Pro 3.01 Freelance 2.01 Organizer 1.1 cc:Mail client	Ami Pro already 12 months old

FL AG 0043534
CONFIDENTIAL

MS6 5007152
CONFIDENTIAL

Office Product Unit 1995 Three Year Plan

Product	Released	Contents	Notes
SmartSuite 2.1	9/93	1-2-3 Rel. 4.0 Ami Pro 3.01 Freelance 2.01 Organizer 1.1 Approach 2.1	Removed cc:Mail client
SmartSuite 3.0	8/94	1-2-3 Rel. 5.0 Ami Pro 3.1 Freelance 2.1 Organizer 1.1 Approach 3.0	Only 1-2-3 and Approach upgraded significantly
SmartSuite 3.1	1/95	1-2-3 Rel. 5.0 Ami Pro 3.1 Freelance 2.1 Organizer 2.0 Approach 3.0	Added Organizer 2.0, but had stability problems and had to do a quick fix
SmartSuite 4.0 (Speculation)	Summer	1-2-3 Rel. 6.0 Ami Pro 4.0 Freelance 3.0 Organizer 2.0 (?) Approach 4.0 (?)	16 bit, Windows 95, and OS/2 planned (1-2-3 will be on 32 bit platforms only)

Table 6. History of SmartSuite releases.

We have reason to believe that we are at risk in category features when compared to our Office 95 products. The areas of overall concern for the suite category include the following:

LotusScript

Lotus will have LotusScript implemented consistently in all of their 32 bit applications. This will include a common language, debugger, development environment, and object model. LS has an optimized intra-application object model that makes scripting very fast for the single application case. They will also support OLE automation, though their automation object model may or may not be the same or as rich as their *product object* model. We do not know if LS will support cross application recording.

Notes

Notes remains our biggest risk. To date Lotus has implemented rather trivial extensions to their applications which were able to do in a few days worth of work. The most recent extensions, NotesFlow, are a small amount of work but obtaining the necessary information to implement the feature has proven difficult. In subsequent releases of SmartSuite we are concerned that Lotus will implement completely closed connections between Notes and SmartSuite applications. For example, Notes might choose to do the work to make Ami Pro a plug compatible Notes form. We are committed to doing the work we are given specifications to.

Cross-platform

The current Notes applications are all Windows 3.1 based. We have been told that the next major release of all the applications is based on core code that will enable them to move between Windows 3.x, Win32 (NT and Windows 95), as well as OS/2. There is no Mac development any more, however, the next Notes release will bring their Notes client to full status. There is a high risk to the "SmartSuite everywhere and Lotus is platform neutral" message that Lotus is sure to use. The fact that they will have 16 bit releases (except for 1-2-3) this summer is of some concern. In general, however, the Lotus Development Corporation cross-platform message is filled with holes and in the long term it is questionable if they can maintain it.

Performance

SmartSuite 3.0 is very small and fast compared to Office 4.3. The primary reason for this is not their development skills, but the fact that most of the applications are 2-3 years old and are far behind in features. Nevertheless, there is concern that the SmartSuite applications will be perceived as applications for the mid-range hardware. We will address these concerns to some degree in Office 95, especially because we can take advantage of the improved resource utilization in Windows.

Innovation

Lotus has a proven track record for popular and useful innovation. This will continue and we will need to be prepared to track this. In the suite category three innovations in the current SmartSuite have proven to be a problem area for us. InfoBoxes are a user-interface element that are modeless dialogs that enable the fast changing of properties for objects. The SmartSuite applications have done a very good job on templates, especially in Approach and 1-2-3. The addition of ScreenCam also caught us off guard and is a useful application.

Organizer

Organizer is a well-liked PIM and with version 2.0 provides some level of group scheduling. The compelling visuals, especially when compared to Schedule+, and the popularity of the product within the PIM category is a problem area for us.

9.2 Novell/WordPerfect

The current suite of applications in PerfectOffice are world class and there is reason for us to follow the progress of this suite very carefully, especially given the strength of Novell's networking and sales force.

The following table lists the contents of the PerfectOffice Professional product, which is distributed on CD ROM.

Application	Version	Standard Edition
WordPerfect	6.1	Yes
Quatro Pro	6.0	Yes
Presentations	3.0	Yes
InfoCentral	1.1	Yes
Envoy	1.0	No
GroupWise	4.1 (client)	No
Paradox	5.0	No
AppWare (Visual AppBuilder)	1.1	No

Table 7. Novell PerfectOffice contents.

Copy Microsoft Strategy

We need to be wary of the fact that Novell is executing a quick turn-around clone strategy for our key marketing features, from PerfectSense to QuickCorrect to OLE 2.0 support to their PerfectOffice developers' program they are stride for stride cloning our high visibility strategy. The important thing for us to do here is to insure that our features have sufficient depth so that they are not easily cloned in a minor update.

FL AG 0043536
CONFIDENTIAL

MS6 5007154
CONFIDENTIAL

Programmability

PerfectOffice programmability is geared much more towards the average end-user and emphasizes the cross application record-and-play scenario. For Office 96 we will be at a deficit in this area, though we will be investigating other technologies (see above). The inclusion of AppWare (formerly Visual AppBuilder) in PerfectOffice professional does not seem to have impacted the category.

GroupWise and Leveraging NetWare

The GroupWise client is very nice and will offer significant competition to Ren in the time, contact, and appointment management space. The significant risk to the entire Office product is the leverage PerfectOffice could obtain from tight synergy with NetWare. In particular the use of the NetWare 4.0 directory services could make PerfectOffice a great network citizen. For example, the extensible NDS could be used to store user profile information such that applications could be run off near-diskless workstations or off of any machine on the network. Also, we expect Novell to push the value of using a NetWare server as a PerfectOffice application server. That is run the server with network installations of all the applications and administer them as servers rather than workstations.

Box of bits and perceived value

The current trend for PerfectOffice is the inclusion of any and all "bits" that are available. In particular the use of Envoy and AppWare show that Novell is eager to increase the perceived value of the suite. With the potential of the CD distribution of user-defined suites, there could be pressure for Microsoft to do the same. Technically, we need to be aware of the possibility of including other applications or services that could redefine the contents of the suite category.

9.3 Competition: Non-Suite Products

Claris Works

Claris Works and other integrated packages continue to pose a threat to the core Office business. First, the threat of just having bundled software that appears to meet (or actually does meet) all of a users needs derails the purchase of Office. This is especially true in the Home and Small Office market. Second, these products, by virtue of doing far fewer things, take significantly less resources to run. This is especially true for multiple applications scenarios (i.e. a chart in a word processing document).

Lotus Notes

The Notes 4.0 document, which is a stream based form in combination with standard controls and a full-featured rich text control, can quite literally eliminate the need for word processing documents altogether, especially in a scenario where the document will never be printed. Over time, especially if Notes adds support for some extensible controls, the need for other document types could be drastically minimized.

FL AG 0043537
CONFIDENTIAL

MS6 5007155
CONFIDENTIAL

10. Quick Overview of the Office Product Team's Office 96 Focus

Assistance

- Social Assistant: Character delivers all assistance.
- Improved Answer Wizard: Uses contextual clues (e.g. type of the selection) to provide better guesses at the help topics a user is seeking.

New User-Interface

- Shared Toolbars and Menus: single code base; treats toolbars and menus as a single concept, making the code simpler and more easily leveraged (e.g. toolbar and menu customization solved together)
- New, more appealing visuals including document thumbnails in binders and on the desktop, and new toolbar look
- Office Binder Improvements

VBA Everywhere

- All products have an object model
- Single VBA development environment

Shared Drawing

- Escher component: Same built-in drawing tools for Word, Excel, and PowerPoint; joint effort with the Graphics Product Unit Escher team. GPU will also provide an Escher server aimed at making business graphics (e.g. Org Charts) easier.

Infrastructure

- Office memory manager, string manager, and resource manager aimed at sharing common routines and enhancing performance

Ren Integration

- Office Explorer
- Office events tracked by Ren journal (e.g. timeline view of document versions)

Software Administration

- Setup for the company
- Improved configuration management

Document Management

- Document Librarian including basic document management, improved content indexing, versioning, and check-in/check-out
- Improved File Open and Save As dialogs, including rich views, and easier to use interface for searching
- Infrastructure to allow rich views of FAT

FL AG 0043538
CONFIDENTIAL

MS6 5007156
CONFIDENTIAL

11. Office 96 Dependencies (andrewk)

The following details the current dependencies the Office 96 project has on other technologies and groups at Microsoft. It is a working list; please see \\office\plans96\misc\office\dep-revu.xls for an up to date list. This list is updated in each month's Office 96 PDL.

11.1 DAD

- Word, Excel, and PowerPoint code complete and ship at same time
- Ren code complete 10/2/95, ship same time. No Mac.
- Doc Mgmt team provides Doc Library, File Open/Save As, Infrastructure for rich views, and content indexing for Jet.
- Project Buttonface editor, probably just ship Office 95/Chicago version. May need new Mac version.
- Configuration Mgmt team provides Setup and Configuration Mgmt tools. Feature list and schedule not final, so not yet accurately reflected in product team schedules.
- Also bunch of cross-DAD product dependencies like Graph, converters, and proofing.

11.2 DD

(Lots of cross-component dependency management in DD within database and VB teams.)

- Sterling team provides Access 96, Data Manager, and IntelliBuilder on same schedule as Office. No Mac.
- VBA³ team provides Visual Basic for Office development environment and Forms³ based VBA forms, including Mac. Will be reviewing schedule in more detail this month--need to make sure have enough time for integration and testing, especially on the Mac.
- Forms³ team provides dialog forms, OCX 2.0, and Datadocs. Schedule tied to other pieces.
- VBA team provides VBA '95, OCX Marshaling, including Mac versions.
- Designers/DaVinci team provides Query Tool, Data Scope, Table/Schema Tool. No Mac.
- DART team provides DAO, OLE DB and providers, QJET, and Red. No Mac.
- VC team: got first drop of Mac tools; however, still waiting on schedule for updates. Committed to compiler support, assembler support, and link for all platforms, and improved profiler. Compiler support done except NT-PPC (7/1/95); Assembler support done--high priority bug fixing available for Alpha and MIPS, no testing and support plan in place yet for NT PPC; Most lLink done except no commitment yet for NT-PPC. Supposed to have gotten an improved profiler NT version on 2/5, but haven't seen yet. Would also like PCode on PowerMac and Lego support (source code hints), but no commitment yet. Working with them on WLM for PowerMac; hoping to get OCX extensions and true Win look and feel for controls.
- Hoping to get Lego for PowerMac, as well as Lego PCode interoperability. No commitment yet.

11.3 BSD

- OLE: OLE 3 interfaces, spec only. Mac in-process servers by 5/95. No bootspeed and RAM usage improvements, multithread work, docfile improvements, or IReconcilableObject work currently planned beyond Win95. Not committed to OCX marshaling (controls group in DD to handle?).
- Meeting with BobMu and team to discuss features like Sharable system objects (GDI and User objects), Virtual memory hooks for optimal memory use (discretionary memory; optimized discard algorithms), Compressed resource types, and Animation support. Unlikely this will make it into Office 96.
- Windows Networking team providing Doc Conferencing APT's
- NT Security Group providing RC4 Encryption; shipping code from RSA
- MAPI improvements, no set commitments yet.
- SYS (Haifa) working with Ren team on MAPI based address book provider

11.4 PSD

- TAZZ team providing Speech recognition APT's
- Investigating Explorer extensibility
- Working to complement O'Hare Internet work and Office on-line document work for 96 and 97
- Ongoing communication and design meetings for Windows 97

11.5 AT

- NLP team providing improved grammar checker (Noah), morphological language data, and more languages for grammar checker. Joint effort with Word team. Probably won't be able to get languages other than English in time for Office 96, but Word will support older grammar checkers for other countries.
- May get graphics compression from Marvel for Word, or may use JPEG. Not definite yet.
- Decision Theory group providing Answer Wizard consulting; no code dependency.

11.6 Consumer

- Utopia/Bob team working with Office on Utopia animation services, schedule 4/7/95 (Win) and 8/11/95 (Mac).
- Character team providing 10 character files, last to be done by 11/2/95.
- Quill provides text for Escher (shared drawing) OLE server. Recent new hires should make FE version possible. Working with Escher team on schedule and final commitments.
- Publisher provides Clip Art Gallery.
- Bookshelf and hooks to Office. Working with them on design and schedule.

11.7 CLWill's team

- Committed to making Espresso work for Office 96. Working with them on schedule and issues. Need SDM and Office resource support.
- Automatable performance measuring tools for Win 95 and NT
- Would like SLM improvements (SourceSafe is a possible alternative if feedback is addressed), but no commitments yet.

11.8 Hardware

- Will exploit 3D mouse when get APT's

11.9 PSS/Supportability

- MSInfo32 to be delivered for Office '95. No commitment or schedule yet for a new Office '96 version, or Mac version.

11.10 Who Depends on Office

- Providing character/social UI, improved Answer Wizard, unified menus and toolbars, drawing, memory manager, string manager, resource manager, binders, doc management, and other shared features to all Office (Word, Excel, PowerPoint, Access, and Ren). Some of these may also go to components (e.g. menus and toolbars to VBA²).
- SDM to Office, Word, Excel, Doc Mgmt, and Works.
- Drawing OLE Server (also replacing Word Art and Org Chart) to Office apps and Publisher
- Setup and configuration management for Office and most of the company

MS6 5007158
CONFIDENTIAL

12. Appendix: DAD Research (chrisgr)

Objective

To investigate user activities, user model and user interaction topics of long-term significance to Office. This will also tend to include topics of interest to the Systems shell group, as Office synergy with Windows will continue to be an important objective.

Time Horizon

Research will center on technologies, focus areas, or other topics that could yield a significant competitive advantage over the next 3 to 5 years. Of particular interest will be topics that represent significant "sea changes" that we could support sooner or better than our competitors by investing in preparatory work. Often we can foresee that these changes are coming but without more work we cannot tell how, when, or at what level we should include them in our key products.

Focus on User Interface

Work will primarily focus on the user interface. Research into the future implementation architecture will be the responsibility of Tony Williams' group and the Office development group.

I define the user interface as the abstractions, representations and activities in the conceptual model presented to the user, and thus those that are (hopefully) in the user's mental model of the system. There will naturally be a close relationship between objects in the conceptual model and in the implementation architecture, but there will not necessarily be a one-to-one correspondence. The user will make valid distinctions between objects that in part share the same implementation. Conversely it may be necessary for reasons of code efficiency or legacy code for the user model to represent objects as the same that are in fact implemented differently.

Focus on Core Areas

Work will focus on core areas for Office, which are defined as those that would impact all or most applications. These are also typically areas that might need to be supported by the System if we found this to be to our advantage. It is assumed that the category-specific product groups will be responsible for their own research unless there is an advantage in doing the research in the DAD research group. However, it probably would be useful for the DAD research group to maintain a list of future-oriented work going on in the product groups and around the company, looking for synergy, and acting as a clearing house for high-level information.

This style of identifying and working core areas is similar to what I did as Director of Applications Interoperability, in which I led the definition of the core feature set for Office, and coordinated the design and roll out of these features. For that work the time horizon was about 1 to 3 years, whereas for the proposed research the time horizon is longer, permitting some additional flexibility in identifying opportunities, understanding them, and rolling them out.

Style of Research

- Generalizing across user activities and applications that have previously been treated as separate.
- Looking into existing bodies of research that have not been fully tapped in our software design work. (For example, decision theory, cognitive psychology, etc.) The AT research group will be an important resource.
- Thinking about how technological change will impact the ways people can optimally perform activities. We will have to develop a deeper understanding of why things are the way they are now, and the underlying motivations being fulfilled.

MS6 5007159
CONFIDENTIAL

Deliverables

Prototypes will be an important way of exploring ideas and communicating results. Papers will also be produced, both while investigating topics, and to communicate the full detail about topics. We intend to deliver and present these at regular, reasonably closely spaced intervals (say 2 months) to keep others informed of topics being considered, informed of progress, and to generate discussion and feedback.

Resources and Process

The DAD Visual Interface Group will contribute design and prototyping expertise. The Usability group will assist with research into HCI literature, and usability testing. It is also hoped to hire an SDE level prototyper, and maybe one more person, their specialty TBD.

We also expect to draw on other work being done around the company, most importantly AT Research, but also Consumer, Systems and DDT.

In addition, I plan to involve members of other functional groups in DAD and Systems, as much as their time permits. I may explore ways of involving academic researchers outside Microsoft in some of this work.

We need to develop a workable way of involving people from around DAD. Part of the solution could be to use Exchange, which would be ideal for maintaining discussion bases on various topics. This would make it feasible for people to participate as a background task because they could pick up a discussion thread and participate in it as time permits. Also this would naturally lead to the use of Exchange for managing the actual design of the 97/98 time frame releases.

Rollout of Results

The work of the DAD research group will have an impact only through its ability to demonstrate ideas or designs that other groups choose to implement, and it's success and continued existence should be measured on its ability to do this. Therefore it is important to maintain lines of communication, keeping others aware of work being done in the research group, and providing plenty of opportunities for feedback of ideas and comments to the research group. This will be more-or-less modelled on the way we worked together on Office 4.x, with adaptations because of the longer time frame available and the lower time commitment others in DAD have to spend thinking about this far in the future.

First Steps

The first two to three months will be spent broadly thinking about user interface problems, issues, challenges and opportunities that we currently face. The goal will be to attempt to develop an integrated understanding of key directions that user interfaces need to evolve, and to identify leveraged areas on which to focus. This will involve alternately thinking broadly and then drilling down into specific areas where it is necessary to understand them better.

Thereafter, the plan is to choose a limited number of areas to investigate in detail, subject to availability of resources.

Research Topics

This paper is an exploration of topics in computing where advances in technology or broadening uses of computers offer opportunities for advances in human computer interaction. The goal is to identify "sea changes" which our applications and systems can leverage to gain market share, drive upgrades and to expand into new markets.

The following are broad areas that appear to offer opportunities or require solutions:

Interaction With the Rest of the World

Support for all types of interaction between the user and the rest of the world. Involves any information not both *created and consumed* by the user his current computing device. Information may be accessed through some form of connectivity or through manual connectivity such as inserting a CD ROM, or through "sneaker net".

Natural User Interface

Developing a better match between the user interface and human abilities: sensory, mental and motor skills. This could include changes to the underlying presentation techniques, user interface metaphor, and interaction techniques.

Scalability and Flexibility

Adaption of the user interface to changing conditions, while maintaining the user's mental model of the system, and as much as possible, providing the ability to perform the same tasks and access the same information

Interaction With the Rest of the World

Support for all types of interaction between the user and the rest of the world. Involves any information not both created and consumed by the user on the computing device he is currently using. Information may be accessed through some form of connectivity or through manual connectivity such as inserting a CD ROM, or through "sneaker net".

This topic includes

Messaging and Conferencing

- Real-time collaboration or conferencing.
- Non-real-time collaboration or conferencing
- Threaded communication vs non-threaded
- May be over an online service, LAN, wireless link, etc.
- e.g. Mail, bulletin board systems, real-time chat systems, MUDs

Reference

- Creation, distribution and storage of documents. (Issue: What is a document?)
- Content: Authoring, viewing (although not all content necessarily online)
- Information may be on an online service, LAN server, CD ROM, or user's own hard disk
- e.g. Document libraries, News publication, CD ROM-based information, etc.

Routing

- Distribution of structured documents along pre-defined paths
- May also include semi-structured task sequences
- e.g. Business processes such as expense reports, bug reports, purchase orders, etc..

Tracking

- Distribution and acceptance of activities
- e.g. Group scheduling, project management, status reporting, contact tracking, etc.

Key issues

What is a document?

- Documents are the reason for existence of most DAD applications
- Documents are the objects that people use to exchange information.

Office Product Unit 1995 Three Year Plan

- What should their structure, capabilities and properties be to support the above scenarios?
- Do documents continue to have the same importance when most information is created and viewed online? Maybe they are largely superseded by threaded knowledge bases, or viewers that show the results of queries or information-magnets.
- Note: There is a whole class of applications that are not primarily for working with documents: Business transaction processing, etc. How do these fit in?

Finding and navigating information

Need to develop a better understanding of collaboration

- What are "projects"?
- How to support real-time collaboration
- What is the magic of Lotus Notes? What's the best way for us to avoid losing applications market share to Note's enabled applications? What's the underlying user conceptual model of Notes, and can it be generalized in ways that we can better support with our more componentized architecture?

Scalability with speed or absence of connection (see below, under Scalability)

Natural User Interface

Developing a better match between the user interface and human abilities: sensory, mental and motor skills. This could include changes to the underlying presentation techniques, user interface metaphor, and interaction techniques.

Technologies available

Social interface

Decision theory

Natural language input (with keyboard)

Natural language interpretation (of existing text)

Speech commands

- Speech as means of access to invisible capabilities
- Speech as a skill-based way of accessing capabilities
- Can provide "impedance matching" between the user's mental model and available capabilities. But, to do this we would need to be very flexible in understanding speech.

Speech dication

Speech output

Gesture

Pen

Issues and thoughts

Expanding the uses of decision theory and artificial intelligence.

- Decision theory could be brought to bear on many of the topics in this paper. What is the optimal mix between direct tool based interaction delegated interaction? To test our theories and calibrate our models, we need more information on user behavior.

Parallel between real-time conferencing and natural user interaction with an intelligent system.

- During a real-time computer conference, one's computer "contains" one or more "very intelligent" agents. I believe that many of the same user interface issues will arise as computers become increasingly intelligent and support more natural interaction techniques.

Identifying user requirements:

- Can we track user actions and identify higher level behavior patterns that would suggest additional user requirements? Could we track this over MSN and offer additional capabilities to the user on the fly? This is also an issue of scalability. (See below.)

Multi modal interaction:

- Various channels of interaction have differing strengths. We need to find ways to permit users to shift between and blend interaction channels such as keyboard, pointing device, speech, gesture, etc. Strengths of one channel can compensate for weaknesses in another.

Navigation:

- We support a wide range of types of navigation in Windows and applications today. A move to 3-D and possibly other interaction devices could provide an opportunity to rethink and unify at least some of these.

Skill based, rule based and knowledge based interaction:

- Push as much interaction as possible to down to styles interaction requiring lower levels of cognitive load, and minimize broad disturbance of interaction as configuration changes.

Scalability and Flexibility

Adaption of the user interface to changing conditions, while maintaining, as much as possible, the user's ability to perform the same tasks and access the same information.

Types of scalability

Customization and componentization

- Dealing with changing capabilities
- Non homogeneous user configurations cause problems with documentation and training

Adding capabilities through upgrading

- This is a key issue because the importance of annuity revenue in the future.
- Incremental addition of capabilities (maybe via subscription via online service)
- Enhances the feasibility of a 12/24 month release cycle

Screen size scalability

- Current approach of making many controls visible does not work on small screens

Connectivity scalability

- Maintain ability to work although connectivity of currently used computing device to the rest of the world varies.
- A key requirement seems to be the ability to cache information as locally as possible (i.e. replication), and the ability to maximize synchronization given the availability/speed of the link. To
- How to minimize user surprise at what is available/not available at a given time

Interaction technique scalability

- Provide a consistent user model across desktop machines, laptops, and palmtops.
- Explore interaction techniques that work on all platforms

Office Product Unit 1995 Three Year Plan

Hardware capability scalability

- Not all users will have the latest hardware
- At a given time, more portable machines will trail in power vs less portable machines.

Issues

Discoverability and recallability of capabilities

Transferability of skills across different user interface configurations

User control over scaling of capabilities

Skill based, rule based and knowledge based interaction. (See above, under Natural User Interface)

Assisting the user in managing scalability:

Analyze user actions and provide diagnostics about improving scaling settings, or about new capabilities to add. Maybe even change settings automatically. Over online service, could even automatically provide suggestions to us about new capabilities to make available, which these techniques would eventually bring to the attention of the user.