From: Malcolm Pearson
Sent: Wednesday, January 10, 2001 12:21 AM
To: Steven Sinofsky
Subject: FW: Driving end-user scenarios for Exchange in the .Net time-frame.

Message Flag: Follow up
Flag Status: Flagged

Organizing for Office.NET

GMail.doc

Can you suggest with whom I should best get engaged with on this? Outlook folks seem like an obvious choice. I also think it would be useful to get in touch with collab folks from a few perspectives – I think that the base messaging scenarios should scale 'easily' into the richer collab scenarios (even if e.g.: Exchange doesn't do too much to implement the actual collab scenario) – further, there needs to be thought about migration of some existing collab scenarios like PublicFolders.

Thanks,

Malcolm Pearson

-----Original Message-----
From: Eric Rudder
Sent: Tuesday, January 09, 2001 10:27 AM
To: Malcolm Pearson
Cc: Bill Gates; Paul Flessner
Subject: RE: Driving end-user scenarios for Exchange in the .Net time-frame.

i think stevesi's group is critical to work with. i enclose some mail below which outlines the teams for the next generation of office.

there are some folks in research worth brainstorming with as well -- anoop's team in particular.

finally, bobmu's team has also done some good thinking on user scenarios, and they are worth meeting with as well.

i realize that this may not be the "singular" response that you were looking for. if you twisted my arm, i would tell you to spend most of your time w. the office team.

bill may have other input in terms of groups or priorities.

we do need to work to make this more clear, and will try and do so in the coming weeks, but in terms of the planning process calendar that paul outlined, i hope this can get you started. i am also happy to spend some time w. you as an individual contributor if you are up for that as well.

-eric

-----Original Message-----
From: Malcolm Pearson
Sent: Tuesday, January 09, 2001 9:25 AM
To: Eric Rudder
Cc: Bill Gates; Paul Flessner
Subject: Driving end-user scenarios for Exchange in the .Net time-frame.
I need to know who in the company you believe owns driving end-user messaging scenarios, especially wrt clients of Exchange Server. I know that there is effort underway to determine who owns the implementation of the client that will support messaging, and perhaps you'll tell me that the answer to these questions is the same.

Plaintiff's Exhibit
6840_G
Comes v. Microsoft

I believe that it is important that we as a company come up with some advancements here to help drive the adoption of Exchange Server both from the point of view Exchange Server as a business and also from the point of view of driving adoption of .Net and Yukon in general.

We have generated some very early ideas within Exchange Server. That said, they won't mean anything if we cannot get this aligned with general company strategy. Further, the things that we're thinking about would require client-side support and so I believe it is important to engage early with other teams involved in this area to make sure the value is going to pan out.

Thanks,

Malcolm Pearson

From: Steven Sinofsky
Sent: Tuesday, December 19, 2000 12:24 AM
To: Bill Gates; Steve Ballmer
Cc: Jeff Raikes; Eric Rudder
Subject: Organizing for Office.NET

I wanted to provide you with this organization announcement before it goes out on wednesday afternoon. This is the transition to our new services organization. This is really a *huge* change for our team and the plans outlined below are *very very* aggressive. I really don't know how much we can get done of this, but we are signing up for a lot right now.
Jeff and I have talked through the personnel details of this.

The main elements of this organization are an alignment around core service scenarios and playing down the "app-centric" organization of the past. The cheat sheet is basically as follows:

- Office Shared Services -- this is where all the shared code and service integration is done. HeikkiK, GrantG, and Antoine continue to lead this. This also includes out "operations". The details of the shared service teams (as much as we know) are below.
- Data and Developer Services -- this is our "structured" data group. This is largely unchanged in composition but the goal is to get much more done in the way of services for developers and end-user functionality. RichardM owns this. You can think of this as the "excel and access" team of Office10.
- Communication and Time Management Services -- You can think of this as the "outlook and powerpoint" teams, but it is now led by RalfHa (Ralf moves to Outlook). The core scenarios are around meetings and communication.
- Authoring and Community Services -- You can think of this as the "word, FrontPage, and SharePoint" teams, but it is now led by KurtD (Kurt moves to Word). Reed will own FrontPage and SharePoint working for Kurt. The goal here is to do a great job for XML authoring, communities of shared documents, etc.
- Assistance and Worldwide Services -- This is a new team focused on using the web to help people learn about, how to use, and to personalize the entire "training" experience. Jeff Olund will lead this team, which is a new role for him (owning product development beyond UA and localization).
- Sustaining Engineering, Marketing, and East Asia are largely unchanged (though we are working through some challenges with our Asian development responsibilities).

In this you will not find a lot of specifics about which .NET services we are using and how, but this is clearly top of mind. I am just being sensitive to the feedback about "too many dependencies" and leading with that.
I'm excited about this change. It is a huge challenge to pull this off in the midst of trying to ship, but I think we can do it and not risk 3/2/01.

To: Office Product Hallway Worldwide
Fr: Steven Sinofsky
Re: Organizing for Office.NET
Dt: 12/20/00

As we enter the final phase of Office10 and get to 3/2/01, I would like to announce some significant changes to the Office organization, but more importantly to our core focus and strategy, that will begin the road to building Office.NET. These changes are outlined below, though I do wish to preface these changes with an introduction so please read on.

It goes without saying that delivering Office10 on 3/2/01 is our primary goal. The efforts on Office10 have been extraordinary and the results show. There is a great deal of excitement building around Office10 as we move towards a launch and wide availability in May. Office10 is an incredibly important product for Microsoft and our focus on shipping a high quality Office10 will insure Microsoft's continued success. As we have talked about at our team meetings and in the previous Office.NET memo <http://ngo/Shared%20Documents/Creating%20Office.NET.doc>, it is important for us to begin the transition to Office.NET while we finish Office10 in order to make for a smooth and efficient transition. The changes announced today are effective immediately for the purposes of planning and staffing Office.NET, with the final transition taking place after we RTM the US product on 3/2/01. Everyone will continue to own shipping their current responsibilities without any changes.

Program management and product planning, and many others, have been working hard on developing a vision document for Office.NET. It is always the case that organization changes need to follow the goals of the product and at this time we have a solid view of the priorities of Ofifce.NET so we can make the first important changes that will help us to better execute on these priorities. Our focus will be on *creating a release of Office that combines the power, flexibility, and usability of our core applications with the benefit of newly developed hosted services and the service business model.* This is a major transition for our team. We will significantly reduce the amount of code we write in our existing code bases while we substantially increase the amount of new code (both server-based services and client code) that we write. We will significantly broaden the types of experiences we include in Office and integrate those into the existing Office experience, increasing the utility of Office for customers. Where we previously focused on optimizing the tasks of editing, formatting, collaborating, and analysis which customers have sometimes said "just" save them minutes or seconds during the day, we will now focus on solving a new set of customer problems and focus on saving hours of work for our customers. This is incredibly exciting and also a tremendous new challenge. We will all be stretched to learn new techniques, new tools, and new technologies.

Our goal for Office.NET will be to develop a release that brings us into the services model where we can begin to offer customers substantial new features and begin the *feedback loop* that is so important in the services world. We want to have one release where we are 100% focused as a team on the same delivery date. It is incredibly important that we transition to this model and begin the process of working with real customers who will begin to use Office.NET everyday to get their work done. For that reason, we will work on a short and incredibly focused schedule. After we deliver the first release of Office.NET we will be in a position to better understand the idea of new types of schedules such as ongoing releases, rolling releases, or having staggered releases. As we finalize the vision statement in January we will have a firm grasp of the feature set that defines success for Office.NET and we will then determine the final schedule. It is likely that we will continue to aim for a schedule that has us releasing Office.NET in the summer of 2002 since that strikes the right balance between an opportunity for depth of work and the need to be in the marketplace. We will be very deliberate in the features and implementation we choose to enable us to get to the market in this timeframe for our leading edge customers. The competitive and business situation demands that we be fully service-oriented as soon as practical.

The principles that guide the new Office.NET organization are as follows:

- Create an environment where the organization reflects our priorities for building the product and service experience, while at the same time acknowledges that customers experience a single integrated experience.
- Change our focus from building *applications and MSO* to building services that are based on core scenarios and a shared service experience.
- Ensure that we have the right structure to collaborate effectively with the small set of critical .NET services being built by other teams. This minimizes the number of dependencies any single group will have in order for each team to have maximum control over their charter. Success for Office.NET includes building on these key .NET platform services.
- Provide new opportunities for everyone in the organization—opportunities to learn new skills, address new challenges, and work with new people on a day-to-day basis. To be concrete, many people will move to new teams on Office.NET. Our employee survey highlights the desire for people to do new things and have new opportunities for growth, and this is a very clear time and opportunity for change. If you have been on your current team or worked with the same set of folks for two or more releases of Office, you might consider this a good opportunity to seek out a new challenge within Office.

The Office team is blessed with superb leadership throughout the team. This announcement is about the Directors and General Managers/VPs of the Office team. Each of them deserves a great deal of credit for leading Office10 and each will be a great leader for Office.NET. Clearly there are a number of details around these changes that are not described in this mail. It might even be the case that you are not certain exactly where your current organization resides in the new organization. Please trust me that whenever one makes changes this sweeping for a team of over 1000 people, it is not possible to incorporate everything in one message. It is absolutely the case that there is a well-defined and critically important place for everyone in this new organization (in fact, if you have any friends who might want to join Office, now is an excellent time to do so, *seriously*). Each of the following managers will be sending out follow up mail in the coming weeks that outlines the next steps for their new organizations.

To be very clear on the process from here on out, please keep in mind the following milestones:

[redacted]

January 2 to mid January

Follow-up details for all of the new service teams including the development, test, group program managers for each of these new teams.

Mid January through February

By *default*, for Office.NET you will work within your current team. But you might also consider at change, and if you decide to move to a new team then you are invited to interview and determine if there is a mutual fit.
Mid February
Program management completes the vision for Office.NET and then starts on feature prioritization, prototyping, and other planning activities. Ownership of Office10 features is cataloged and distributed appropriately through the new organization.
January and February
In response to requests for more authoritative training, technical training for all team members will be offered during this time. Topics will include SQL, Windows 2000, XML and others. All technical team members are strongly encouraged to attend.
March 2, 2001
Ship Office10
Mid March
Schedule announced for Office.NET along with specific planning review milestones.

The new teams we are creating will each be responsible for existing Office10 features and code. This organization is making an explicit commitment to sharing the responsibility of existing core code and new services within teams. Office's core code is not a *legacy* but an asset worth hundreds of billions of dollars. We are taking to heart the lessons learned by other parts of Microsoft that have attempted to transition to next generation products by farming out or isolating the *old* as the bulk of people went off to be 100% focused on the *new*. We have consistently learned that this type of transition underestimates the energy required to keep the existing code healthy and fails to deliver on the right amount of integration and a clear migration path for customers. The great value of Office.NET will come from great services, but it will also come from strategically and carefully updating our core products and also precisely choosing key integration points between our core products and new services and new client code. This requires a high level of sophistication and strong leadership—both of which I am confident the Office team possesses.
The following descriptions will highlight the existing code that each of these new teams owns. The new teams, missions, and leaders for Office.NET are as follows:

Office Shared Services

Office.NET, as with any release of Office, counts on a consistent and shared experience across the whole product. Office Shared Services provides the *Office experience* as a service, which includes (but is not limited to) the subscription experience, management of state/identity/profiles, internet deployment, shared user-interface (web and client), and the operational elements of the service infrastructure. As we have had shared feature teams in the past, we will have shared services teams each with a development manager, test manager, and group program manager who are responsible for developing, deploying, and integrating shared services. The Office Shared Service teams will own the current MSO code. One new aspect to the Office Shared Services is service operations. Office.NET will be incredibly focused on building services that are scalable, secure, robust and also dollar-cost efficient. We will be tracking many important metrics around the services we create (number of servers, users-per server, disk space required, and CPU requirements). Through the Shared Services Operations we will manage our overall efficiency, but all of the Office.NET team will be responsible for building efficient services that allow us to be a profitable business. We are fortunate to have the experience of the OfficeUpdate team to lead the way in our operational efforts for Office.NET.
I am pleased to announce the following leaders for Office Shared Services:

- Antoine Leblond will lead the Office Shared Services Development team.
- Grant George will lead the Office Shared Services Test and Operations team. John Zanni and the OfficeUpdate team will report to Grant and assume the critical responsibility of Office.NET operations.
- Heikki Kanerva will lead the Office Shared Services Program Management, Product Planning, Design, and Usability teams.

The following are the Office Shared Service teams for Office.NET:
Office Foundation Services. The goal for this group is to deliver the set of services that are *musts* for Office.NET to be in the game. These services include ones that are needed by Office.NET as a whole or by a certain set (more than one) of Office.NET components. Examples of these are logon, billing, directory and other shared foundation services needed by the different Office.NET components. Because there will be a lot of interaction between this team and the.NET infrastructure teams within other parts of Microsoft, it is crucial that we have a dedicated team focused on these services as these form the core of a consistent Office service experience.

Office Deployment and Roaming Services. The focus of this group is on three areas: installation and maintenance over web, secure and reliable licensing model, and technology for roaming the Office.NET user experience across the internet. In order to be a successful service the critical challenges such as installation, size and granularity of installation, use of web as the resiliency source, authentication of the user, roaming user settings over net, remote app installation and others will be tackled by this team.

Office Development and Release Services. This team will continue the work of making sure we have reliable and timely releases of Office.NET client available all the time. This team will own establishing the new practices of releasing services and working with the operations team on how we will have Office.NET *builds* during the development of Office.NET. Release Services will work very closely with the Sustaining Engineering Services as we define the model for keeping our customers current with Office.NET. In addition, the Office Development and Release Services Team will be responsible for most of the tools that run our development process - the build pipe, test automation framework and related tools and release processes. They will also be responsible for defining and developing (in conjunction with the Operations and Infrastructure team) those tools that drive our release-to-the-web process, monitor usage, tools that help manage scalability issues, etc.

Office Document Storage Services. This team will focus on the scenarios enabled by storing Office documents in the cloud, and on making sure the experience of saving, retrieving and searching documents is a great one. Examples of services owned by this team are document conversion, virus checking, searching, and down-level platform rendering. A key scenario for this team will be to enable the collaboration on documents that are stored in the cloud, building on our excellent *Send for Review* work in Office10. This team will work closely with the both the Knowledge Management Product Unit and the SharePoint services team to make sure we have a great document storage strategy with a great user experience.

Office Instant Messaging and Notification Services. The focus of this group is to integrate the instant messaging experience and make it a seamless part of Office.NET and provide a central service for all different Office.NET parts that need to use notification. The first part of the work is to take MSN Messenger make it part of the Office.NET user experience and availability everywhere across the product. The second part is to provide a framework for the user notification service. Any Office.NET event (from file updates, email, meetings, etc.) will use this service. Because of the breadth of integration across all of the Office user experience we wish to accomplish with IM and the accompanying notifications, we are creating a shared service team to take on this effort.

Office Satisfaction and Quality Services. The goal of this group is to continue the charter we have started with Office Watson and handle the detection of any kind of errors, reporting them and making it easier for the user to connect with Microsoft and to get a personalized experience on *closing the loop with customer.* This team will own the integration with the professional support on Office.NET. This team will also own keeping the Office customer's installation healthy including system stability, registry repair, as well as the incredibly important efforts around security. As we have learned with Watson, there is a wealth of information to be gathered around collecting and analyzing customer feedback and this team will own the infrastructure and mechanism for making this a seamless part of the Office.NET customer experience. This team will work closely with Sustaining Engineering Services.

Office User-Interaction Services. This team will own the face of Office.NET. Whether the user connects from their main machine or an airport kiosk, they will experience a consistent UI that integrates all the new Office.NET features, specifically customized for that user. The team will own the service integration model into the applications, the overall look and feel of the services, how they interact with each other, how we customize the UI on the fly to adapt to each user and advertise new and upcoming services.

Data and Developer Services

The goal for the Data and Developer Services team is to help developers and end-users easily create, manage, organize and share data on Office.NET. This team will develop a collaborative data-centric web application that is targeted at end users and will provide an *out of the box* experience that will enable customers to create, view, and share their data with others over the web. It will be tightly integrated into the Office.NET service experience and build upon (and into) some of the essential hosted services provided by the SharePoint Services team. One of the most underused aspects of the web is the ability for end-users to tap the power of vast amount of web-available data and integrate that into their own web-based view of that information, collect and further analyze information, or combine multiple sources of information—Office.NET will greatly enhance this experience. Building lists, sharing spreadsheets, viewing reports, and working with external data sources and schema are all part of the vision of this application. This application will have broad appeal, and we will measure our success by the number of people that use this application.

The service-based world offers both opportunities and challenges as we strive to meet the needs of the developer community. The details of our approach to developers are still being worked through, but our commitment to this customer is clear. Office.NET's efforts will be very approachable from a developer perspective, but will strive to be upwards compatible with the Visual Studio.NET framework. We have a tremendous opportunity to re-think our approach to building a tool that is both powerful in terms of the types of collaborative data-centric applications it enables, while still being extremely approachable. We will measure our success by assessing how easily the tool allows us to consume and integrate services in building our application.

Data and Developer Services will focus on Office's core competencies, but we will not limit ourselves to traditional database or spreadsheet design. This team will combine the best of Access and Excel into a single, unified experience that scales from a simple list (Excel-like) to a structured set of lists with relationships (Access-like). We will use the Office Web component and Designer technologies and learnings to offer new UIs on this data and dynamically build forms and views that render on multiple devices and form factors.

I am pleased to announce that Richard McAniff will lead the Data and Developer Services team. This team will have responsibility for Excel, Access, Web Components, and Programmability. The Office Designer people will also continue to report to Richard.

Communication and Time Management Services

It goes without saying that communication is an increasingly important part of the work day—email, telephone, faxes, schedules, and meetings are all the communication tools that individuals spend time coping with on a daily basis. The Communication and Time Management Services group will enhance Office's role as the essential tool that handles all of a person's daily communication and time management needs in an integrated, one-stop-shopping, out-of-the-box fashion. Communication encompasses traditional and emerging modes of exchanging information, both in real time (e.g. meetings, telephone, and voice over IP) as well as asynchronously (e.g. email, faxes, recorded presentations). Time Management encompasses all home and work calendaring needs of individuals and groups.

Today much of this information is locked away in an Outlook PST or on an Exchange server. Our service work in this area will free this information to roam with customers no matter where they go (airport kiosks to home PCs) and enable seamless access to the full fidelity of mail, schedule, and PIM information from any browser-based PC around the world. At the same time we recognize that our customers have a work life and a home life but the integration of this valuable information across these two is very poor, for example trying to see a schedule that includes family commitments outside of work in addition to your work schedule.

One of the most important new scenarios we will enable in Office.NET is the *more effective meeting,* which will include the entire meeting process as well as incorporate a new hosted meeting service. The power of combining our time management and communication work into a single team allows us to focus on this critical *knowledge worker* scenario and provide leadership in an area where customers are clamoring for solutions to these everyday problems.

I am pleased to announce that the Communication and Time Management Services team will be led by Ralf Harteneck. This team will have responsibility for Outlook and PowerPoint.

Authoring and Community Services

The *document* has been the heart and soul of communication for our customers for over 15 years and the role of Authoring and Community Services is to lead the way in innovating this core functionality into new types of documents and integrate it with our advances in web authoring and team collaboration. The Authoring and Community Services team will build on Office's strengths in document authoring, web authoring, and team collaboration.

This team will focus on delivering services that make it dramatically easier for people to create, organize, and use individual and team web sites that include documents from any source, but particularly Office documents. By building powerful, collaborative communities for individuals and teams, where they can store and share information, collaborate together on projects, and share expertise and services with other users, we will radically improve the process of document creation well beyond that of emailing an attachment (or even Sending for Review). Authoring and Community Services will be making two significant technology investments. First, the phenomenal excitement around SharePoint has shown the power of an end-user web site. This team will deliver SharePoint to customers as a server product but will also lead the way by providing the infrastructure necessary to deliver Office SharePoint Services as a hosted service. Several teams on Office.NET will be contributing to and building richness on top of SharePoint Services. We can think of SharePoint Services as providing the team infrastructure necessary for our customers. Second, this team will provide a role for XML for end-users creating documents and dynamic web sites. There are significant opportunities for Office to bring XML to the *masses* in a friendly and incredibly useful way.

A new area for Office that will be owned by the Authoring and Community Services team will be building an entirely new application and code base to provide the experience for creating *lightweight* or *ephemeral* business documents, brainstorming at an offsite, or serving as a core adjunct to Office.NET's online meetings. As we have seen an increase in the role of email (short documents) and a lessening in the lengthy business memo, we have not done as much as we could to really improve the core writing experience for customers. Documents now are shorter, have more structure (think of all the bullets people use now), are often the spring board for the use in a variety of applications (when a customer jots down some ideas for a slide in Word or receive an email message, but then have to re-type them for PowerPoint). Documents serve as brainstorming lists but we have no tools for enabling multiple authors on a document in real-time. The possibility that many of our customers will have a TabletPC also offers us the chance to think through the role of ink and tablets for the common case. The team will work closely with the TabletPC team, using the new Tablet APIs for ink, and allowing for a smooth integration with both the tablet platform and Word. This is an exciting opportunity and one that can really show the role of innovative client applications.

I am pleased to announce that the Authoring and Community Services team will be led by Kurt DelBene. Reed Koch will continue as General Manager of FrontPage and the Office SharePoint Services team and report to Kurt. Both FrontPage and SharePoint are core elements of our Office.NET hosted web site and customization strategy, and with Reed's capable leadership we will also continue to focus on the unique business opportunities we have in these still young markets. Overall, the Authoring and Community Services team will have responsibility for Word, FrontPage, SharePoint, and the Text Services Group.

Assistance and Worldwide Services

Our ability to effectively assist customers in their work, whether it is getting started on a task, learning new ways of doing things, or finding support when things go awry, has been limited by the constraints of disk space, richness of the computing environment (audio, video, etc.), lack of personalization and a real-time feedback loop, and even the artificial divisions of content created by our own internal organizations (UA, Press, PSS, etc.). The connected state of the Office.NET customer will enable us to overcome these constraints and deliver a set of assistance services that will, for the first time, displace the person down the hall as the first and best resource for help getting your job done with Office.NET. This team will own the task of creating as well as collecting, organizing, and improving the various sources of assistance information so it can be indexed and searched from a single point-of-entry, whether that's the Assistant in the clients or the Assistance hosted service. One of the most innovative sites developed in the past few years is the Template Gallery. We will combine this effort with Office's most-heavily trafficked site, Media Gallery, and find new ways to use these existing assets to help customers save hours getting their work started.

This team also has responsibility for delivering Office.NET around the world. In this regard we will continue to localize the client applications and .NET services that are applicable outside the US (as we expect many of them to be), but will also tailor our offerings, when appropriate for our business goals, to include market-specific services in those markets.

I am pleased to announce that Jeff Olund will take on this expanded responsibility as General Manager of the Assistance and Worldwide Services team. Joining this team will be the Media Gallery and Template Gallery content and business development teams that have been part of OfficeUpdate. The Localization and Complex Scripts teams will continue to be part of the Assistance and Worldwide Services team.

Sustaining Engineering Services

Office.NET is about having an ongoing relationship with our customers where as part of the subscription we service the customer's Office installation. The MOSE team has been hard at work planning and building the infrastructure and technique for delivering updates to customers in a smooth and seamless manner and is already on the cutting edge of the service relationship with our customers. With Office.NET, Sustaining Engineering Services will be much more integrated into the development of the next product as we build in features and infrastructure to support an *always up to date* desktop. With our current Office (9 and 10) components as a starting point, they will continue to lead the evolution of web-based detection and repair, helping to define and fully leverage the new diagnostic and deployment capabilities in the Office.NET world.

Sustaining Engineering development, testing, and program management will continue to delve into the full range of our technologies, products and services as they plan and deliver our ongoing service offerings

Jeanne Sheldon will continue to lead the Sustaining Engineering Services team.

Microsoft Product Development, East Asia (MPD)

The East Asian markets represent the fastest growing geographies of the world for Microsoft and Office has led the way in developing products that are focused on the specific needs of these markets. We are fortunate to have a talented development team that dives in and provides valuable customer research specific to the Asian markets and can turn that research into valuable contributions to the Office product. MPD will continue to own and drive key elements of Office.NET such as the IME and market specific products, features, and services.

Akio Fujii will continue his successful leadership of the MPD organization including the R&D offices in Japan, China, Taiwan, and Korea.

Office Product Management

Office.NET is a new business for Office. As we move from a purchase of a license to a subscription relationship with customers we have significant business issues to address. The Product Management organization is gearing up for the launch of Office10, but at the same time they have been working hard to identify new business opportunities and to build the appropriate revenue models that will enable us to succeed with Office.NET.

Joseph Krawczak will continue to manage the Product Management team.

Next Steps

At this point I am sure there are a lot of questions. First and foremost, please keep in mind that there are many new opportunities as well as opportunities to continue working on the types of customer problems you have been working on—both options are open to everyone on the team provided there is an agreeable fit and understanding between managers. Just as a handy reference, the following lists the new Office10 teams and the existing Office10 responsibilities:

[redacted]

Office Shared Services

Office Shared Feature Teams, OfficeUpdate operations

Antoine, HeikkiK, GrantG

Data and Developer Services

Excel, Access, Designer, Web Components

RichardM

Communication and Time Management Services

Outlook, PowerPoint, OfficeArt and Office Activation

RalfHa

Authoring and Community Services

Word, FrontPage, SharePoint, Text Services
KurtD with ReedK leading FrontPage and Office SharePoint Services
Assistance and Worldwide Services
User Assistance, Localization, OfficeUpdate content teams
JeffO
Sustaining Engineering Services
MOSE
JeanneS
Microsoft Product Development, East Asia
Microsoft Product Development, East Asia
AkioF
Office Product Management
Office Product Management
JosephK

Over the next weeks, for everyone the most important things to do and think about are:

- Continue to do everything necessary to ship Office10 on 3/2/01!
- Read the follow up communications on the new teams for Office.NET.
- Discuss your interests and career goals with your manager.
- Consider what team and what types of challenges you might want to take on for Office.NET—you might choose to stay with your existing team and attack these same problems with a new focus on services, or you might aim for a different challenge on one of the new teams. The choice is yours to make, but this is an excellent opportunity to try something new, especially if you have worked on two or more complete (major) releases on your current team.
- Feel free to ask questions in email or in person of any of the Directors/GMs/VPs—all of us are dedicated to building an amazing Office.NET and we want to make sure that the transition from Office10 to Office.NET goes as smoothly as possible.

The transition from one release of Office to the next is never easy. Those of you that were involved in the transition from Office 4.x to Office 95/97, or from Office 97 to Office 2000 know how much changed between releases. Office.NET is an even bigger change and we are radically rethinking our business and product strategy. So the next month or so might be a little stressful for some of you as the details continue to unfold. I just ask that you show your patient side and most of all, keep asking questions—things are going to be incredibly fun and exciting as spring approaches.
Please join me in congratulating Antoine, Grant, Heikki, Richard, Ralf, Kurt, Reed, Jeff, Jeanne, Akio, and Joe on their roles in our new organization. We are incredibly fortunate to have such a strong team that is ready, willing, and able to take on the enormous challenge of transitioning Office to the next generation of Office.NET.
On to 3/2/01 and Office.NET!

GMAIL: Collaborative Group Email
New Ideas Beyond Zaplets

Anoop Gupta and Mary Czerwinski
Draft: 8/8/00

Zaplets from Firedrop.com provide a valuable means for collaboration among people. They combine some of the powerful features of web (e.g., shared access to the latest copy of data/content) along with the collaborative features of email (e.g., discussion threads, notifications), while exploiting peoples' familiarity with the email UI (e.g., specifying participants in a discussion using the "To:" field). Other compelling features include template-based authoring for common scenarios (e.g., polls, invitations), rich visualizations (e.g., bar charts, pie charts, pivot tables), a fun consumer look and feel and interactivity (e.g., buttons to vote). There is a service model associated with zaplets: new templates for zaplets are continually being added, the web-pages that form the basis of the zaplets model are hosted by firedrop.com, and the service is used to manage your list of contacts, distribution lists, and active zaplets. Zaplets make it easy for Outlook users to import their contacts list. Zaplet-like technology has the potential of getting rid of the clutter of stretched out threads of email on a discussion topic, where the content is thinly spread out across tens of email messages. With branching threads, where everybody is not replying to the latest email, extracting/following the content is even more difficult as is the management/filing of the threads for later access.

However, as they stand, zaplets have many weaknesses. Here are some of our quick thoughts about how Microsoft has an opportunity to extend and enhance some of the positive features of zaplets into a truly rich, fully functional group email application. Some of the zaplets weaknesses likely arise because they do not (can not) assume much about email clients and servers used to access/serve zaplets and many are issues they simply haven't thought about yet. Microsoft has an opportunity to leapfrog Firedrop and provide a much richer, integrated, and compelling solution for collaborative/group email (*gmail*). Roughly there are four categories in which improvements can be made. These are on: 1) creation of gmail; 2) client-side management of gmail; 3) notification model and interface on updates to gmail; 4) server-side management of gmail.

- **Creation**: Creating and sending a gmail should be essentially the same as creating and sending an email. This is not true for zaplets.
  - To create a zaplet you are essentially thrown into filling out a web-form. You are led to the form either from the zaplet.com web site and/or hitting the "new-zaplet" button on the web page displayed inside your email in an existing zaplet. You should be able to create using ^n or from standard menu pull downs. Selecting a template and previewing will need additional UI, but this too can be more tightly integrated.
  - Zaplets contact lists and distribution lists are not integrated with email. I cannot expand aliases, … I should be able to use standard Outlook features to fill out the To, CC, list.
  - Zaplets "to-do" lists and scheduling features are not integrated with the Outlook calendar. I should be able to see if there are conflicts in my schedule or deadlines for action items that I might receive via a Zaplet.
  - Many features in Outlook, such as making an email high-priority, flagging it to request a response, type-of-message indicators, … should also be available to gmail.
  - Ability to use an existing gmail as a template for new gmail to be sent out. This is currently not allowed within zaplets.
  - Ability to "reset" a gmail … so another round of voting can be done … e.g., for our weekly group lunch, whether to order Chinese/Thai/…
  - Ability to create new templates directly from the new email interface … as we allow for Forms today … though the UI is still not very nice today.
  - The Creator/Sender of a Zaplet is not given special status as is currently provided in Outlook (e.g., New Meeting Request). This makes the UI somewhat cumbersome for the originator.
  - Related to this, I can't send out a reminder prompt or summary of a zaplet to the whole "to" list, or even a subset of the members, without creating a new zaplet.

    - Summary zaplets, telling all that the issue is closed and summarizing the results, would be a nice addition.
    - A "what's new since the last time I was in here" feature would be beneficial.

- **Client-Side Management**: Once a user receives a gmail, how they respond to it and manage it in their inbox should be similar to email – this is not true for zaplets.
    - Hitting standard email "reply" button should allow you to respond to gmail, rather than as in zaplets where there are multiple separate HTML "reply" buttons within the zaplet … and hitting the standard email "reply" button does the wrong thing. The same is true for "forward", and other standard email buttons.
    - There is no way to sort on zaplets … just like there is on types of messages with attachments etc (one has to use search with keyword "zaplet"). The fact that "zaplet" is there in all headers is distracting.
    - The "size" field when displayed should reflect the actual amount of data that will be retrieved when gmail is opened. Currently, this is implemented poorly in zaplets … size is shown to be the size of email with just a URL embedded in it. However, since page behind URL is auto-loaded when email is opened, the size is wrong. This can be very bad when accessing on low bandwidth devices – I may make wrong prioritization decisions. The size also changes as new people add comments. This should be correctly reflected with a somewhat loose consistency model.
    - The semantics of deleting a gmail should be same as that of email. Not true for zaplets. E.g., Copies are maintained at the server even though both parties who received the zaplet have individually deleted it from their inbox. A zaplet owner can independently go to the zaplet.com web site and delete his/her zaplets from the web site portal. This can potentially "orphan" the emails that users have in their inbox that point to that zaplet. We need to better understand exactly what they do, but it does not match the email semantics -- I have my personal copy of what got sent out and others delete actions do not affect my copy – which is what we would like to provide.
    - There is no easy way to "branch off" of a zaplet and have a private conversation with others about the zaplet content. This could be functionality similar to "whisper" in chat rooms and is a feature that is often requested in collaborative software, like NetMeeting.
- **Notifications**: For collaborative/group email, notifications are key. Notifications could tell us when another participant has been added to the discussion or when new content has appeared, so that we can check out the gmail again to see the updated content. This is a very rich area for innovation.
    - The fact that the zaplets "update" notifications are from "Zaplet Notification" rather than the actual person updating (e.g., "Anoop Gupta") does not allow you to filter that notification, e.g., should you bother taking that opportunity to look at the update. Gmail notifications should indicate the person updating the zaplet in the "from" field.
    - Zaplet notifications may be on/off. If off, the user has to manually decide when to go and recheck the zaplet for updates. This saves them from a lot of notification email clutter (as discussed below), but they may miss timely announcements of important updates. If notifications are turned on, they come as email to the end-user. This results in numerous update emails, notifications and clutter in the Inbox and message window. The older email messages corresponding to the same zaplet are totally useless at this point, as *all* contain the latest updated information. This obviates the potential benefits of Zaplets. Gmail should provide more "intelligent" notification. For example, older notifications should automatically be deleted when a new one comes. Conceptually, there should be only one gmail in the Inbox per discussion topic, and that gmail should be brought to the user's attention appropriately (e.g., by bold text, and bringing it to the top of the sorting hierarchy by time, making it appear unread or flagged as updated…).
    - Gmail should have still smarter notification mechanisms … e.g., based on some of the priorities work being done in ASI. E.g., when and how the notification appears should be based on its overall priority based on importance of topic to me, who updated, what I'm currently working on, and how busy I am, etc.…
    - Gmail should link with peripheral awareness UI for update notifications. For example, the Collaboration and Multimedia group is working on the SideShow prototype. Dragging the

gmail to the "SideBar" will result in an automatic creation of a "ticket" and icon for that gmail. The icon will show how many updates have arrived since I last checked on that particular gmail, who has made the updates, ... We imagine simple UI that lets user control what information is displayed in the peripheral awareness display – some of this may also be part of the template. Such UI also addresses another problem with zaplets as they are today – people may turn notifications off because of clutter, and then the zaplet may get totally buried within the rest of the email, with the user losing track of updates to that topic. On the SideBar the user always know what are pending open discussion topics.

- **Server-Side Management**: Currently a zaplet seems to be an "active server page" on a web server, with the email simply containing a URL-link to that web page. The URL is automatically expanded/fetched when email is opened or when is the page is refreshed. Storing important discussions/gmails on web servers on the Internet raises many issues that need to be addressed to make gmail ubiquitous.
  - Web server vs. email server as store: Web servers with Active Server Pages (which is what zaplets are likely based on) are not very efficient engines as compared to an Exchange/Platinum store for managing content. For example, updating an existing email in the folders of recipients when a notification comes in is not particularly elegant in the Zaplets model ...nor is listing the set of people that have added comments since it was last read. Exchange/Platinum are also likely better at providing a unified data access model that resembles traditional email store, forward, search, delete, ... semantics for gmail.
  - Multiple distributed servers: Using Exchange/Platinum will also raise new challenges. Imagine that people on the To: list of a gmail are assigned to different Exchange servers. (Note: Replication of content is avoided for people on the same server.) Now the "virtually shared" content needs to be replicated on multiple servers and yet kept consistent across the multiple servers. Furthermore, when concurrent updates are made to the gmail, then we need to establish a global order in which all recipients see the changes. There are several possible solutions. One solution is to use a variant of algorithm used to maintain thread-structure in newsgroups, e.g., on UseNet. By having each update track and transmit initiating gmail UID, parent update UID, and its own UID, it should be possible for multiple servers to independently reconstruct the global order. As an alternative solution, we may use the notion of a "home server", where changes are serialized before being transmitted to other recipients Exchange servers. Thus each gmail would have a "home server" associated with it based on some criterion, e.g., the home server of the originator of the gmail. We can also accommodate a mixture of servers that live within the Intranet and out on the Internet, as long as servers on the Internet follow the recommended protocol. We need to do some protocol tweaks so as to make the latency of adding a comment appear low to a person who is not directly connected to the home server (slightly weaker consistency model???).
  - Diff-based propagation: We can reduce server communication load by just propagating differences between gmail updates between servers rather than the whole gmail content (e.g., send only the new discussion content).
  - Security: Imagine gmail between two people X and Y, where each is on an Intranet of a different company. Again, I think the multiple server model proposed earlier will work for gmail. As long as Intranet email addresses are used, the content will never reside on the outside Internet.
  - Deletion: See some of the discussion under the client-side management bullet.
  - Size: The Exchange/Platinum server can easily associate the size of the gmail to be the size of the referenced URL rather than just the link.
- **Miscellaneous**:
  - Making status of participants (available, unavailable, on-phone, in meeting,...) available within the To:/CC: line and ability to initiate a communication (phone, email, IM,...) with any participant in the group. Ability to record this interaction as part of the discussion within gmail. This could operate much like Messenger does today or be even more context aware (e.g., a la Priorities).
  - Expanded set of templates: E.g., include Office documents inline in the message.
  - "In-context Annotations": Ability to make "in-context" annotations rather than threaded discussions at the bottom of documents as zaplets provides today.

- Better integration from Messenger to Gmail and vice versa. For instance, access to your Gmail via Messenger, and the ability to send templates out via Messenger, etc.
- Social statistics and reputation information around who has read whose comments.