Cindy Charleson

From: Pete Higgins
To: Cindy Charleson
Subject: FW: Word retreat notes
Date: Thursday, July 15, 1993 7:26AM

pls print

From: Steven Sinofsky
To: Bill Gates; Chris Peters; Lewis Levin; Mike Maples; Nathan Myhrvold; Pete Higgins
Subject: Word retreat notes
Date: Sunday, July 11, 1993 6:27PM

These are my raw notes from the Word retreat. Hope they help serve as a reminder of things we discussed. Feel free to forward these as you see fit, or to write up something a little more formal. This is mostly a list of issues raised in the various discussion groups.

I have marked really important thoughts (my opinion) with '***'. Any errors are of course my own note taking.

Overall the two big themes were structure and online docs. Structure in terms of (a) SGML imposed structure and (b) using structure to assist the user in developing professionally formatted documents and doing more automatic work. Online doc issues were (a) proliferation of redundant work for solving parts of the issue and (b) user-interface issues.

Group 1 - Structured Documents
==========
Implement conditional text (needed at the very least for our user-ed)
Make outlining easier to use by taking cues from the users structure
For single sourcing help/print there is a big issue of going from linear to structured docs
Is structure a high end feature only (yes, in terms of using SGML-like tools, but all documents have structure)
Ex post-formatting would be a big win--turn my document into a professional document (like autoformat but use more of a model based approach)
Recognize based on structure things like phone numbers, social security numbers, etc.
Every time the user introduces a new format, tag that as a style in order to make use of it later in a structured manner (BOLD == Emphasis 1 style). This would take advantage of the fact that direct formatting is still the most popular way to achieve a desired output
Do we need another namespace for the structure or can we use the heirarchy we sort of already have
**Eliminate outline view and use a gutter like Excel
**Enhance the tagging engine to learn by example
**Make the tagged unit of text something the macro language understands and manipulates and possibly OLE monikers
**Developer version allows the extension of the auto tagging rules
**Add picture formats to autoformat
Implementation: use Fields as the way to implement these new strctured entities
Nathan: get ChrisP information on structured docs from the IBM/Almaden researcher [me too]

Group 2 - Online Documents
==========

Bill put up a big (yet still partial) list of editors and authoring efforts going on at Microsoft.
  (headcount was estimated at 100--2x the number of Word developers)
Many, the group felt, evolved because of a lack of online document support in native Word.
Benefits of online documents: distribution, searching (indexing, content structure), links! (including monikers), and animation (better name would be 'code' since the idea is that there is code behind the document)
Issue: What is the U/I for reading documents on the screen? History, outliner, fonts, speed, scrolling, layout elements (footnotes, headers), etc. Can we use the extra horizontal width for showing hierarchy or other navigational aids like MSDN? PPT is an example of an app that has a specific "view" mode (slideshow). What would Word's look like?
Issue: Word as a viewer is very large; output device issues and consistency in printing; the online viewer should be a small subset of even output functions since perfect layout costs lots of code.
Competition:
  Acrobat (printed page is God, problems when trying to understand underlying structure such as hyphenated words).
  Common Ground (printer driver based has the same problem). Notes is enhancing the built-in word processor.
  WordPerfect shipping Folio-Lite.
  Ami - subset product
  Frame
  PLS - searching
Action items:
  Some work in Word 7, a lot of work in Word 8
  Indexing work (Cairo)
  On-screen reading
  Rationalize various authoring/editing efforts
  Tell the world our vision viz competition
  Determine super/subset relationships of the universe of MS viewers/editors

Group 3 - Applied Intelligence
==========

"Elephant model" - remember what the user does
Make Word more adaptive and model both the task and user
Environment issues and Word: autosave or persistent files
Writing:
  auto grammar/spell check
  learning auto correct (common spelling errors, common backspace mistakes)
  auto indexing, toc, hyperlinks
  auto conversion of ideas (3x5 cards) into document structure
  make sense of sloppy writing
Formatting:
  personalized learning in the background
  better templates (like powerpt)
  print v. online layout issues
Design Assistants:
  What are you going to write
  Adapt U/I to task (change menus, toolbars)
  Example: AutoReport
Issues:
  Learning what you do is not so great since you do dumb things--need to learn what you mean
  Auto-remove unneeded U/I elements (menu MRU) for user
  Autoformat needs to learn from a sample document not inference
  AutoAuto
  Utopia is the right direction, but part of the swing between modal and modeless (get the clown to shut the fuck up. Other. Other. Other. Spell-check) [line of the retreat]

**Apps need to realize that they are the ones with the power to set standard user-interface and add to the style guide. Need better coopertation. The ability to subtract from the style guide does not appear to be anywhere. What is a toolbar? Is it a shrunken modeless dialog? Why isn't the u/i group in on such decisions?

Group 4 - Synergy
==========

The group assumed program managers would continue to do opportunistic things. This is the 5 year view.

All text is Word.
All tables are EXCel.
Interoperability is gained by deconstruction.
When you buy one MS product you want/need to buy more.

How do we do this?
- break products into components services
- pick owners for services
- provide levels of services
- package services as products
- basic services are widely available to other apps (from the OS) but we need to be sure our competitors don't benefit too muc from these services). En example here is our fear of installing basic converters in the OS

Basic services need to be used in OS--file formats, U/I
Table objects in word need to be programmable just like XL sheets today
**Need to market the idea that Microsoft has the best Windows applications (need a compelling reason)
New services/features:
- C compiler integration (program editor uses Word)
- Customization spans apps (toolbar customized in Word is reflected in Office)
- Optimize for office at the expense of catgory features
- Page layout should be treated the same across applications

Office Shell is a great way to make progress on this front.
There were many other opportunities for commonality that failed not because we couldn't agree, but because we never tried (print/layout model)
**Word/Mail integration is the single most important Office Shell feature
**Word 7 will need to expose the basic Forms approach (especially for mail: To, From, etc). Also need to promote fields and properties to/from summary information (like purported Application Field Exchange in Notes)
Issue: Word/Publisher integration - are we doing enough
Issue: Word/PPT integration (why can't PPT be written in Word or VB?)
There was a general discussion about the low quality of printer/video drivers and how much of a problem this creates for all development teams (PPT is especially suseptible to this)

**The key to real synergy is one piece of code to do the same thing across applications. One program manager. Do we want to have middle feature teams that span word/excel? The organization needs to change.

Group 5 - Research and NLP
==========

There was some general information sharing by Karen about the state of NLP in general and the work at MS. She feels the time is right to commercialize their efforts and is working very hard with Word to make sure that happens.

The levels of NLP Karen defined from "Star Wars" to today's technology:
- discourse
- pragmatics
- concepts
- senses

logical form (MS has a fair start here)
syntax (MS has a good start here)
morphology (MS is very solid here)

THe possibilities for using NLP is the next releases of Word include:
(1) intelligent grammar checking
(2) intelligent grammar checking with alternate phrasing suggested
(3) email quality machine xltation
(4) summarizing/shortening documents

(1) is very realistic within the next two years. The Word team will assign two people to work with research to make this a possibility. 2-4 become harder as work needs to progress beyond the "logical form" to senses/concepts/pragmatics/discourse.

There are some general issues with regard to moving research into the product groups. With Word the process worked as follows:
(a) identify common areas
(b) move product headcount to research (in a directed manner)
(c) narrow the focus and deliver product group code

This process needs to be fostered with an annual meeting where all the lead program managers and developers in the product groups can get a better idea of where research is. Also the product groups need to do a better job at identifying and either pursuing or getting research to pursue interesting ideas.

BillG Wrap-Up
==========
This retreat is a good thing to do--continue on a yearly basis
Look at DTP to make sure we are stealing all the cool features without confusing users or making Word hard to use.
Everyone should participate in Advanced Technology issues--what is the mechanism for tapping this creativity (Bill gave example of interns with cool ideas)
Explore new input technologies
**The real challenge is making sure the marketplace understands how good Word 6 is in comparison to Ami.