

Cindy Charleson

From: Pete Higgins
To: Cindy Charleson
Subject: FW: Excel 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: Excel retreat notes
Date: Sunday, July 11, 1993 6:27PM

These are my raw notes from the Excel 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.

The big theme of the Excel retreat was Structure--how to obtain it and then how to take advantage of it. This theme also came up a lot at the Word retreat. There are some synergy issues with outlining and other structure issues (styles, etc) that should be addressed.

JonDe implementation overview
=====

A number of issues were raised during this talk:

**maintain units for numbers

move recal dependents from list to a tree (from Kruper's description this is being done implicitly anyway)

Do we need an Excel specific font (easy to do)

How do we make the save command go away?

Improv Demo
=====

The toolbar/dialog thing is very interesting. Is it a direction we wish to head?

The duality of the views can be very helpful

What is the linguistic strength of Lotus Script [here is a somewhat interesting presentation that MikeRis gave a while back--I'm sure he can get more updated information, though I don't think anyone has looked at the language per se]

<<File Attachment: LSCRIPT.PPT>> - Attached

Group 1 - Data Modeling, Access, Analysis
=====

Modeling

simulation - most models are simulation

extract structure from existing sheets and then use to assist users by correction and auditing

spreadsheets are used to create things based on what the end report should look like (this is the power of the model so we should leverage it)

need smart ways to extract structure

need to create the outline for the user

**need a model that says "I have selected ONE 'thing'" now manipulate this whole list of things
shared formulas impose structure - how can we leverage that

**auto strong typing

english language formulae generated for user

need better unification between XL and Access -- why do we copy the data to access to create a
report--because XL isn't a good reporting engine

XL competes with flat file DB by adding more rows and a report writer/wizard

if the column headers were moved to a different sheet in XL5, then this is the same as the schema
definition in Access

**build reports that generate a spreadsheet outline so they can be interactively browsed

**need a way to associate formatting with a data-type for easier reporting

need annual report reporting quality

**extend pivot tables to be more like consolidation

Data Access

better parsing of input data from random source (Lewis's customer example)

normalize existing sheets or incoming data

take advantage of NLP for Query tool (NLI work)

**need a better tool for exposing schema to users

use real NL formulae

(example from NumberBuddy of formulae being represented as machine tapes)

Data Analysis

we supply functions but users have no clue how to do them

do we have statistics wizards that explain basic principles to users

forecasting - wizard to tell user what to do when a new dataset shows up

**AutoShovedown when adding new rows

**Need to do a better job at working with time ordered data

Group 2 - Workgroups and Programmability

=====

Programmability

Various types of programmability available to users today:

end-user based

ISV (solution provider) based

capsules and data flow

Needs of ISV programmers:

need to turn lots of stuff off

add cell-specific information (like a phone#)

restrict U/I (this is getting harder with proliferation of U/I -- every new element requires specific code to
turn it off)

**ability to replace (subclass) behavior - for example FileOpen (this requires specific events: intent,
doing, done)

Features:

range/sheet subroutines (after some discussion many concluded that the sheet level is acceptable to
this, especially if there is a way of structuring the sheet so it is merely a view on the sheets below it)

CASE - is there a way for a high level description to drive the creation of a SS

SuperRecorder:

alerts when something might fail

inserts template VBA code for error handling

loops

**toolbar for Pause, Look at Code, insert loop, etc.

**better hooking: supply key routines externally, command objects, need better forms such that parts of Excel (key parts) are available in this customizable forms model

Monolithic nature of Excel is a hinderence to this (footprint on disk for runtimes prevent vertical solutions)
Setup: keep portions on network and ask for the mas requested, delete unused portions after a certain period of time. Keep track of where to find key components
Spreadsheet compiler: can XL be available as a .LIB? [Has anyone seen Visual Baier yet? I saw it as PC Expo]
**Need an event model

Workgroup Issues

multi-user editing and concurrent access

**change logs: Word and Excel can share experience here since Word has lots of work on making this work and be efficient

Routing: serial and broadcast exist in XL5, but need to add tracking and email rules

**Get serious about Diff/Merge and change logs, split/combine sheets

Need both change logs (revision marks) and diffs

Ability to declare a SS as "shared" which will keep all changes in a different stream

**Enhance navigation by having link between formula and data (similar to how charts do it today)

Attach code to cells for update events

Multivalued cells

feedback loops: approval, comments, tracking, changes, notification, auditing

obtain organizational process to auto generate this feedback loop

programming: check in/out, code documentation, document structure

Group 3 - Synergy with OS and Apps

=====

Synergy with Windows

share visuals (shell should have the same icons, for example)

Shell (Explorer) should use autofilter to navigate files

Word selection model in text control in shell

Excel scratch pad

**Excel Applet or viewer

Push .XLS as the data exchange standard (viz Acrobat) but problems with printing since that takes so much code it is hard to use that as the viewer

**OLE boottime technology problems

**Office shell preloads app code in idle loop

**Office shell uses title bar (ISV compatability problems)

**Use SDI to be one step ahead

Synergy with Other Applications

**ChrisP grid of death (Golf and SQL Server interaction :-)

Office applications

annual report scenario

Word owns header, footer, water mark, etc

**cross app wizards based on task scenarios

Excel needs better page layout but can't use Word unless word is made to handle things like very wide pages

System-wide UNDO/change log

Merge word/xl to a single EXE

Group 4 - User-interface advances

=====
**Keep in mind that there is a UI group that is waiting for apps to talk to them. They have written down and tested many ideas (no top-level menus, move to popup modeless dialogs, etc). But there is no proof that this really works better.

blank sheet problem:

** one way to deal with this is to provide lots more sample sheets and tie them in with documentation and help system (learned from Windows SDK that more is better, even though users expect MYSHEET.XLS to be in the sample code)

Provide extra sample code in a "developer" edition and charge more

Have code behind sample docs so you can point to something and have it do that to your own document

QuattroPro - works with your data and not just samples in dialogs and other places

Discoverability

**balloon help could be really good if done 100% and made faster (Mac was faulted because it was slow, poor graphic choices, and because no body did it everywhere)

**IntelliDraw (Aldus) unifies all various help systems to one box, which could ease clutter and save on authoring

Intelligence

autodetermine correct chart type (build in a cross between Tufte and USA Today)

calculated formats based on data (dates, currency work today)

need to obtain structure

Structure

detects lists, time sequences, etc

sheet subroutines (started off as range subroutines)

replace cell level protection by sheet level protection+views

auto outline detection

**type/subtype relationships added to cell formats

>> range checking, constraints, dimensional analysis

**step mode for F9 recalc

**sticky autocolumn width

**why does printing ever show #####

Group 5 - Presentation/Visualization

=====
Visualization: want to use Excel online everywhere

Today charts are used for presenting information, but the real big use of charts is for discovering information

**Time series are candidates for animation so do it

3D modeling and rocking/rotating (scatter charts)

User customizable charts (in XL5)

Transition effects for charting based on metafile (better than ppt)

Interactive charts and viewing (change to log scale easily, for example)

**programmable charts

graphical abstractions required

chart would be composed of graphical elements-the test of this is the ability to do "bubble" charts

make every aspect of a chart a calculated, hence customizable, value

chart needs knowledge/structure

time periods (daily, weekly, monthly)
GIS information (make use of a CD and ship useful data sets)
understand more about user's data
histogram based on sort order (everytime the user says Sort he/she is telling you something),
similarly Pivot really tells you a lot
make ability to teach XL about structure explicit so it can be 'programmed'
big data sets are important for discovery and visualization (100,000's data points)

BillG comments

=====

Issues raised by Word retreat:

receivers of documents need a viewer. how good a viewer is excel? how much functionality?
improve XL's read mode
user tracking (ATBD work)

Marketing

**clear differentiation, need a separate identity to avoid price wars MARKET LEADERSHIP

Programmability

language exposes internal data structures, uses VBA forms
workgroup: merge, diff, split, combine

Synergy

office shell programmability
systems/apps - need U/I to be the same
shell should dive into apps file structure [need to structure docfiles to take this into account]
expose properties from Summary Info for "field exchange"

U/I

need features that can't be done in macro language in three months by competitors

Connect Excel to constraint based drawing package (someday). There should be a simple protocol for two-way communication and really good samples that show the power of this.

Action:

have these meetings regularly within BU
internal structure reforms of code are a great thing
program mgrs need to come up with inimitable features