Debra Vogt

From: Greg Whitten
To: Bill Gates; Brad Silverberg; Jim Allchin; Paul Maritz; Pete Higgins
Cc: Mike Maples
Subject: Object Technology and Chicago
Date: Friday, September 25, 1992 11:40AM

You can choose to distribute this or not. I don't want my comments creating any more confusion than there already is today.

The following are two scenarios for the state of our object technology when we ship Chicago. The first is a minimum risk strategy that we are following today. The second strategy is a little more aggressive and forces the various product groups to provide a more uniform level of support for the technology. A fallback would be to achieve at least this second level for Cairo; however, there is not any focus to even plan on doing this.

Today's plan for Chicago

OLE 2.0
- infrastructure - object model, activation, storage, and monikers
- in-place editing
- drag-drop and new transfer model
- IDispatch for programmability

OLE 2.01
- draw layer support
- TBD for getting commitment to producing draw and annotation servers

Object BASIC
- host application interface
- can call IDispatch interfaces (remote call)
- can bind to interfaces (remoting support is TBD)
- can NOT create new object classes

Applications (Excel, Word, ...)
- support OLE 2.0 and Object BASIC
- support IDispatch for programmability
- TBD is support for OLE 2.01 function

Chicago
- ship with at least OLE 2.0 support from above
- ship with stripped down VBASIC with OB 1.0 technology
- TBD support for OLE 2.0 with applets
- TBD support for OLE 2.0 with new or existing shell
- TBD support for new applets and OLE 2.0/2.01 servers

Multimedia
- TBD support for updating MM OLE 1.0 servers

In my memo Programming Model for OLE 2.0 Applications and Objects I covered a number of recommendations in Section 9. My conversations with the OB team, OLE team, and PaulMa/BradSi/DavidCol have left me with the distinct impression that it will be hard for them to push for anything which is outside of the above. This means that the OB-app-OLE issues that I raised will not be able to be addressed in the Chicago timeframe. To do more would require the "different groups to work outside of themselves" to quote one person. I would add that it would also require that management actually assign resources (real people - not just open headcount) to address the open problems which would potentially take away from other product features.

What we could have if we devoted resources to specific problems

The second strategy would require us to get a little more serious about the level of object technology that ships with or just after Chicago. This means committing to a specific set of objectives so that we have

multiple components of our system supporting the same things. Broad based support for a technology sends a better signal to our customers than lots of depth in technology in individual components. The breadth gives the customers a better guarentee that the desired interoperability of the components will be present and can be useful to them. We don't have a plan for either Chicago or Cairo that says what this broad based support entails. Products that do sign up to interoperability goals today believe that support for individual parts can be dropped unilaterally. This only weakens the story for all the other components (end users who purchased other parts of our "system" expecting uniformity in our support).

1. OLE 2.0/2.01 support
   applications and system components as containers
   applications, system components, and OB as in-place servers
   multimedia in-place servers
      standard control interface (VCR like)
   broad based annotation and drawing support

To do a better job with this will require that the OLE team help or jump start key products with their feature designs for OLE. The OB-host app interface is onekey area. Our multimedia efforts could use some more help, as well. Authoring difficulties for multimedia can be somewhat relieved by the appearance of more containers for multimedia objects. This pushes more of the problem back into the domain of just being easy to create individual multimedia objects.

2. programming model support (covered in more detail in Section 9.1)
   should be supported by applications and system components
   develop generic application/document/"containment" models
      including view and data separation
   look forward to Cairo for generic interfaces for exposing
      application information in a homogeneous fashion
      and support it in our application modelling
   standard interfaces for time/frame-based control
   see Section 9.1 for other issues

3. application/object customization model
   OLE and programming interfaces should allow OB "programs" to be
      embedded into a document (can't do this with OB 1.0 plan)
   the following should be supported by apps like Excel/Word/shell?
   standardized interfaces for customization
      including exposing new methods for customized operations
   OB should support creation of new OLE object model classes/servers
   application/object default initialization should be part of this
      want to provide more robust customizations than today

4. exploiting the transfer model to increase interoperability (see Section 6.2)
   the current definition of OLE 2.0 allows reasonably rich visual construction
      of information documents/applications
   it is possible to use the OLE naming architecture to provide richer
      data level connections and transformations of information
   a transfer tool can be designed that will open up the naming architecture
      and allow more general monadic, dyadic, n-ary operations
      on information when connected to a programming language
      like Object BASIC
   look at existing app to app transfer, move to new transfer model, and
      fill in some of the missing conversion possibilities

5. forms model
   needs immediate ownership
   need to determine how to layer or combine OLE server, control, and VBX
      interfaces
   develop a text control that fits into the above model

6. Chicago component design (also very useful for future small 32-bit Windows)
   need to look at the following issues
      how to shrink working set by reuse of components
      OLE 2.0/2.01 support

programming interfaces
how does the object move up to Cairo
against the following components
file manager
program manager
control panel
packager
help
text control
packager (alias links)
DOS box
new applets/servers - draw/annotation/multimedia

I think that we should devote a little more effort in 1 to 4 above to increase the consistency between the components that we deliver. This work translates into products that work together better, form a more coherent solution for office desktops, and should help us sell more complete office solutions. The standardization in the area of programmability should be very important to 3rd party ISVs and MIS organizations in using our applications to be a part of mission critical applications over the next several years before we move more strongly into providing finer grained component solutions.