From: Rick Rashid
Sent: Friday, April 14, 1995 4:17 PM
To: bens; danli
Cc: adamr@microsoft.com; bradsi@microsoft.com; johnlu@microsoft.com; paulma; thomasre@microsoft.com
Subject: RE: Internet (aka Web Windows)

I agree that we need to support the teleconferencing applications, but its worth noting that there is a real difference betwe that and the "remote virtual desktop" which Citrix and others are beginning to provide for NT. They really address differe concerns.

If we could encorporate (perhaps as a control or mime type) a "remote desktop" interface to NT servers through our browser it would allow people to do seamless transitions between web browsing and telecomuting and would allow servic companies to use Web pages to provide specific Windows application services. For example, I could be accessing Microsoft's Web pages and there could be a url to a group of NT servers which would allow me to run an arbitray Window application within that environment when I followed the link. This could be an Office-based application, for example, a V app or virtually anything else. I could, for example, use this to provide Web access to an existing database application w potentially sophisticated user interface that would be hard to convert into a "Web forms submission".

Imagine a company which is using an internal Web as a corporate information network. Such a company may want to pu specific NT server-based applications on line for its employees. An employee could browse until he/she found (for example) a page talking about company benefits and there would be a url which could invoke a server application which t employee would use access the benefits database and make protected changes. This could be preferable in most cases downloading the app, setting it up and then running it locally against a backend server (like we do today for United Way donations).

For "telecommuting" this approach could provide a URL link which would allow someone with the right password and user to use the Web to connect to a particular server.

The advantage of the approach Citrix and other NT-based systems have in this arena is that they've handled the issues o security within the server plus the notion of a real remote desktop rather than just shared apps and they require very little mechanism on the browser side.

Given the existence of some of these systems already it may be possible to just license them and get this functionality up quickly as a feature our browser could invoke.

Again, these uses are different from the teleconferencing uses which are also very valuable and should be addressed as Ben points out.

-Rick

| From: Ben Slivka
| To: Dan Ling; Rick Rashid
| Cc: Paul Maritz; adamr@microsoft.com; bradsi@microsoft.com; johnlu@microsoft.com; thomasre@microsoft.com
| Subject: RE: Internet (aka Web Windows)
| Date: Friday, April 14, 1995 3:01PM
|
| X-Exchange-Message-ID: C=US;A= ;P=MICROSOFT;I=RED-PSD-MSG950414150111KNX004C00
|
| rick, dan, adam, and I met thursday afternoon to discuss what we could
| do help Windows remain relevant in the Web world. We came up with
| the following:
| 1) adamr should look at extending his conferencing stuff to permit running
| Windows applications remotely -- you would run a particular client piece
| on a Windows client machine, and it would work against a real Windows
| app running on a server machine (most likely a Win32 app running on

| Windows NT, for scalability and security reasons). While we could look at using the Citrix stuff, that technology seems a bit less maintainable, and unnecessarily divergent from the conferencing hooks. On the other hand, adam does need to get a demo to see what the performance is like.
|
| 2) bens needs to look at what kind of enhancements we can make to the Microsoft Internet Explorer to really leverage the Windows platform:
| a) Add HTML tags for all of the controls in Windows 95 (list view, tree control, color picker, toolbar, etc., etc.) -- these can give a page a zippier look and feel, can be done very easily (since the Win95 controls are already there), and raise the bar for our browser competitors.
| b) Quickly design and implement a secure, extensible control architecture for Internet Explorer. Conceptually these are just Windows Controls (as opposed to more complicated, slower OCXs) that either: i) take over the entire client area of the browser (kind of like the docObject thing we've discussed with Office), or ii) can exist in a rectangle on an HTML page. ==> We'll create an HTML "client" control from the browser, and have the VRML folks at Intervista write the VRML "client" control (they're excited about this, I met with Tony Parisi today).
| c) To simplify control distribution, we would have GUID and versioning information either in the HTML doc that needs the control, or maybe even have a central registry at www.msn.net. So, when the browser sees a control it doesn't have, it just fetches it from a canonical place at www.msn.net.
| d) The control is digitally signed by MS, and we use the same UNWRAP.LIB as we're already planning to use for the auto-update feature.
| e) Permit client-side scripting in the browser -- provide a VBA subset that is "safe" -- i.e., no file I/O, registry edits, etc. that could in any way damage the local machine or transmit information from it out to the web. VB really lets us leverage the investment in VB learning that corporate customers have made.
|
| 3) bens needs to add editing ability to the browser at some point, and figure out how we can take advantage of existing MS technology to support spell and grammar checking.
|
| 4) paulma -- our web server strategy is very disorganized and weak right now: BlackBird is not focused on the Web, NT is doing just a basic web server. We really need to focus on great authoring tools, web site management tools, and cool servers.
|
| 5) paulma -- we have to really evangelize the server stuff & control authoring so we make sure the ISV community is focused on delivering great "Windows" web applications.
|
| Note: While we should probably support OCXs in the above framework, they should not be our default implementation choice.
| --bens
| ---
| |From: Rick Rashid
| |To: bens; danli
| |Cc: paulma
| |Subject: RE: Internet (aka Web Windows)
| |Date: Thursday, April 13, 1995 10:00AM
| |
| |I think the key points here are:
| |
| | 1) Today the Web is not a friendly place to write

| |"interactive applications". Browsers (e.g Netscape's) will
| |address this by adding APIs and they are alreaddy beginning to
| |do this at a dizzying rate (e.g RealAudio). We have to do
| |something about this or the Browser becomes an OS-Independent
| |"Shell" and the Browser APIs become the application platform
| |interface.
| |
| | 2) We have a critical asset: access to a huge ISV
| |community and a collection of time tested APIs and tools for
| |building applications.
| |
| | 3) How can we leverage our assets to create a Web-based
| |API for interactive applications?
| |
| |My first cut at this was to say: how about just adapting our
| |API for the Web? Obviously the Citrix people have done part of
| |this as have the teleconferencing people (e.g. the PictureTel
| |code we just bought).
| |
| |Dan Ling has separately pointed out that we could develop a new
| |or at least "related" API -- perhaps based on the use of
| |cached, downloaded, digitally signed (for security) OCXs with a
| |Web protocol for interacting with them.
| |
| |If you take the controls idea forward you could accomplish a
| |great deal with relatively straightforward work. We could, for
| |example, provide OCXs for 3D animation using our Rendermorphics
| |assets, 2D animation using our DirectDraw assets, 3D sound,
| |etc. along with a lot of more obvious traditional controls
| |which could be interacted with over the net. We could even
| |distributed CDs full of controls for ISVs to use and users to
| |put in their machines to avoid downloading. We could create a
| |whole "Web Controls" infrastructure with specialized versions
| |of VB and VC++ just for Web apps and special NT mechanisms for
| |invoking apps, security hooks, etc.
| |
| |
| |
| |
| |
| |-Rick
| |
| |--------
| || From: Dan Ling
| || To: Rick Rashid
| || Subject: FW: Internet (aka Web Windows)
| || Date: Thursday, April 13, 1995 9:33AM
| ||
| || Continuing to point out the pitfalls: Note Ben's comment at
| || the very end that Window's apps don't look cool ..... the
| || apps that look cool (e.g. Encarta etc.) aren't the ones our
| || ISVs know how to make. So we have a complex infrastructure to
| || make not so cool looking apps. I think we need to accelerate
| || the new cool things e.g. 3D graphics with animation, video,
| || audio etc. rather than focus on the vanilla text-based Windows
| || apps. But if we focus on the new stuff rather than the old
| || stuff.....shouldn't we just design the app to be remoted
| || rather than messing with the Windows event queue??
| || Dan
| || --------
| || From: Paul Maritz
| || To: Ben Slivka; John Ludwig; Dan Ling; Rick Rashid
| || Subject: FW: Internet (aka Web Windows)

Date: Thursday, April 13, 1995 9:03AM

X-Exchange-Message-ID: C=US;A=
;P=MICROSOFT;I=RED-10-MSG950413090347CNX00D400

Bens - you may want to see if rashid and dan ling can spend few minutes with you to brainstorm. I would be happy to join, but don't let me be the bottleneck. I am meeting with Rick at 2pm today - if we get done early, I may try to give you a call.

Probably the easiest thing to do would be to call up Citrix and ask them to actually set up a live demo. Their stuff supposedly works over whatever LAN one has (and thus over TCP/IP), so it is a question of getting a copy of their client code up here (which supposedly works only over DOS now), getting it set up over TCP, and connecting thru to a server of theirs in Florida.

Ed Iacobucci's email address is edi@citrix.com. He has been telling Rick that a "beta is in the mail" for weeks now.

From: bens
Sent: Thursday, April 13, 1995 12:59 AM
To: johnlu; paulma
Subject: RE: Internet (aka Web Windows)

Having pooh-poohed this idea in paul's office, I now back peddle and agree that we should post-haste determine the feasibility of this stuff. As paul suggests, there are two key issues:

1) lag between mouse/key events on client and visible response on client from server -- this is the problem that lead the OS/2 PM folks to dismiss X-Windows as not high enough quality for a PC.
   ==> We could do a simple system hook (I believe) to insert random delays in sending mouse &keyboard events on a local system, and this would be a very close approximation.
2) while GDI commands may be quick to transmit, when the app gets into blitting mode, then you're in "download the gif" hell. Think about toolbar buttons, images in dialogs, etc.
   ==> Again, we could model this by hooking BitBlt and StretchBlt and introducing a variable delay that is linear in the size of the bitmap. If this delay provide problematic, we'd have to get into the client-side image caching business (ala the persistent cache we have in our Internet Explorer for gif, jpg, html, etc.)

We should get some smart USER (scottlu) & GDI folks together and talk about this stuff, but only after someone has sat down and watched the Citrix stuff in action.

Now, having said that, let me point out that our existing Windows UI is pretty ugly -- grey buttons, black text, few graphics. Even if we do Remote Windows, web pages simply look cooler...
—bens