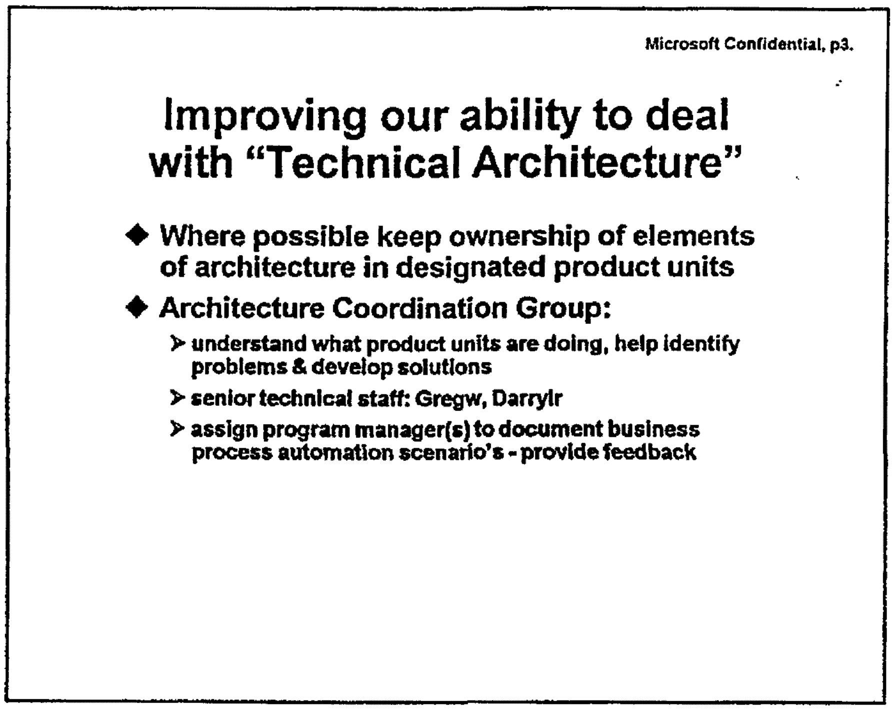
Technical Synergy Meeting - 1/14/94

- Occasion: 3 year plan cycle, formation of architecture coordination effort
- Challenges/Opportunities facing MS: specifically common architecture & "business process automation"
- Introduce Architecture effort:
  - Inventory of Architectural Issues - right ones? priorities?
  - Decide who are "owners" for next steps
  - Get feedback/suggestions

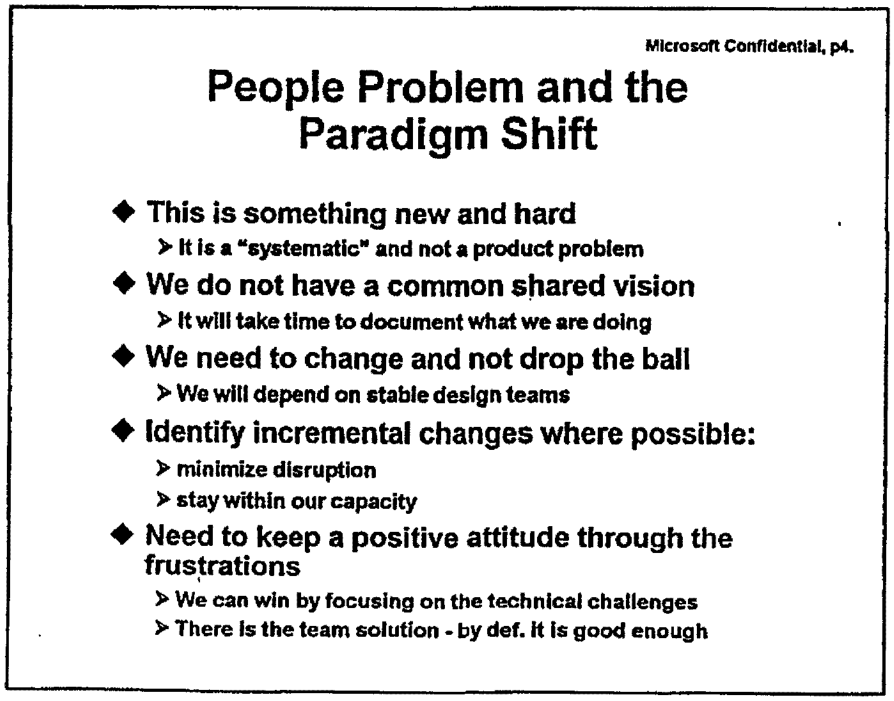
Microsoft Product/Business Challenges

- Commodization:
  - Competitors now have comparable applications
  - What will customers value long term?
- Area's where MS is weak:
  - group productivity (Notes), group development (Powersoft), network server/services (Netware)
  - can/will cause platform shifts that will further disadvantage MS apps and systems
- Internal efficiency/effectiveness:
  - use of common code & services
  - investment in long-term architecture



Improving our ability to deal with "Technical Architecture"

- ◆ Where possible keep ownership of elements of architecture in designated product units
- ◆ Architecture Coordination Group:
  - ➢ understand what product units are doing, help identify problems & develop solutions
  - ➢ senior technical staff: Gregw, Darrylr
  - ➢ assign program manager(s) to document business process automation scenario's - provide feedback



People Problem and the Paradigm Shift

- ◆ This is something new and hard
  - ➢ It is a "systematic" and not a product problem
- ◆ We do not have a common shared vision
  - ➢ It will take time to document what we are doing
- ◆ We need to change and not drop the ball
  - ➢ We will depend on stable design teams
- ◆ Identify incremental changes where possible:
  - ➢ minimize disruption
  - ➢ stay within our capacity
- ◆ Need to keep a positive attitude through the frustrations
  - ➢ We can win by focusing on the technical challenges
  - ➢ There is the team solution - by def. it is good enough

Are MS products valuable to customers?

- ◆ This addresses MS products aimed at "professional market":
  - ➢ products by: Systems, DAD, DDT, WGA (not consumer)
- ◆ Approach taken:
  - ➢ Interview variety of customers ("activity based planning" at a very high level):
    - ⇨ Small (approx. $1M revenue): Gator, Crossroads
    - ⇨ Medium ($5-$30M+): Prologic, Schultz/Miller, Boxmaker, Warren Supply, Physician's Micro
    - ⇨ Large ($100M+'s): Nationwide, Bankers Trust, Chevron (Canada)
  - ➢ Write-ups available on \\NTSRVR\INFO\3YRPLAN\PLAN93 (contact Kayb if access is denied)

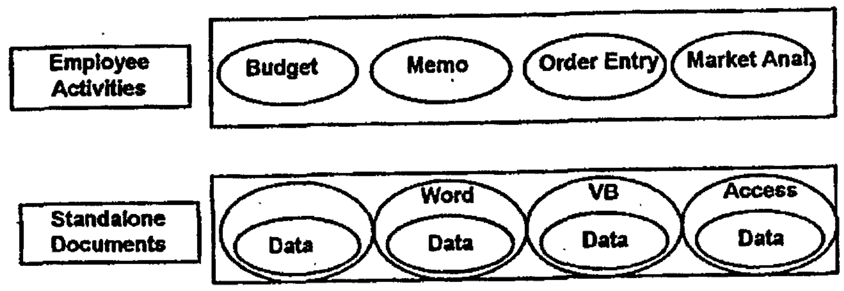
Macro Conclusions:

- ◆ Processes vs. Documents
  - ➢ Businesses (large & small) are fundamentally about <u>processes,</u> eg.
    - ⇨ take & fulfill an order
    - ⇨ register a new employee
    - ⇨ approve a loan, etc.
  - ➢ What is valuable to business (i.e. will pay non-commodity $'s for) is the <u>improvement and automation of these processes</u>
  - ➢ Most business processes are built around <u>shared data</u> (customer lists, project plans, orders, etc.).
- ◆ Once elements of a solution/process are in place - very slow/hard to change, products get locked in = opportunity & threat

Macro Conclusions:

- Current MS products focus on <u>individual documents</u>:
  - ➤ our current apps view documents primarily as "stand alone" or as an end in themselves; not an integral part of a process:
    - ⇨ it is hard to share and re-use data between the documents that represent different steps/aspects in the process.
    - ⇨ do not adequately address the customization (development) inherent in automating processes
- Customers want both: to address document preparation in context of automating their processes

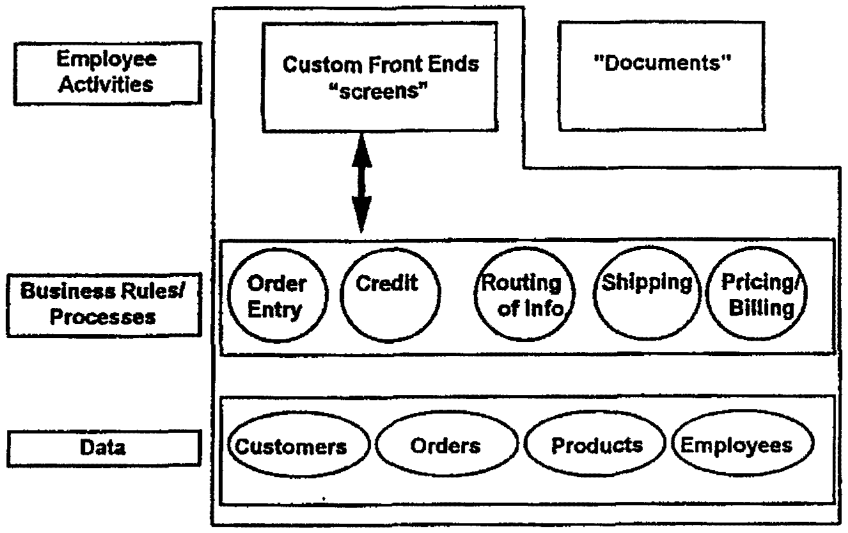
Microsoft World View



Microsoft Confidential, p9.

Current Customer State - 1

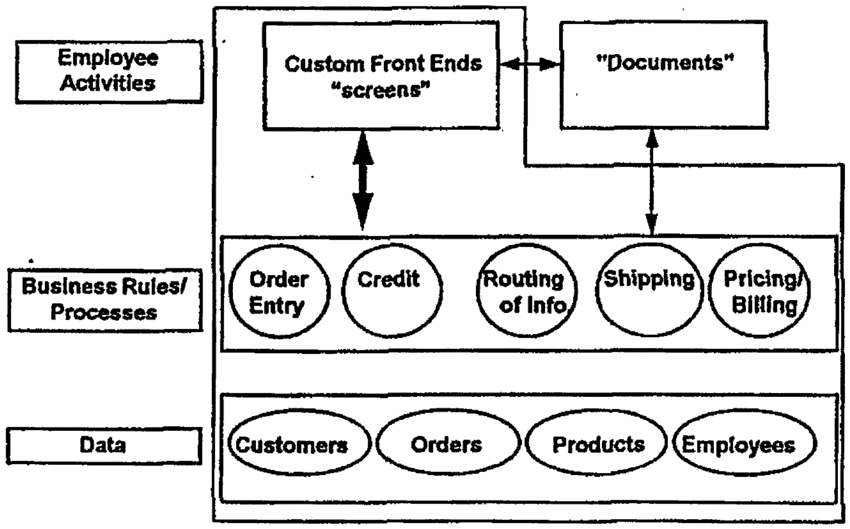
2 separate worlds



Microsoft Confidential, p10.

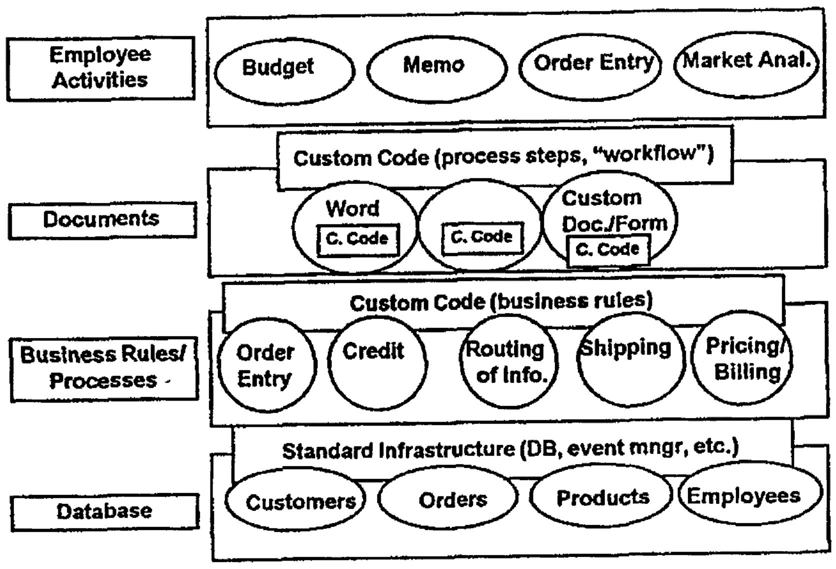
Current Customer State - 2

Partial Links: eg. 'Cut & Paste' bridge between worlds

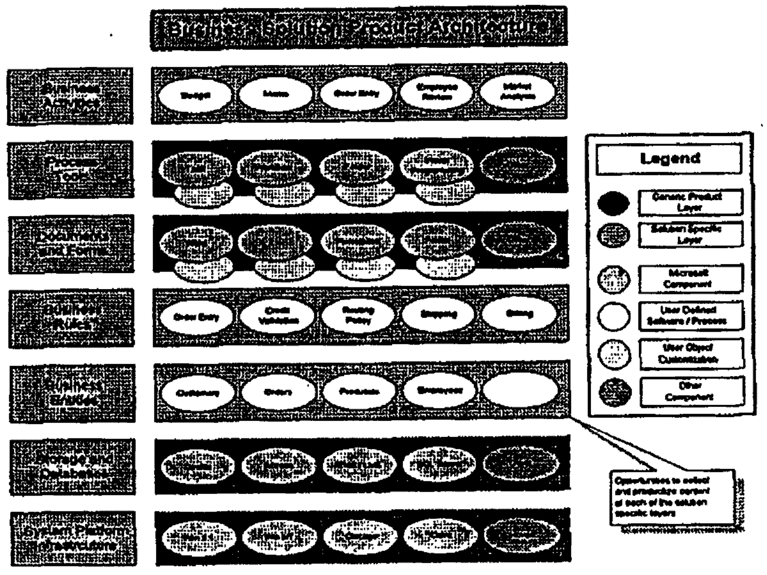


Microsoft Confidential, p11.

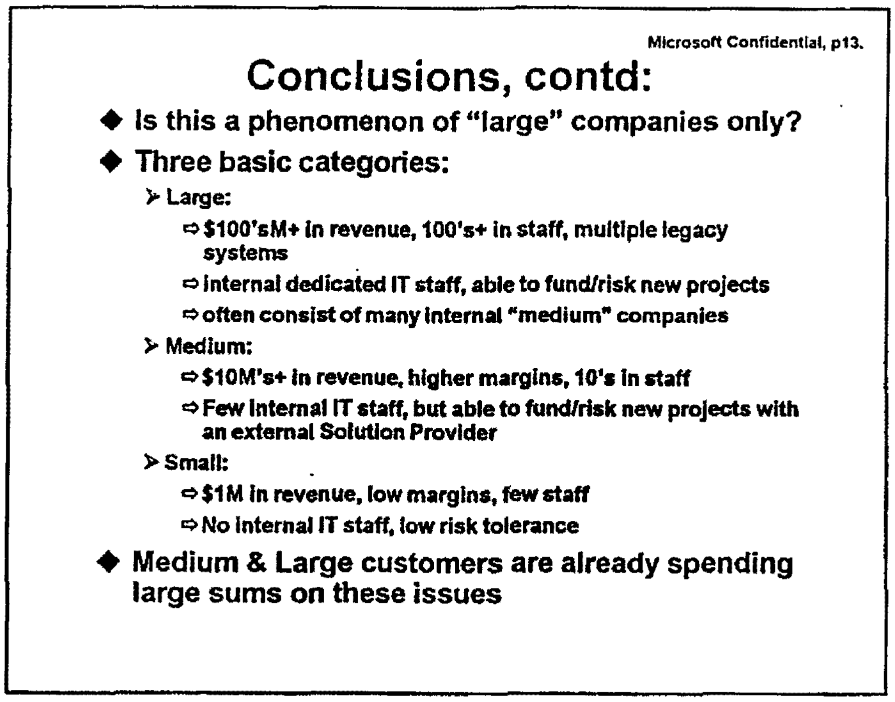
What customers want:



Microsoft Confidential, p12.



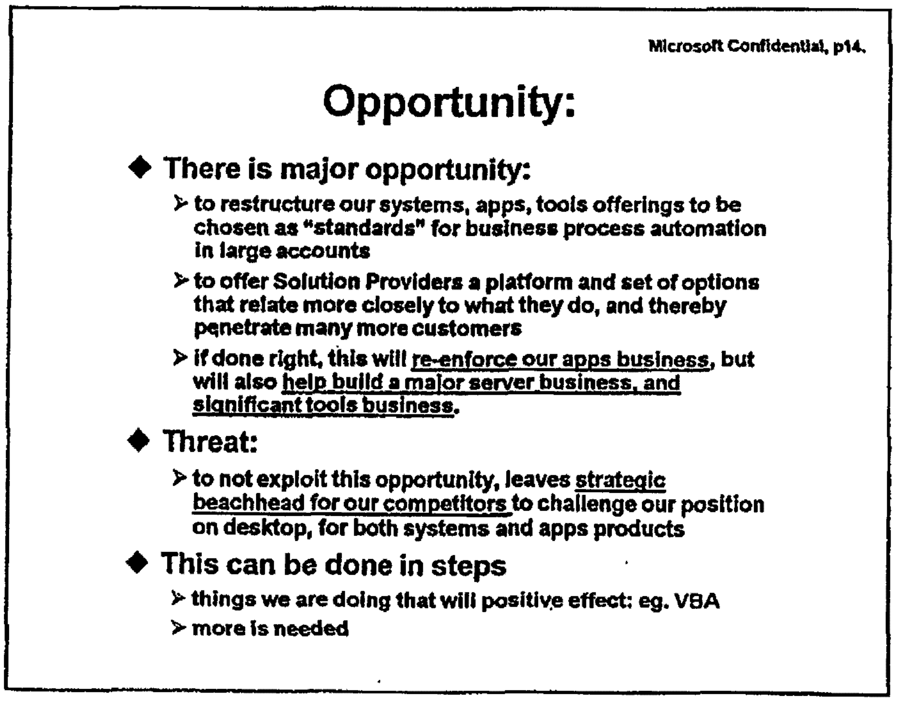
MS7059063
CONFIDENTIAL



Microsoft Confidential, p13.

Conclusions, contd:

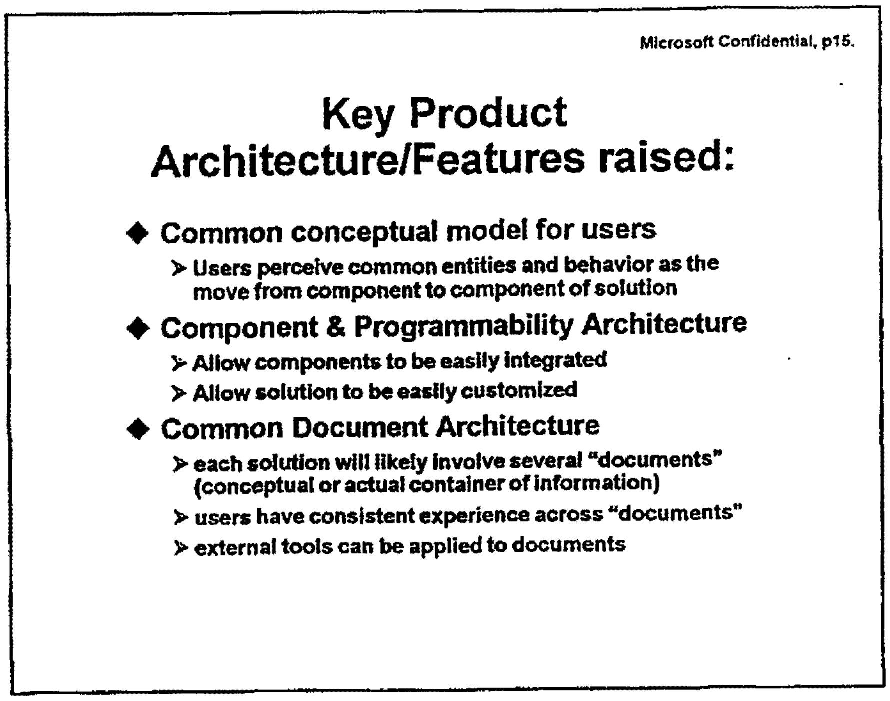
- Is this a phenomenon of "large" companies only?
- Three basic categories:
  - Large:
    - $100'sM+ in revenue, 100's+ in staff, multiple legacy systems
    - internal dedicated IT staff, able to fund/risk new projects
    - often consist of many internal "medium" companies
  - Medium:
    - $10M's+ in revenue, higher margins, 10's in staff
    - Few internal IT staff, but able to fund/risk new projects with an external Solution Provider
  - Small:
    - $1M in revenue, low margins, few staff
    - No internal IT staff, low risk tolerance
- Medium & Large customers are already spending large sums on these issues



Microsoft Confidential, p14.

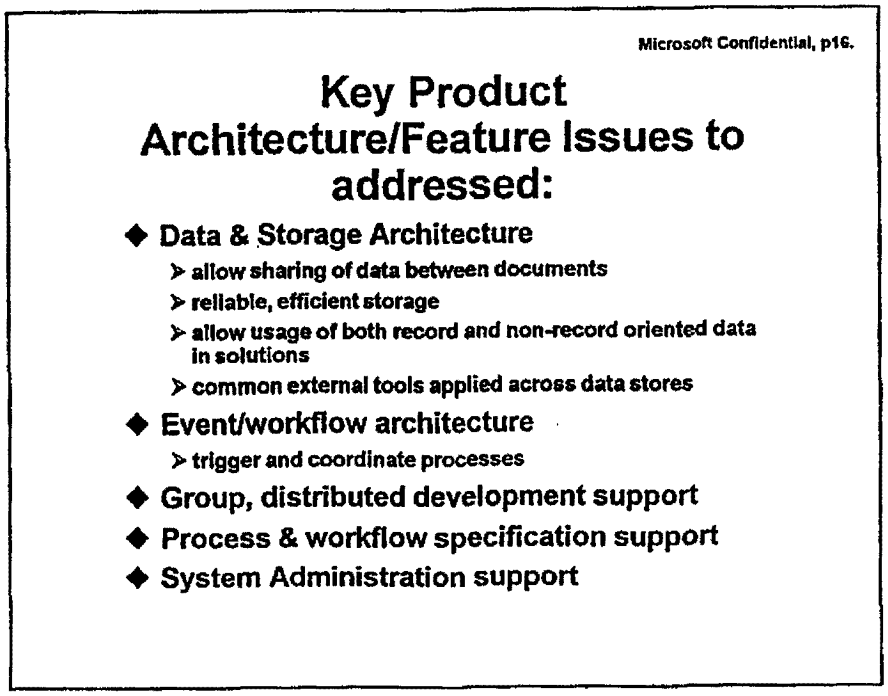
Opportunity:

- There is major opportunity:
  - to restructure our systems, apps, tools offerings to be chosen as "standards" for business process automation in large accounts
  - to offer Solution Providers a platform and set of options that relate more closely to what they do, and thereby penetrate many more customers
  - if done right, this will re-enforce our apps business, but will also help build a major server business, and significant tools business.
- Threat:
  - to not exploit this opportunity, leaves strategic beachhead for our competitors to challenge our position on desktop, for both systems and apps products
- This can be done in steps
  - things we are doing that will positive effect: eg. VBA
  - more is needed



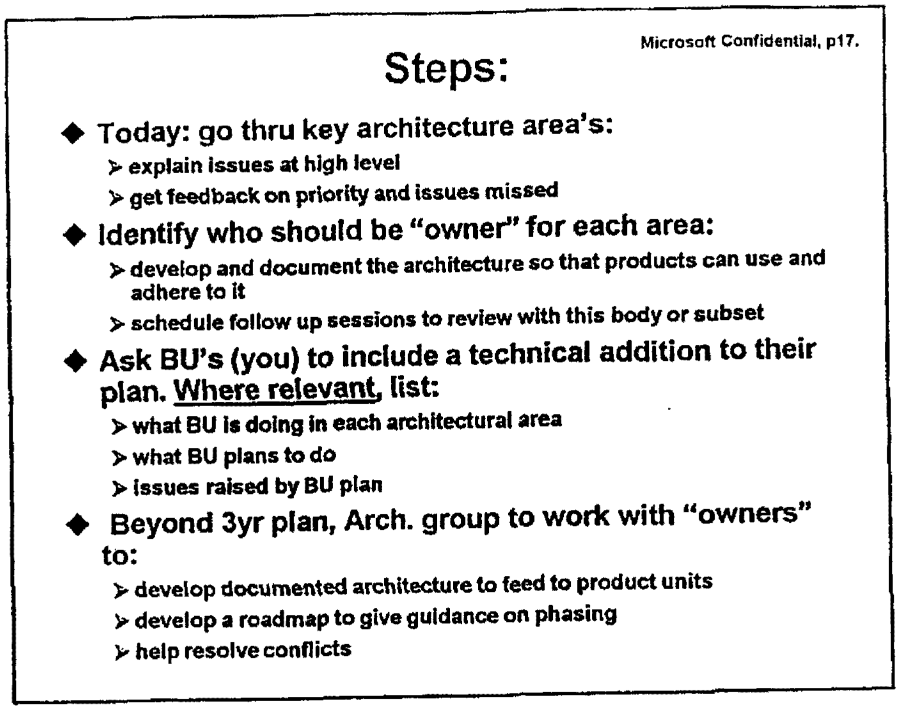
Key Product Architecture/Features raised:

- Common conceptual model for users
  - Users perceive common entities and behavior as the move from component to component of solution
- Component & Programmability Architecture
  - Allow components to be easily integrated
  - Allow solution to be easily customized
- Common Document Architecture
  - each solution will likely involve several "documents" (conceptual or actual container of information)
  - users have consistent experience across "documents"
  - external tools can be applied to documents



Key Product Architecture/Feature Issues to addressed:

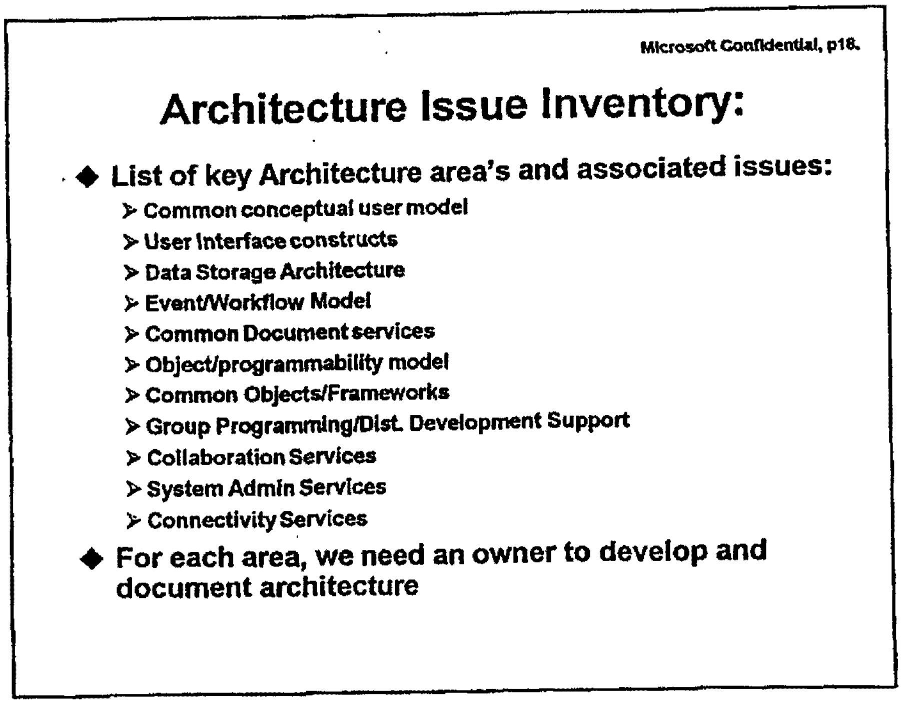
- Data & Storage Architecture
  - allow sharing of data between documents
  - reliable, efficient storage
  - allow usage of both record and non-record oriented data in solutions
  - common external tools applied across data stores
- Event/workflow architecture
  - trigger and coordinate processes
- Group, distributed development support
- Process & workflow specification support
- System Administration support



Microsoft Confidential, p17.

Steps:

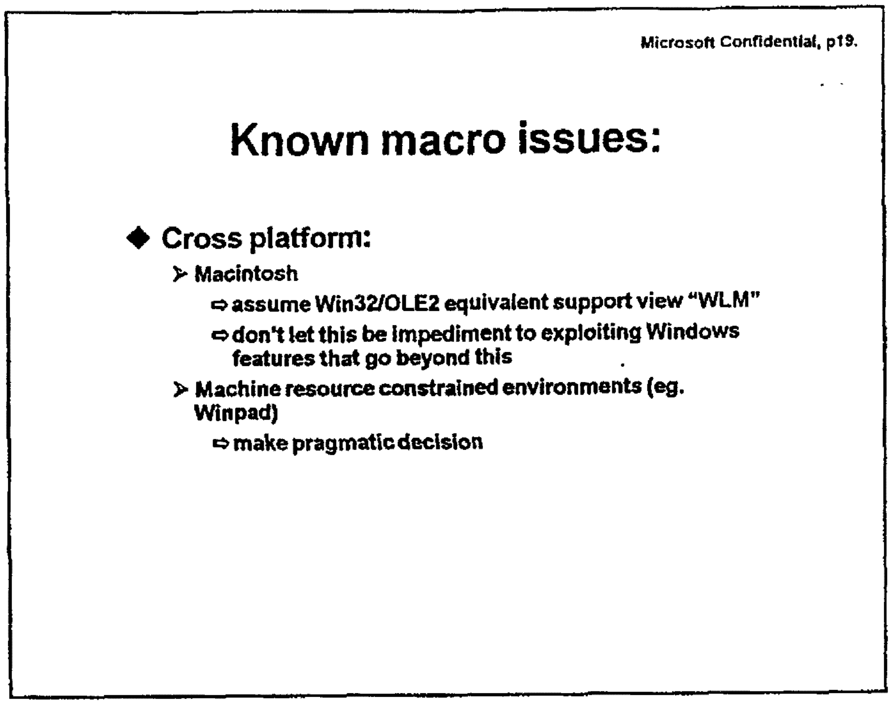
- Today: go thru key architecture area's:
  - explain issues at high level
  - get feedback on priority and issues missed
- Identify who should be "owner" for each area:
  - develop and document the architecture so that products can use and adhere to it
  - schedule follow up sessions to review with this body or subset
- Ask BU's (you) to include a technical addition to their plan. Where relevant, list:
  - what BU is doing in each architectural area
  - what BU plans to do
  - issues raised by BU plan
- Beyond 3yr plan, Arch. group to work with "owners" to:
  - develop documented architecture to feed to product units
  - develop a roadmap to give guidance on phasing
  - help resolve conflicts



Microsoft Confidential, p18.

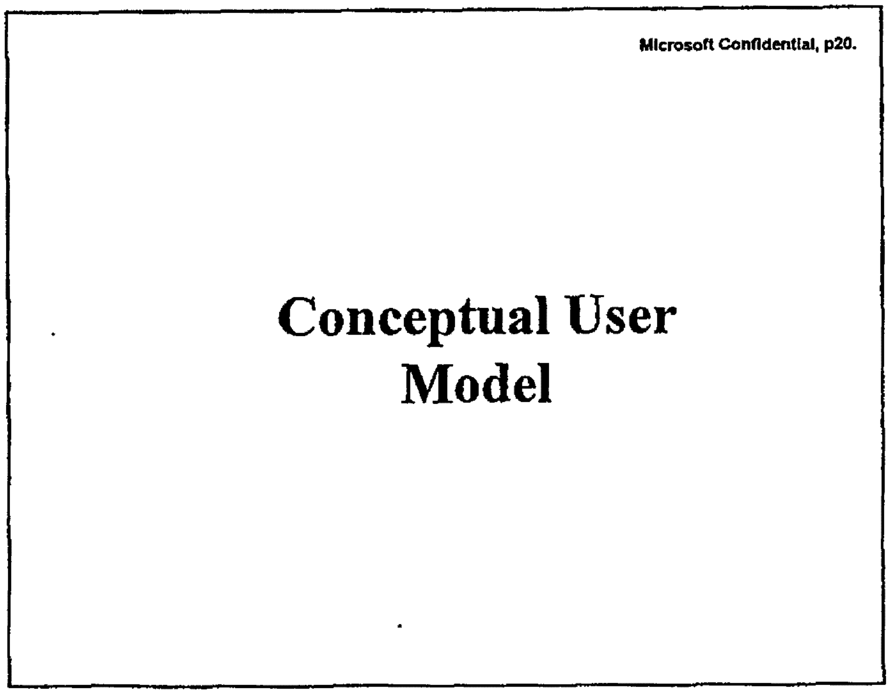
Architecture Issue Inventory:

- List of key Architecture area's and associated issues:
  - Common conceptual user model
  - User Interface constructs
  - Data Storage Architecture
  - Event/Workflow Model
  - Common Document services
  - Object/programmability model
  - Common Objects/Frameworks
  - Group Programming/Dist. Development Support
  - Collaboration Services
  - System Admin Services
  - Connectivity Services
- For each area, we need an owner to develop and document architecture

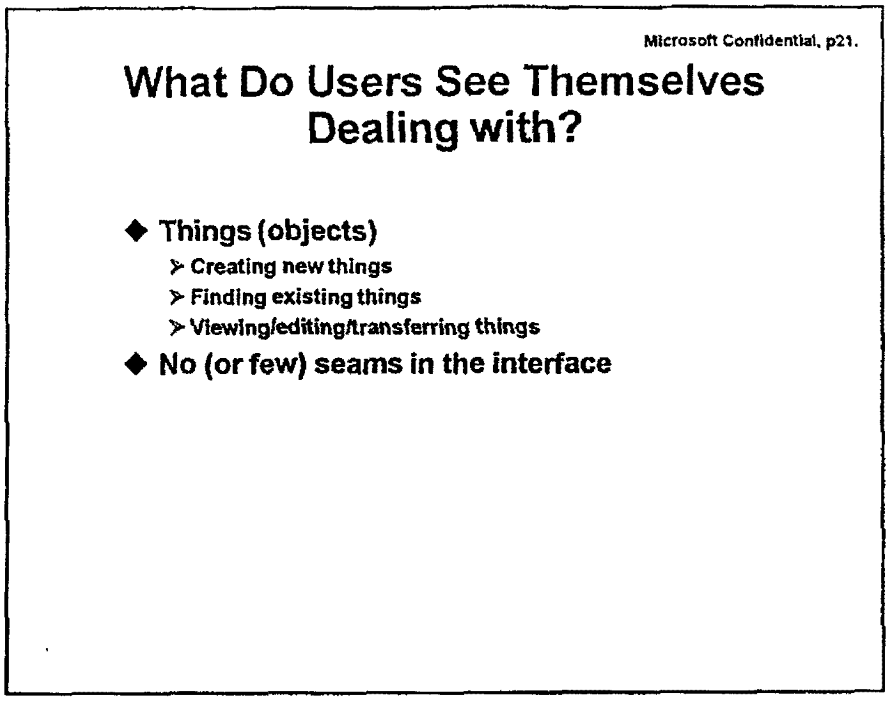


Known macro issues:

- Cross platform:
  - Macintosh
    - ⇨ assume Win32/OLE2 equivalent support view "WLM"
    - ⇨ don't let this be impediment to exploiting Windows features that go beyond this
  - Machine resource constrained environments (eg. Winpad)
    - ⇨ make pragmatic decision

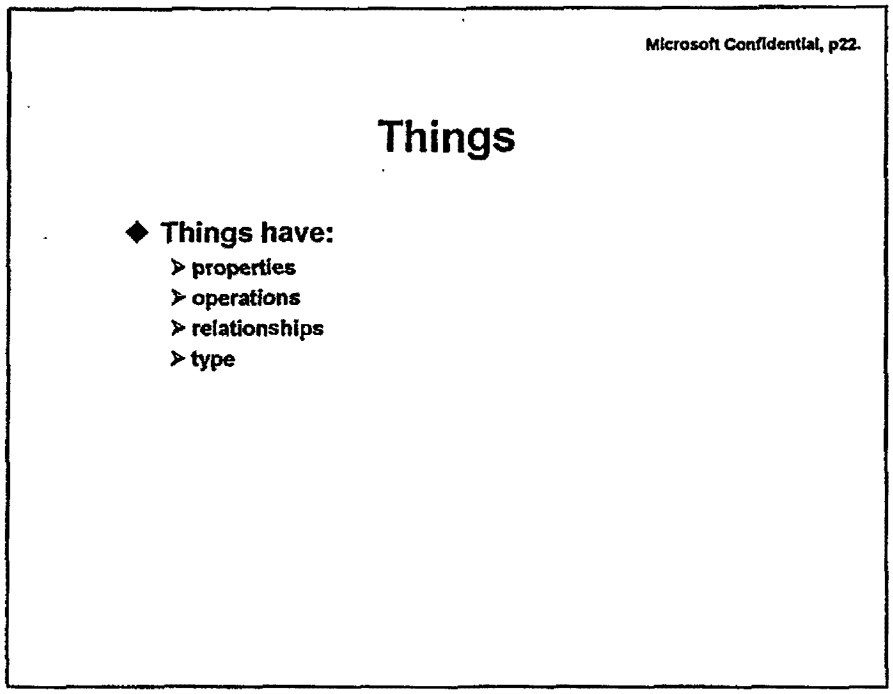


Conceptual User Model



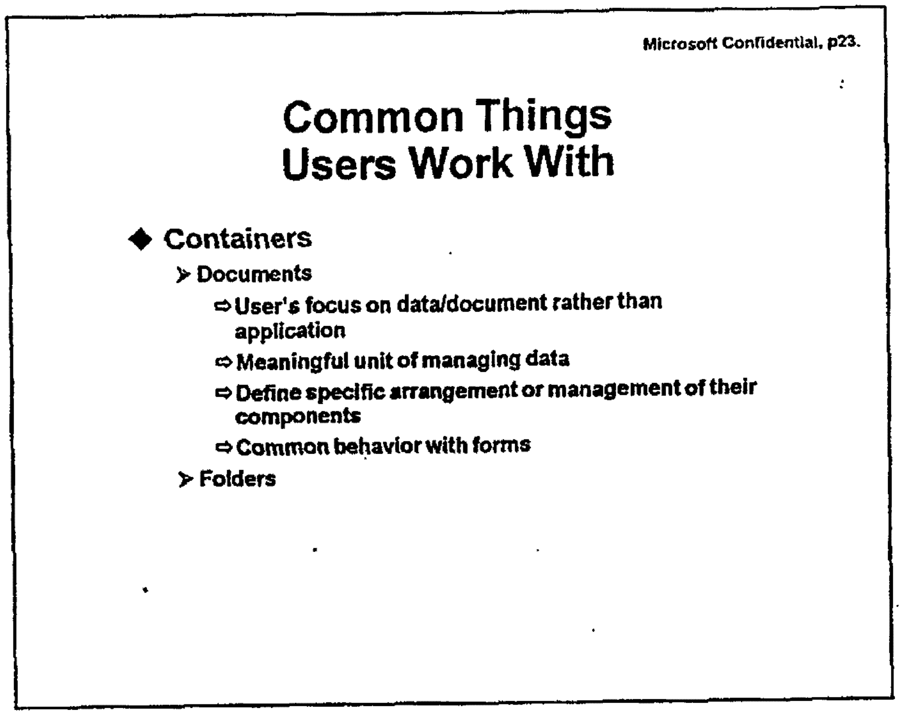
What Do Users See Themselves Dealing with?

- Things (objects)
  - Creating new things
  - Finding existing things
  - Viewing/editing/transferring things
- No (or few) seams in the interface



Things

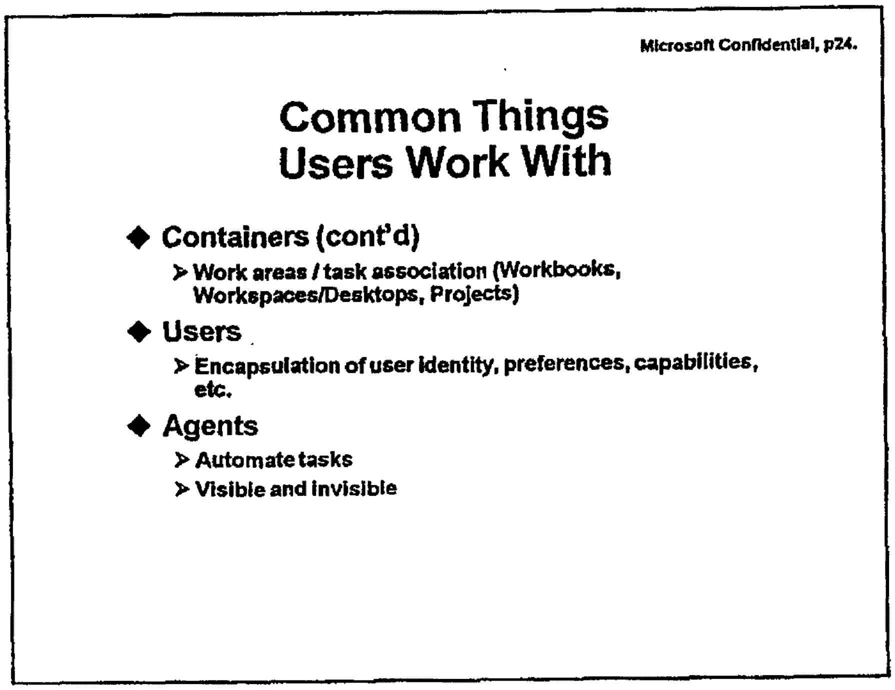
- Things have:
  - properties
  - operations
  - relationships
  - type



Microsoft Confidential, p23.

Common Things Users Work With

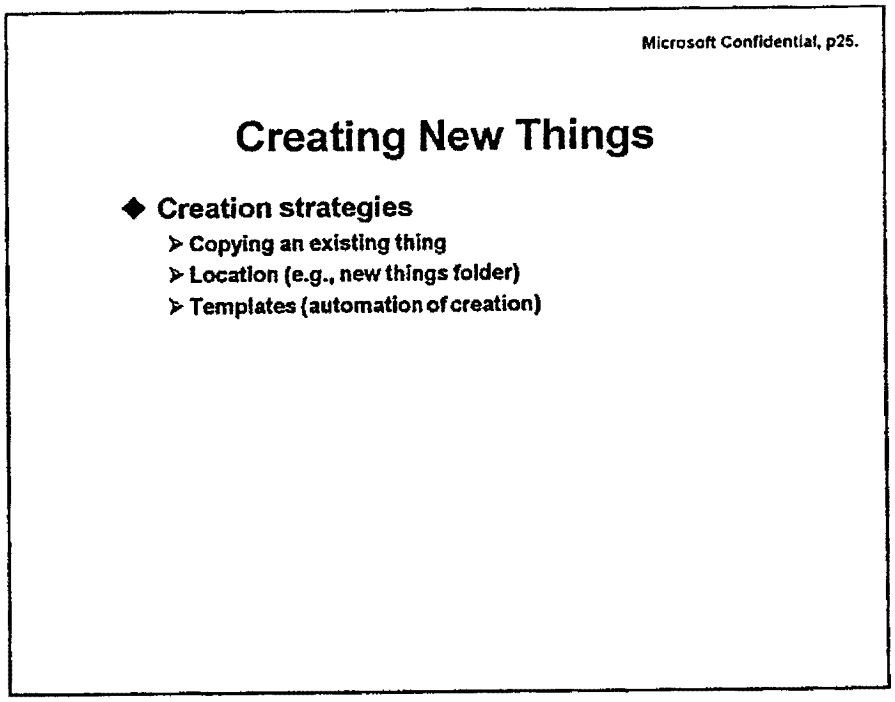
- Containers
  - Documents
    - User's focus on data/document rather than application
    - Meaningful unit of managing data
    - Define specific arrangement or management of their components
    - Common behavior with forms
  - Folders



Microsoft Confidential, p24.

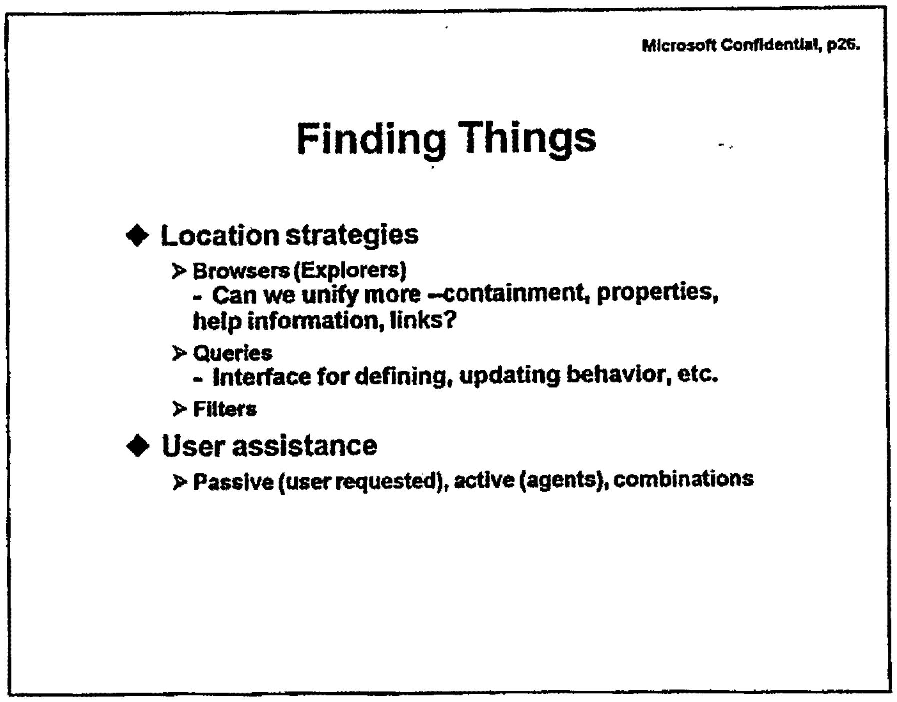
Common Things Users Work With

- Containers (cont'd)
  - Work areas / task association (Workbooks, Workspaces/Desktops, Projects)
- Users
  - Encapsulation of user identity, preferences, capabilities, etc.
- Agents
  - Automate tasks
  - Visible and invisible



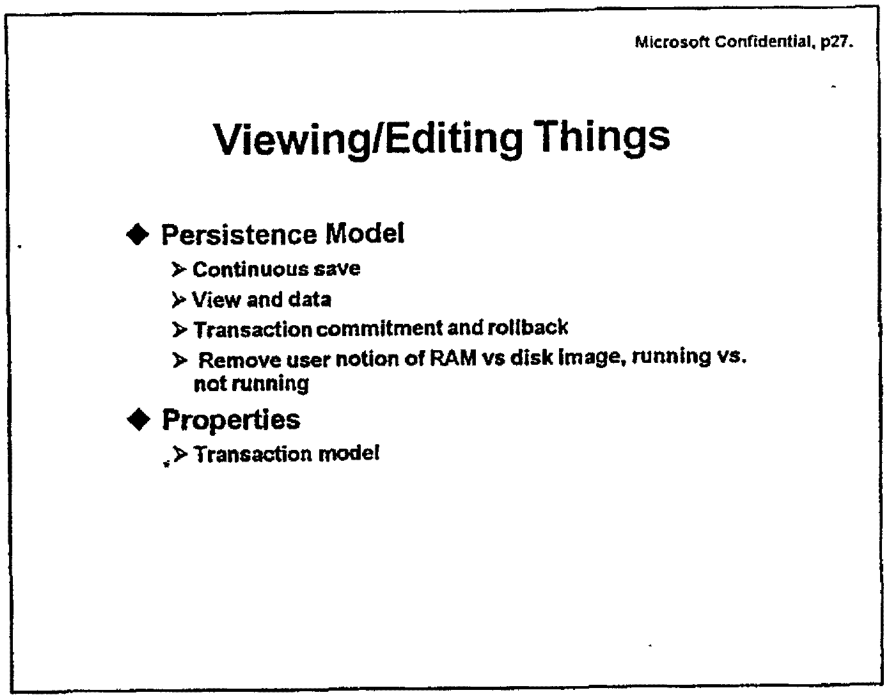
Creating New Things

- ◆ Creation strategies
  - ➢ Copying an existing thing
  - ➢ Location (e.g., new things folder)
  - ➢ Templates (automation of creation)



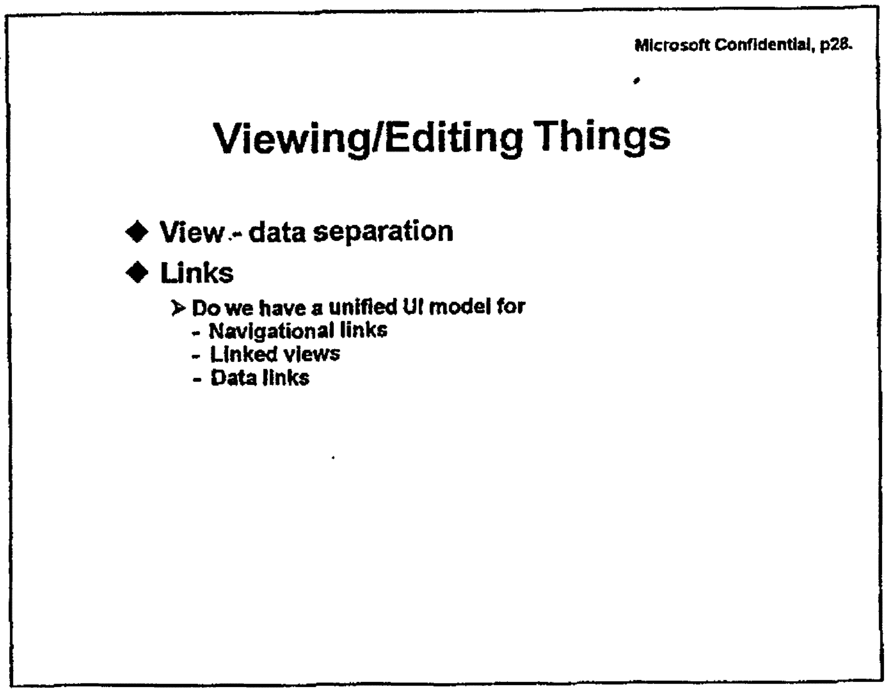
Finding Things

- ◆ Location strategies
  - ➢ Browsers (Explorers)
    - Can we unify more --containment, properties, help information, links?
  - ➢ Queries
    - Interface for defining, updating behavior, etc.
  - ➢ Filters
- ◆ User assistance
  - ➢ Passive (user requested), active (agents), combinations



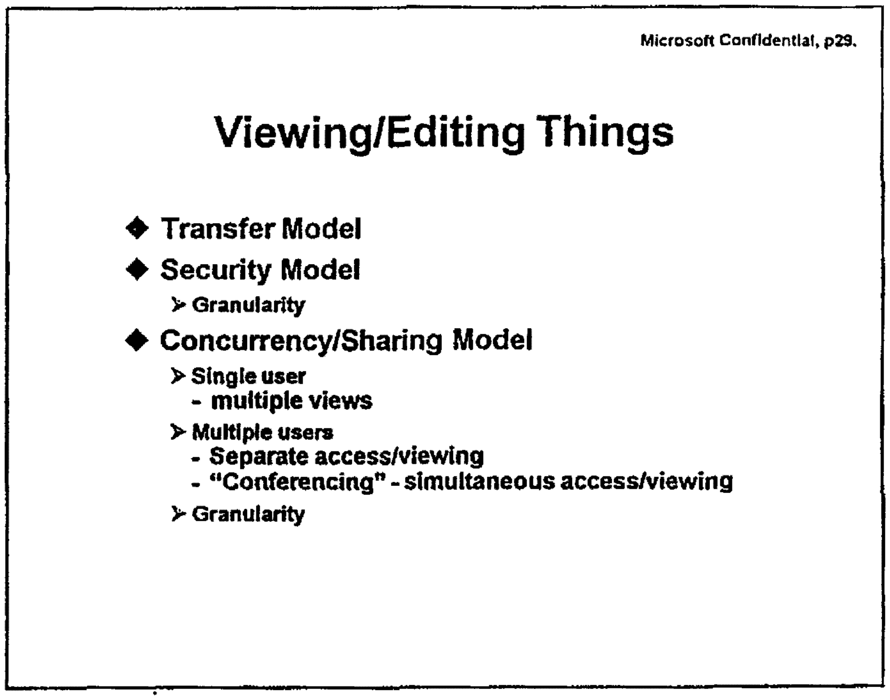
Viewing/Editing Things

- Persistence Model
  - Continuous save
  - View and data
  - Transaction commitment and rollback
  - Remove user notion of RAM vs disk image, running vs. not running
- Properties
  - Transaction model



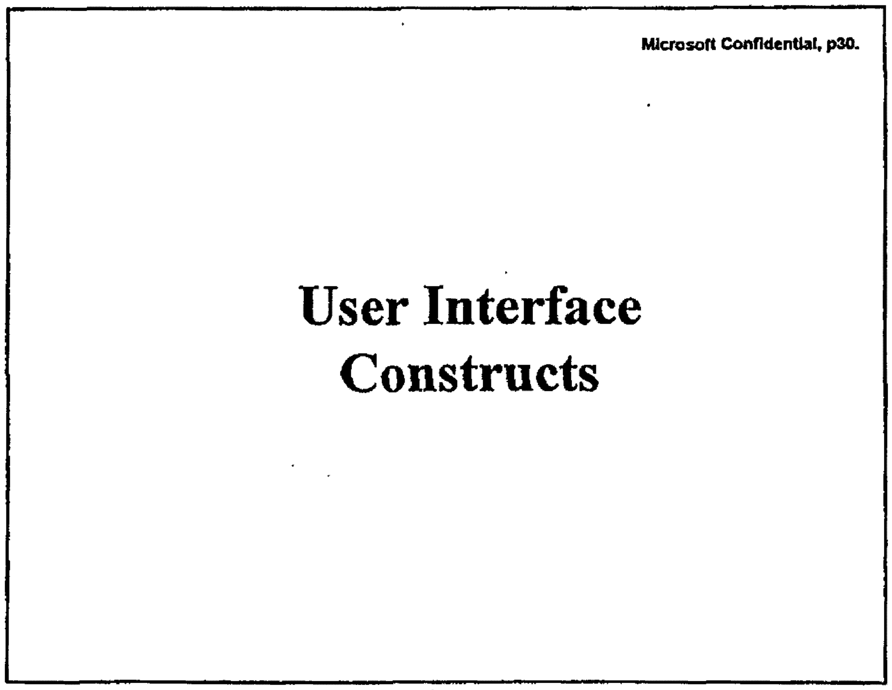
Viewing/Editing Things

- View - data separation
- Links
  - Do we have a unified UI model for
    - Navigational links
    - Linked views
    - Data links

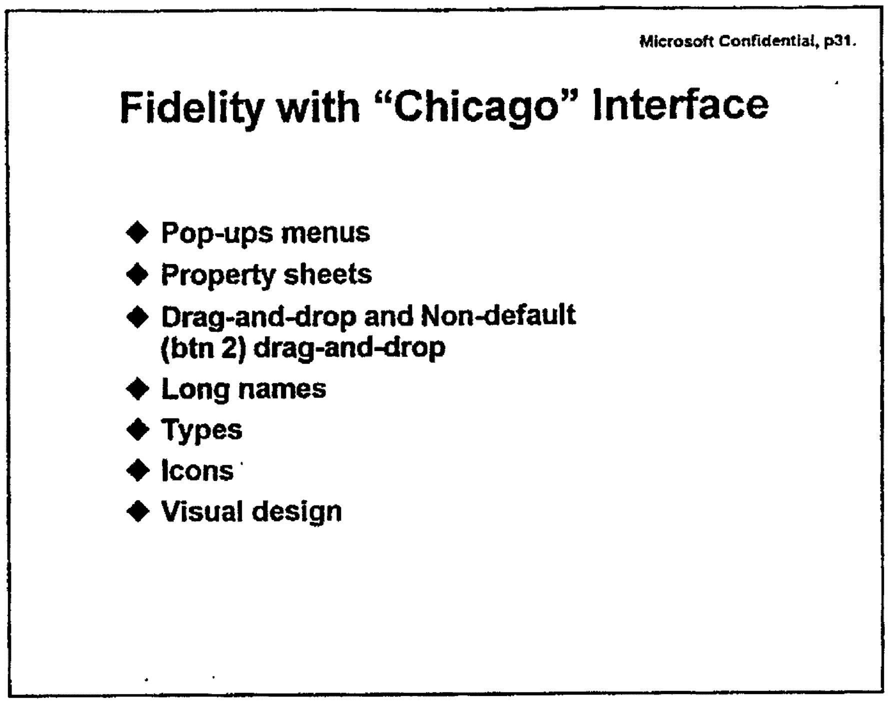


Viewing/Editing Things

- ◆ Transfer Model
- ◆ Security Model
  - ➢ Granularity
- ◆ Concurrency/Sharing Model
  - ➢ Single user
    - - multiple views
  - ➢ Multiple users
    - - Separate access/viewing
    - - "Conferencing" - simultaneous access/viewing
  - ➢ Granularity

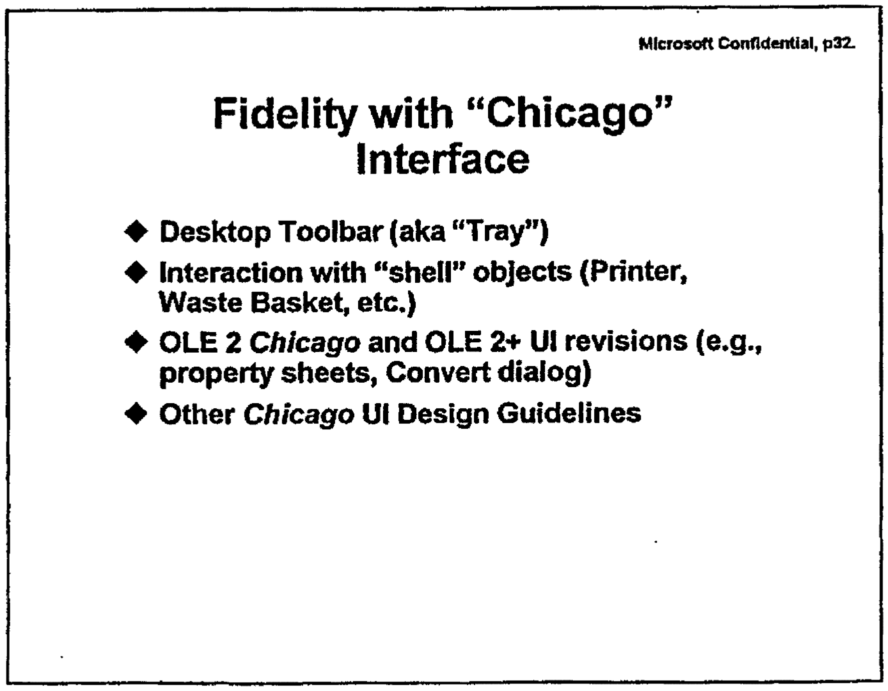


User Interface Constructs



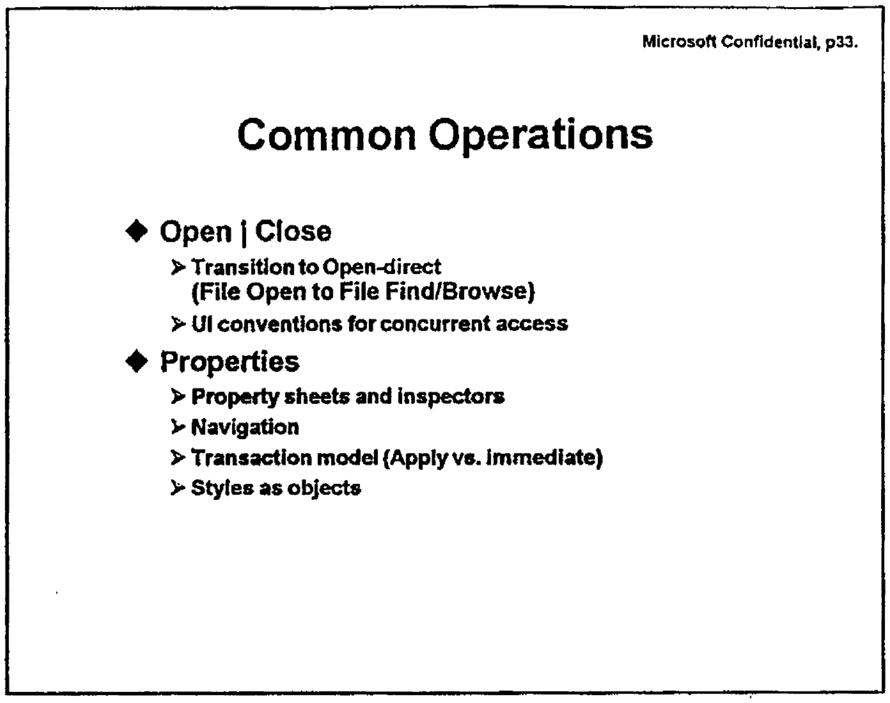
Fidelity with "Chicago" Interface

- Pop-ups menus
- Property sheets
- Drag-and-drop and Non-default (btn 2) drag-and-drop
- Long names
- Types
- Icons
- Visual design



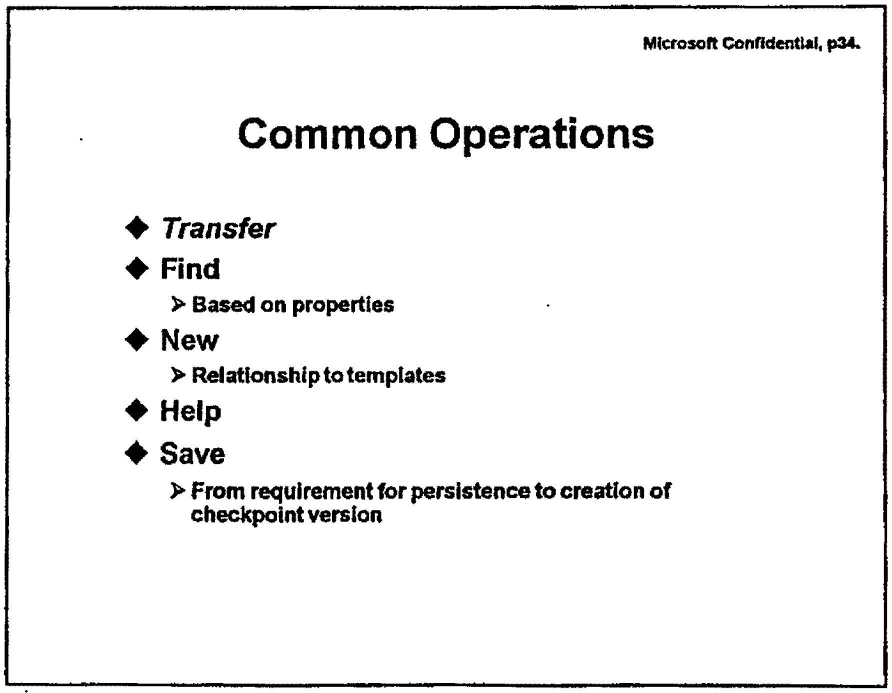
Fidelity with "Chicago" Interface

- Desktop Toolbar (aka "Tray")
- Interaction with "shell" objects (Printer, Waste Basket, etc.)
- OLE 2 *Chicago* and OLE 2+ UI revisions (e.g., property sheets, Convert dialog)
- Other *Chicago* UI Design Guidelines



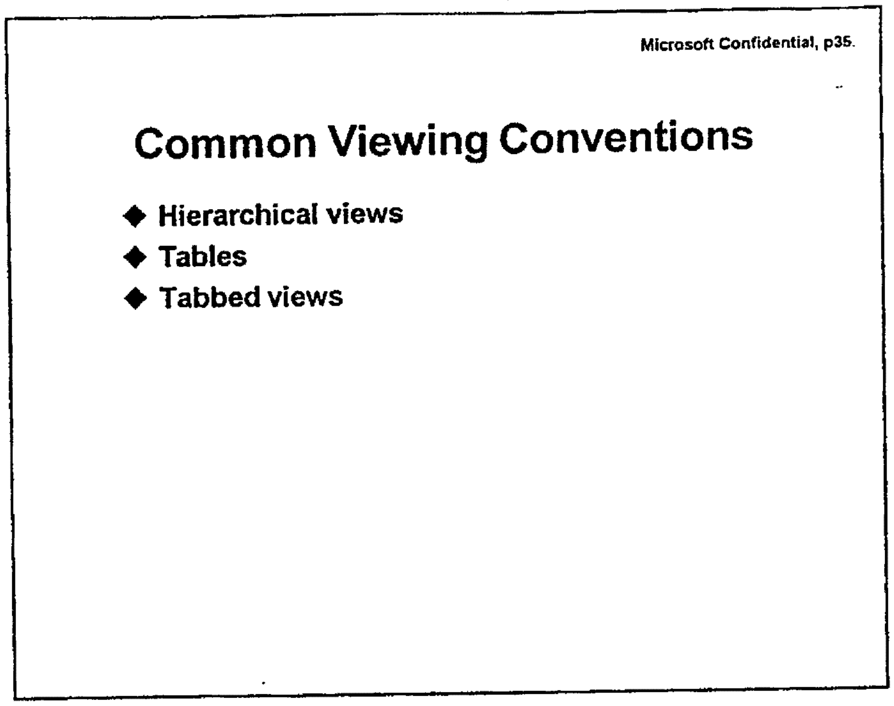
Common Operations

- Open | Close
  - Transition to Open-direct (File Open to File Find/Browse)
  - UI conventions for concurrent access
- Properties
  - Property sheets and inspectors
  - Navigation
  - Transaction model (Apply vs. immediate)
  - Styles as objects



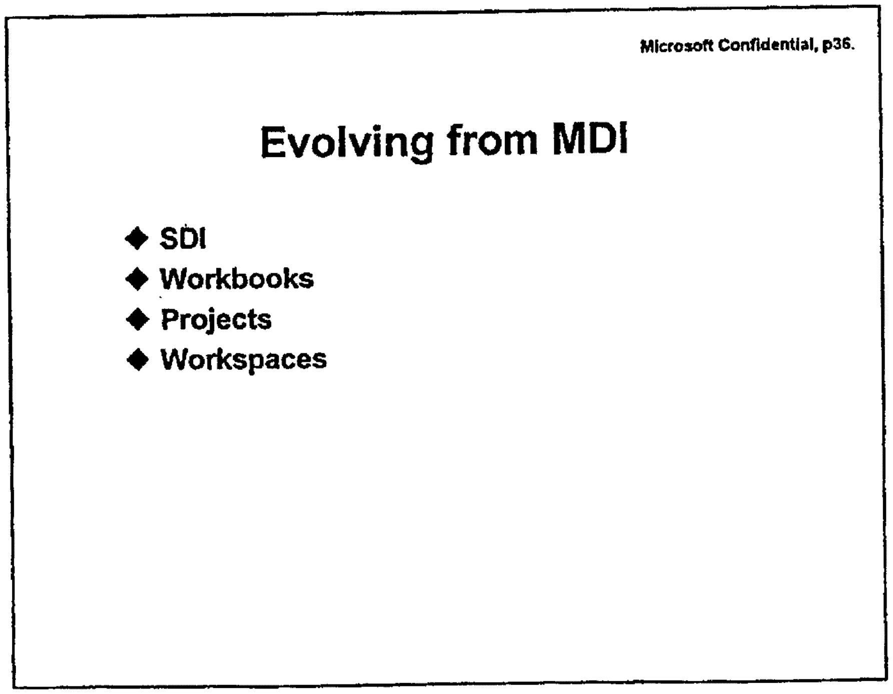
Common Operations

- *Transfer*
- Find
  - Based on properties
- New
  - Relationship to templates
- Help
- Save
  - From requirement for persistence to creation of checkpoint version



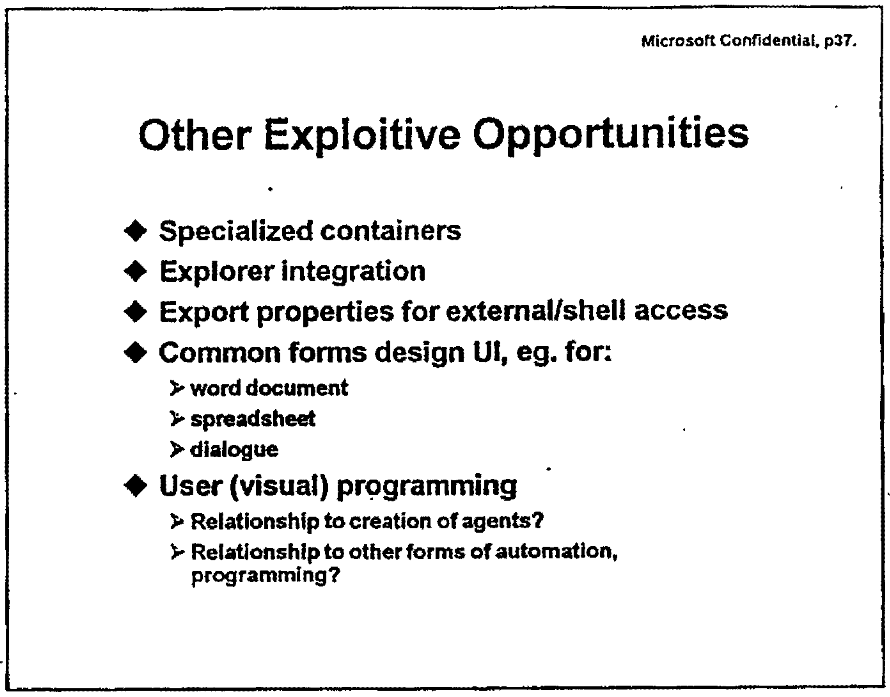
Common Viewing Conventions

- Hierarchical views
- Tables
- Tabbed views



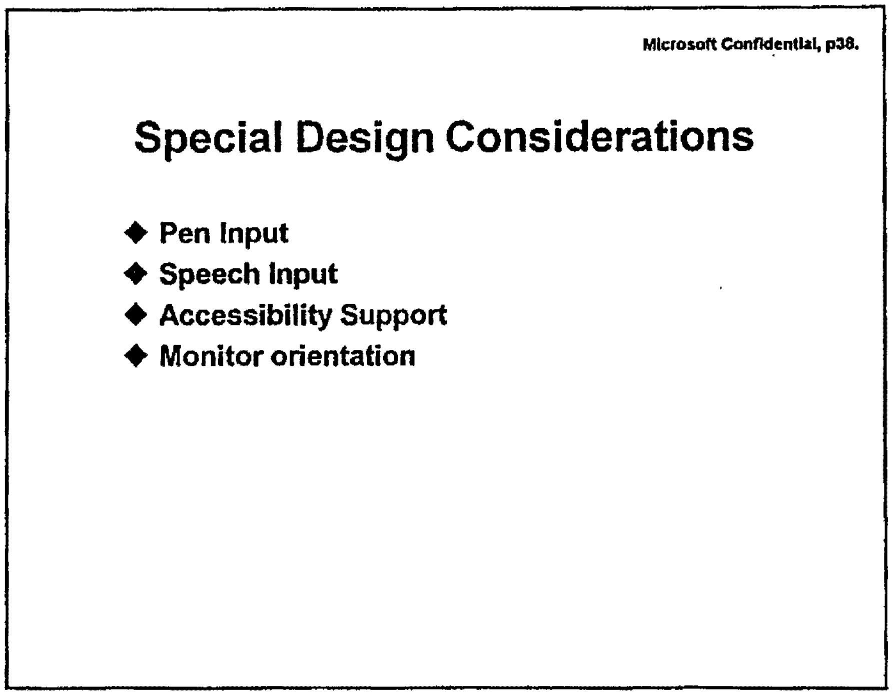
Evolving from MDI

- SDI
- Workbooks
- Projects
- Workspaces



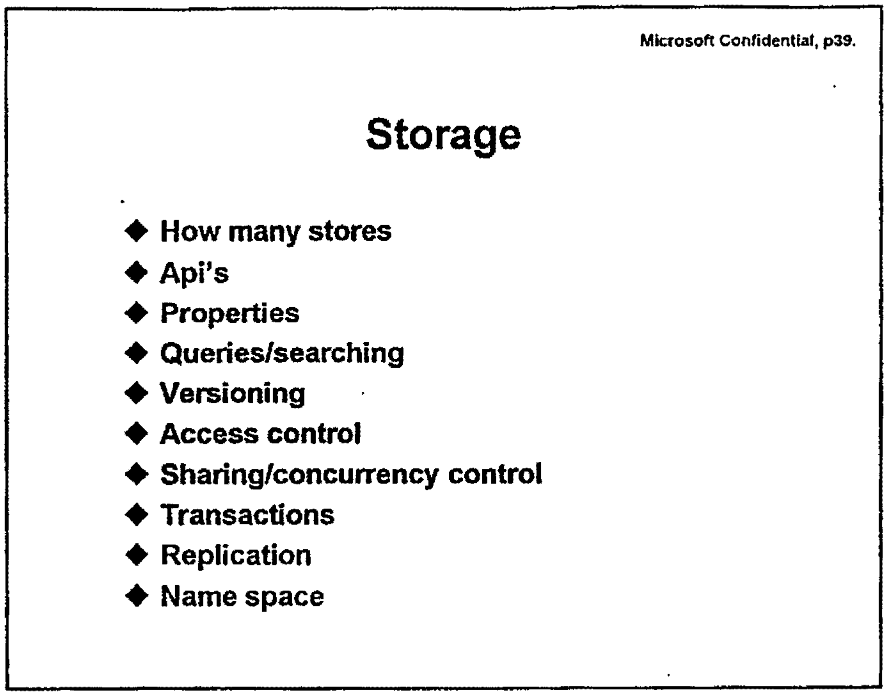
Other Exploitive Opportunities

- Specialized containers
- Explorer integration
- Export properties for external/shell access
- Common forms design UI, eg. for:
  - word document
  - spreadsheet
  - dialogue
- User (visual) programming
  - Relationship to creation of agents?
  - Relationship to other forms of automation, programming?



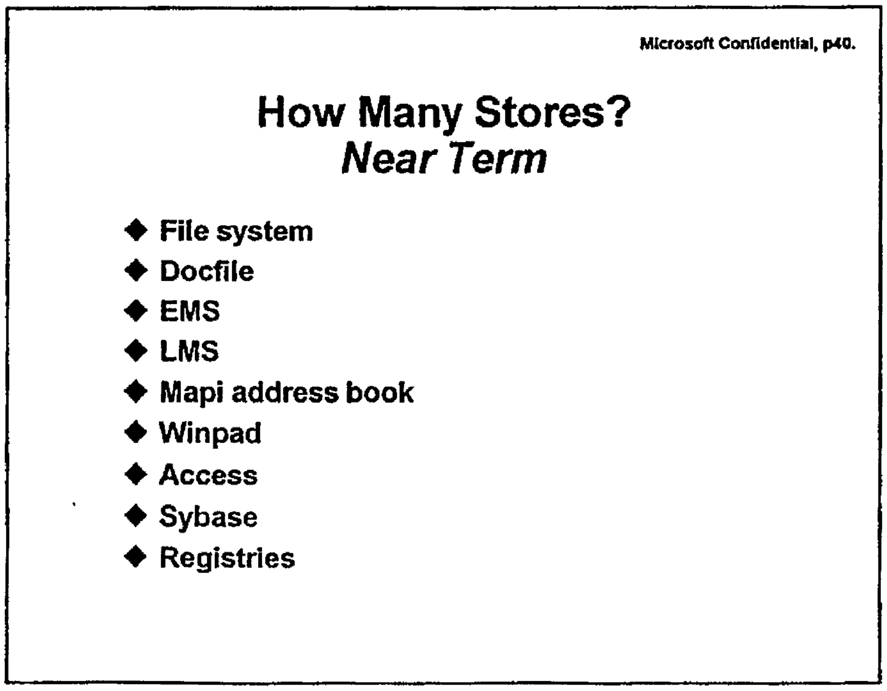
Special Design Considerations

- Pen Input
- Speech Input
- Accessibility Support
- Monitor orientation



Storage

- How many stores
- Api's
- Properties
- Queries/searching
- Versioning
- Access control
- Sharing/concurrency control
- Transactions
- Replication
- Name space



How Many Stores?
Near Term

- File system
- Docfile
- EMS
- LMS
- Mapi address book
- Winpad
- Access
- Sybase
- Registries

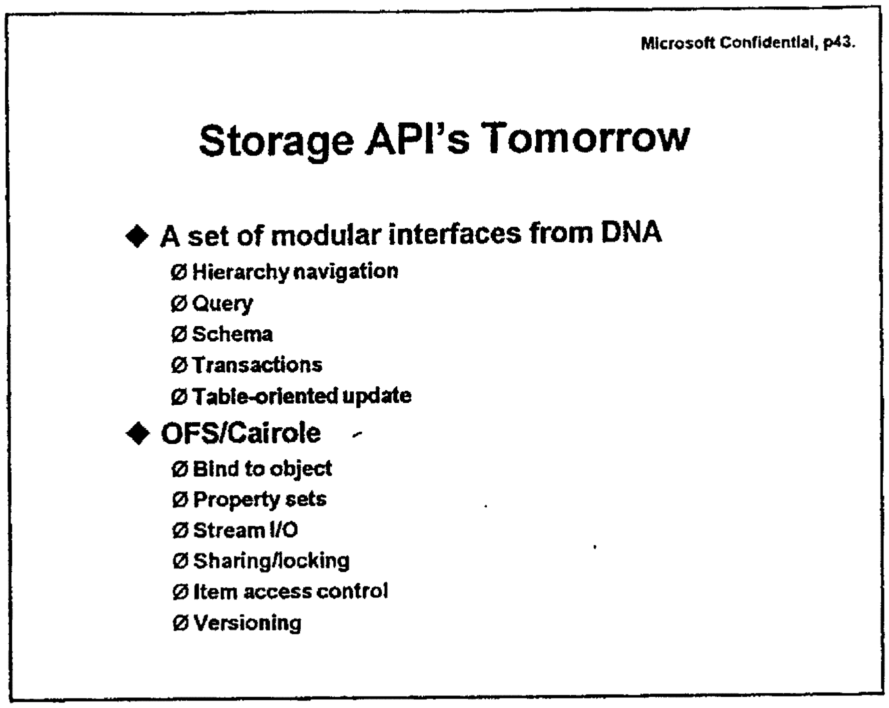
How Many Stores?
Long Term

- OFS
- Lightweight record store
- Server database engine

Storage API's Today

- Record-oriented stores
  - Ø ODBC
  - Ø Jet
- Nonrecord-oriented stores
  - Ø File api
  - Ø Docfile
  - Ø MAPI
  - Ø ODBC

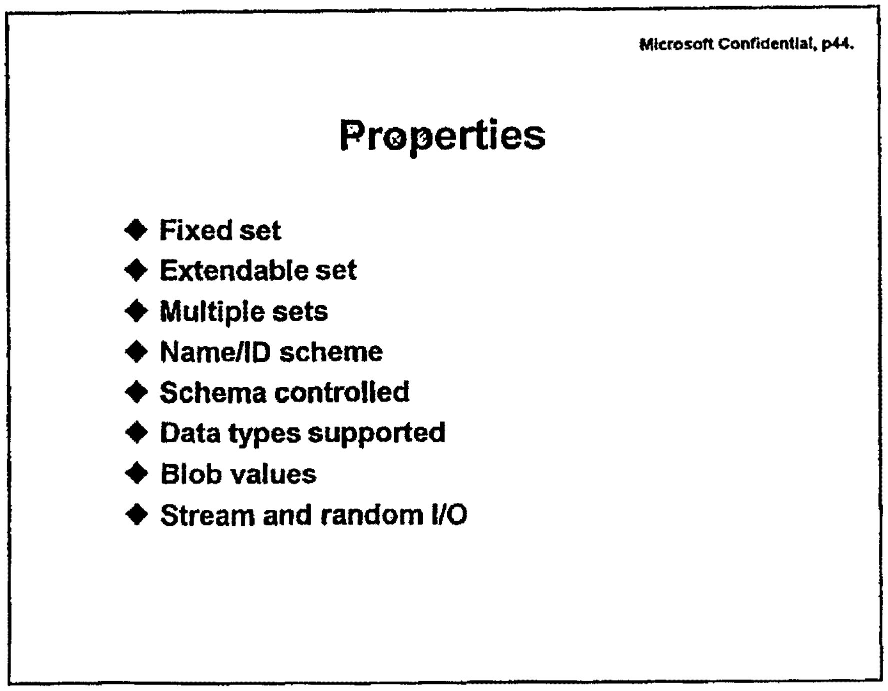
\ WinPad



Microsoft Confidential, p43.

Storage API's Tomorrow

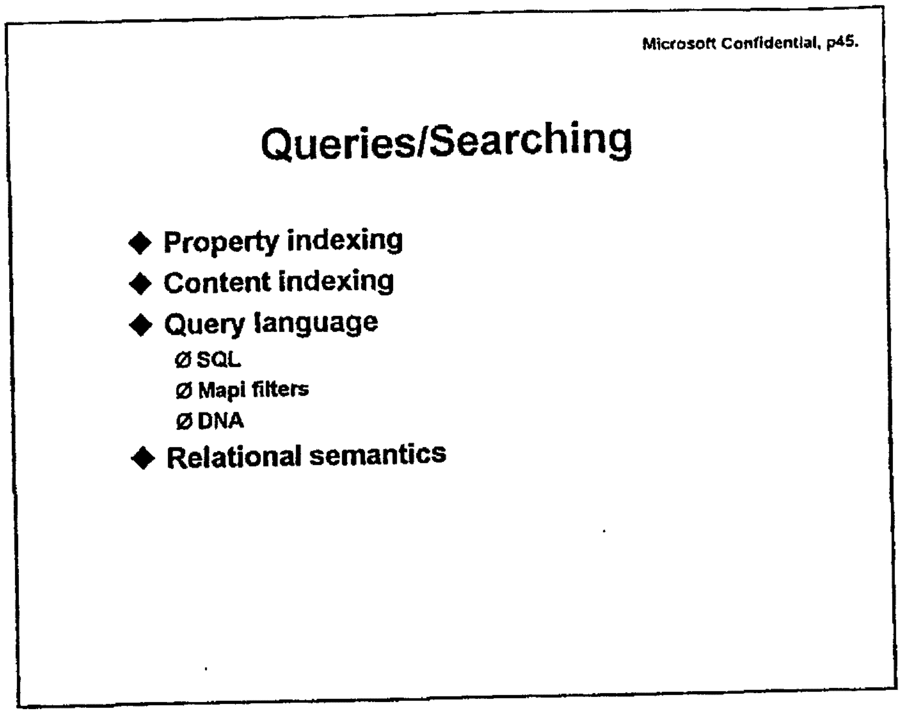
- A set of modular interfaces from DNA
  - Ø Hierarchy navigation
  - Ø Query
  - Ø Schema
  - Ø Transactions
  - Ø Table-oriented update
- OFS/Cairole
  - Ø Bind to object
  - Ø Property sets
  - Ø Stream I/O
  - Ø Sharing/locking
  - Ø Item access control
  - Ø Versioning



Microsoft Confidential, p44.

Properties

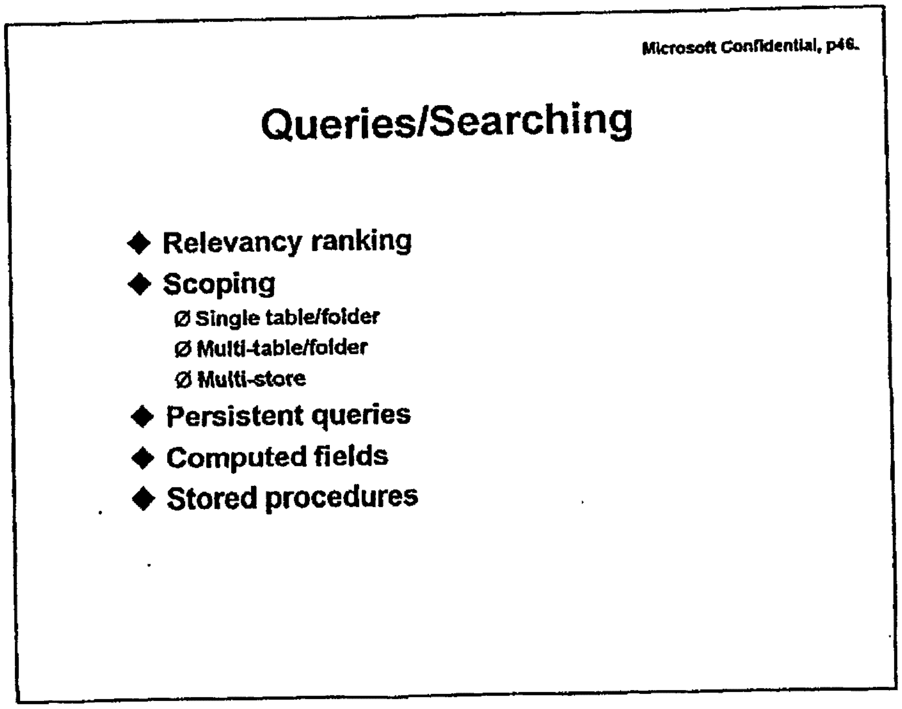
- Fixed set
- Extendable set
- Multiple sets
- Name/ID scheme
- Schema controlled
- Data types supported
- Blob values
- Stream and random I/O



Microsoft Confidential, p45.

Queries/Searching

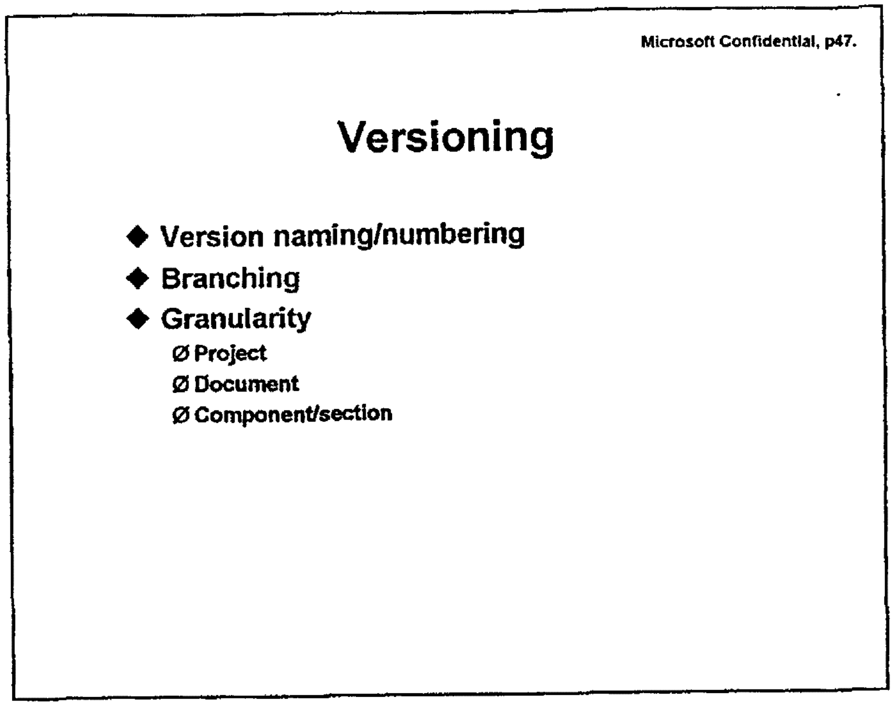
- ◆ Property indexing
- ◆ Content indexing
- ◆ Query language
  - Ø SQL
  - Ø Mapi filters
  - Ø DNA
- ◆ Relational semantics



Microsoft Confidential, p46.

Queries/Searching

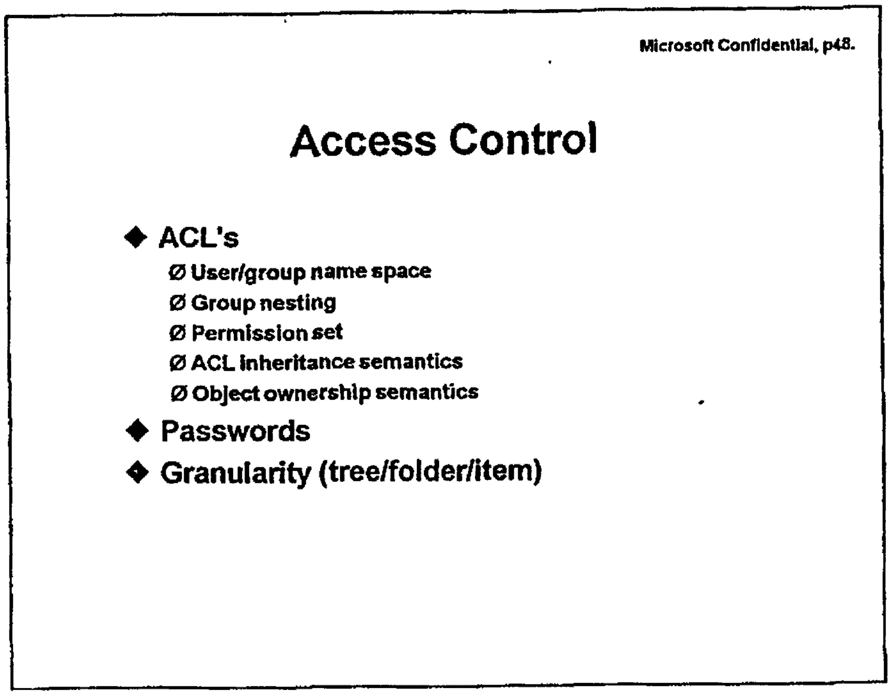
- ◆ Relevancy ranking
- ◆ Scoping
  - Ø Single table/folder
  - Ø Multi-table/folder
  - Ø Multi-store
- ◆ Persistent queries
- ◆ Computed fields
- ◆ Stored procedures



Microsoft Confidential, p47.

Versioning

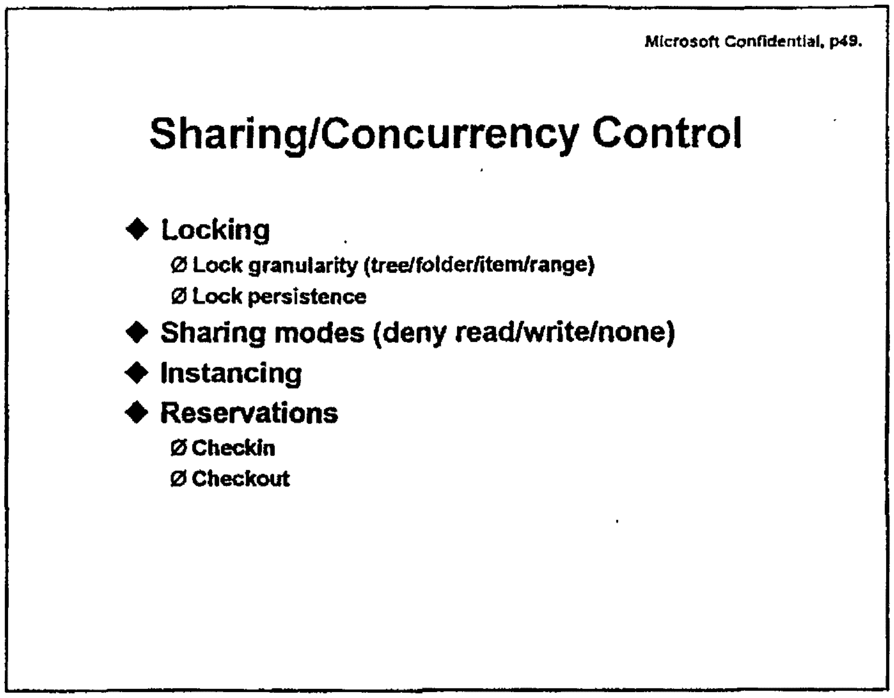
- ◆ Version naming/numbering
- ◆ Branching
- ◆ Granularity
  - Ø Project
  - Ø Document
  - Ø Component/section



Microsoft Confidential, p48.

Access Control

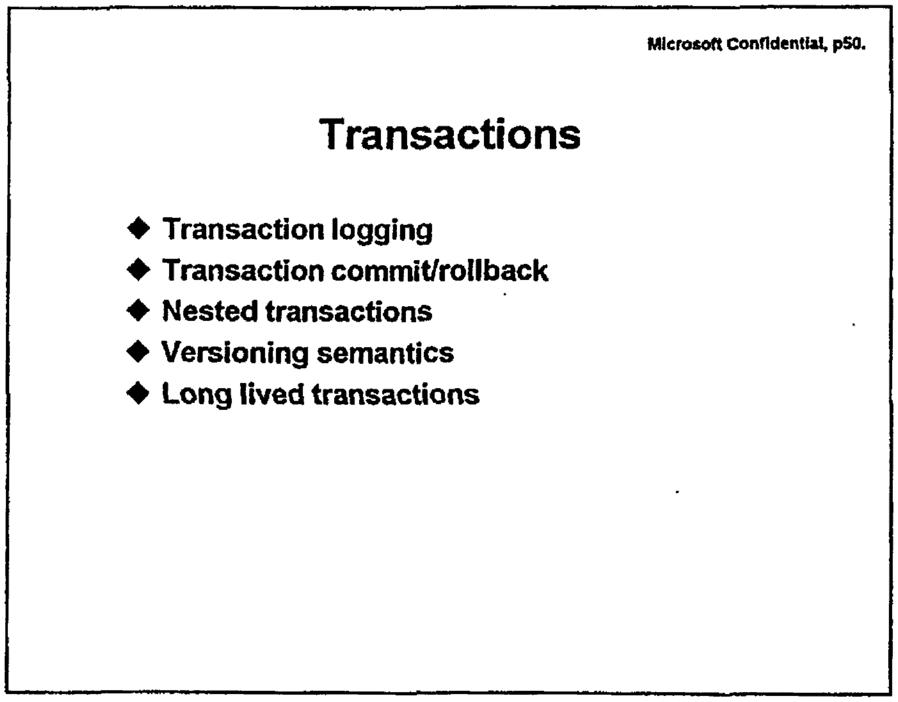
- ◆ ACL's
  - Ø User/group name space
  - Ø Group nesting
  - Ø Permission set
  - Ø ACL inheritance semantics
  - Ø Object ownership semantics
- ◆ Passwords
- ◆ Granularity (tree/folder/item)



Microsoft Confidential, p49.

Sharing/Concurrency Control

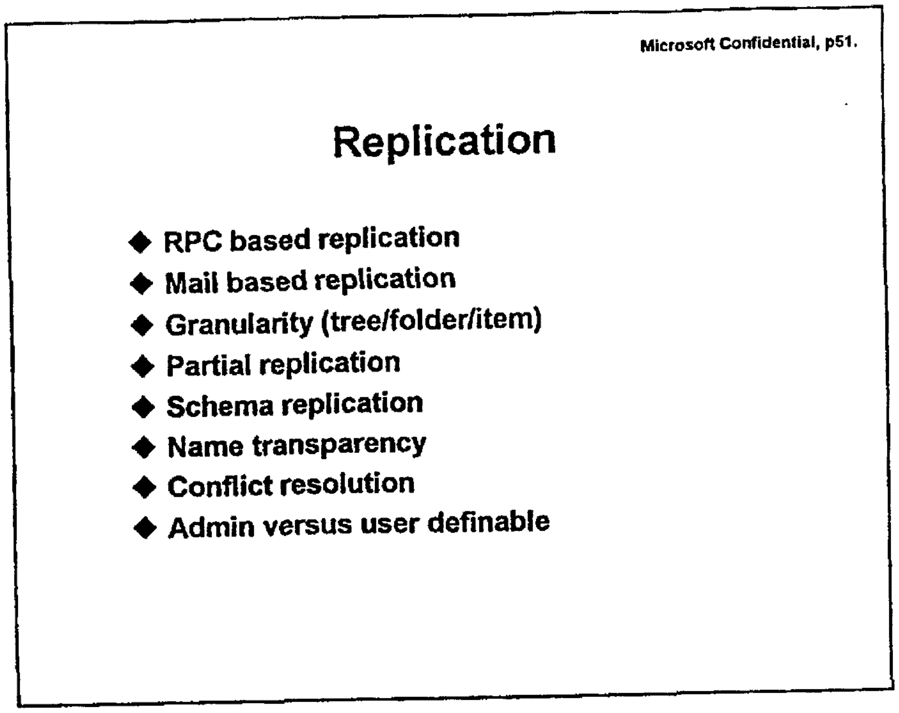
- Locking
  - Ø Lock granularity (tree/folder/item/range)
  - Ø Lock persistence
- Sharing modes (deny read/write/none)
- Instancing
- Reservations
  - Ø Checkin
  - Ø Checkout



Microsoft Confidential, p50.

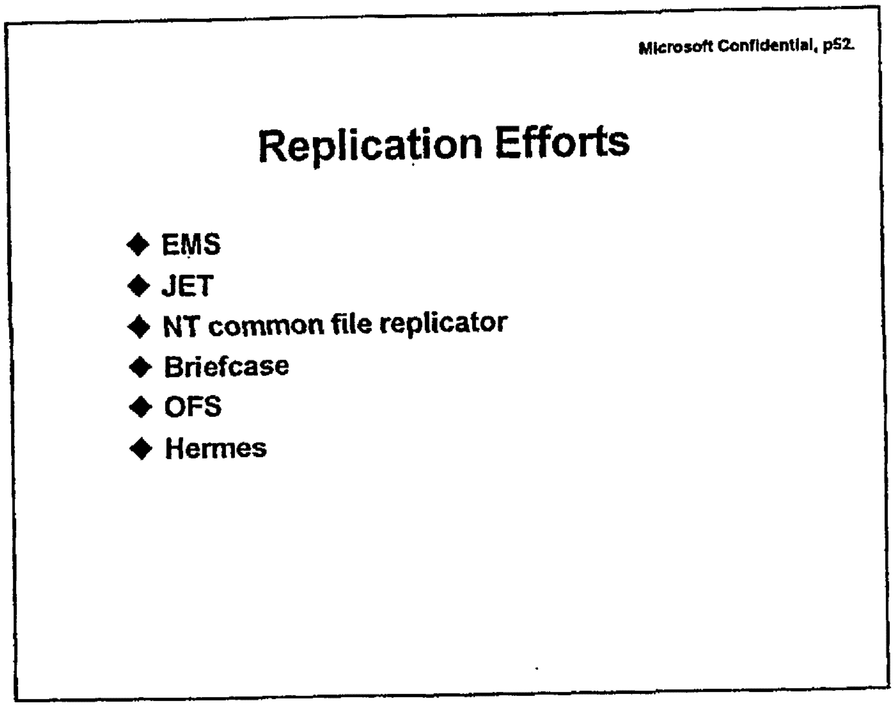
Transactions

- Transaction logging
- Transaction commit/rollback
- Nested transactions
- Versioning semantics
- Long lived transactions



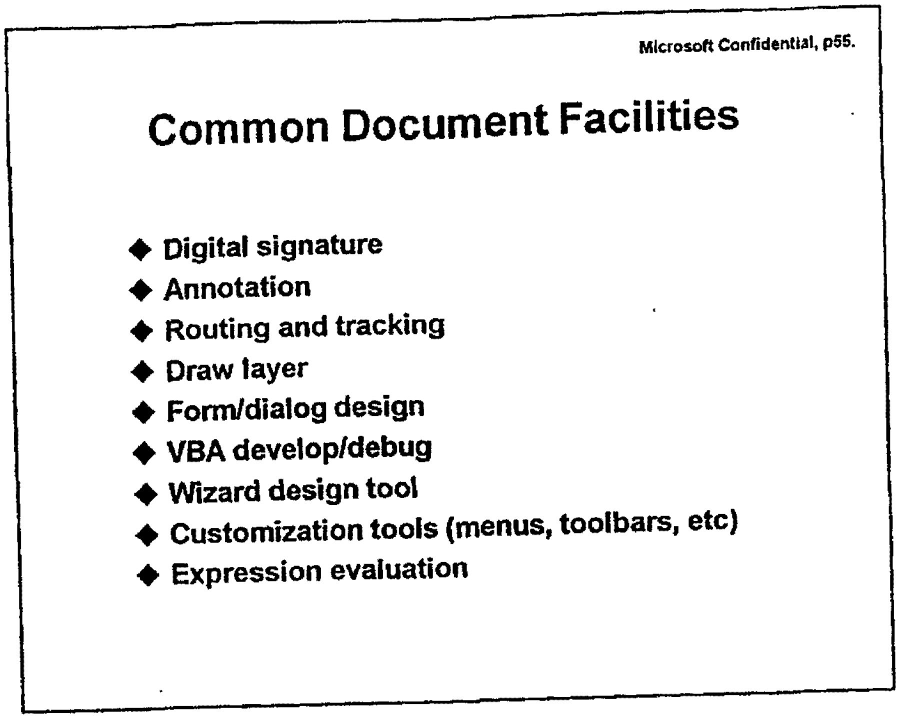
Replication

- ◆ RPC based replication
- ◆ Mail based replication
- ◆ Granularity (tree/folder/item)
- ◆ Partial replication
- ◆ Schema replication
- ◆ Name transparency
- ◆ Conflict resolution
- ◆ Admin versus user definable



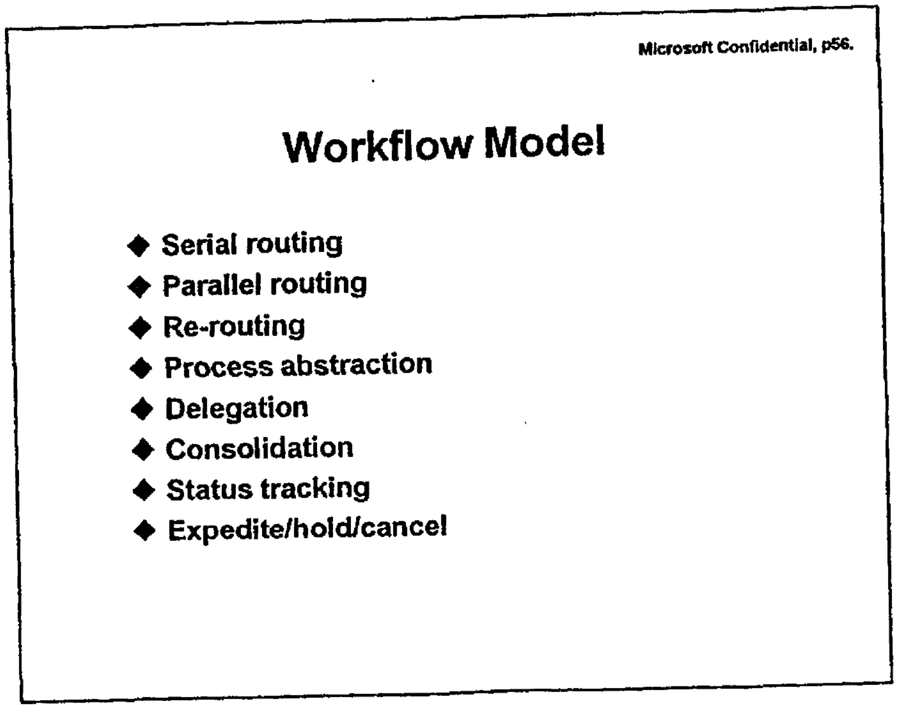
Replication Efforts

- ◆ EMS
- ◆ JET
- ◆ NT common file replicator
- ◆ Briefcase
- ◆ OFS
- ◆ Hermes



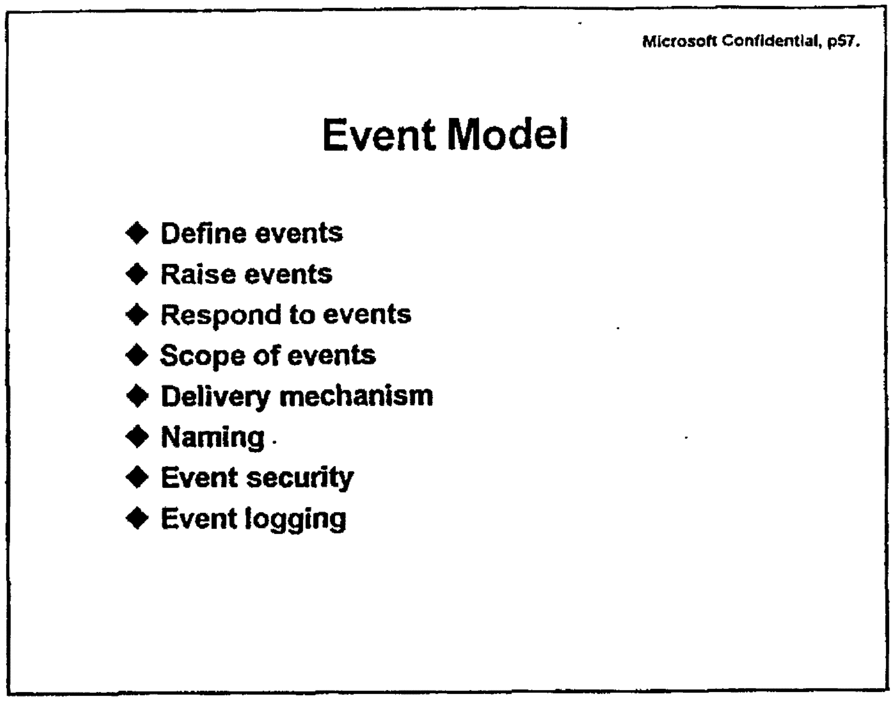
Common Document Facilities

- Digital signature
- Annotation
- Routing and tracking
- Draw layer
- Form/dialog design
- VBA develop/debug
- Wizard design tool
- Customization tools (menus, toolbars, etc)
- Expression evaluation



Workflow Model

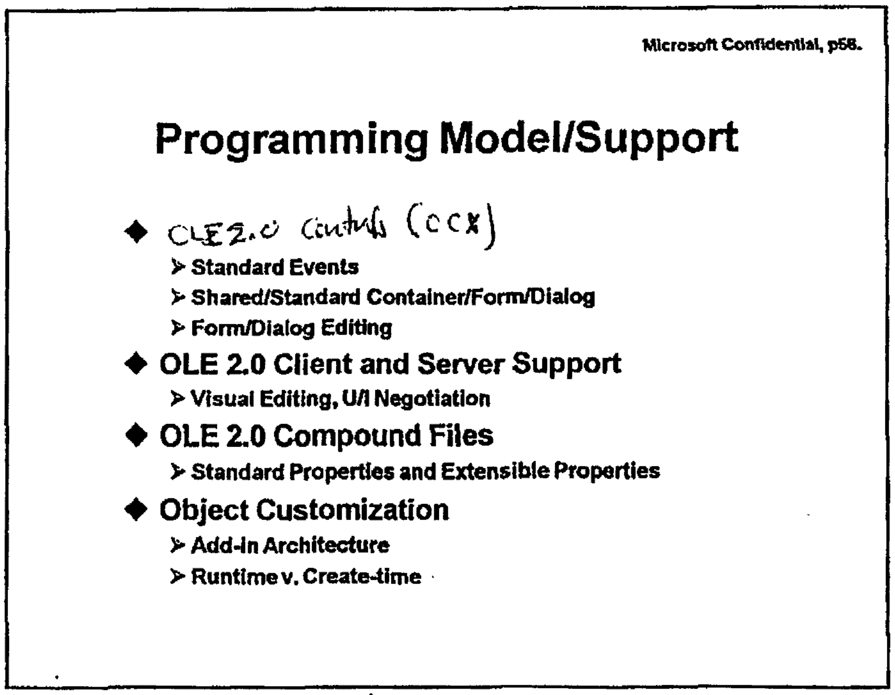
- Serial routing
- Parallel routing
- Re-routing
- Process abstraction
- Delegation
- Consolidation
- Status tracking
- Expedite/hold/cancel



Microsoft Confidential, p57.

Event Model

- Define events
- Raise events
- Respond to events
- Scope of events
- Delivery mechanism
- Naming
- Event security
- Event logging



Microsoft Confidential, p58.

Programming Model/Support

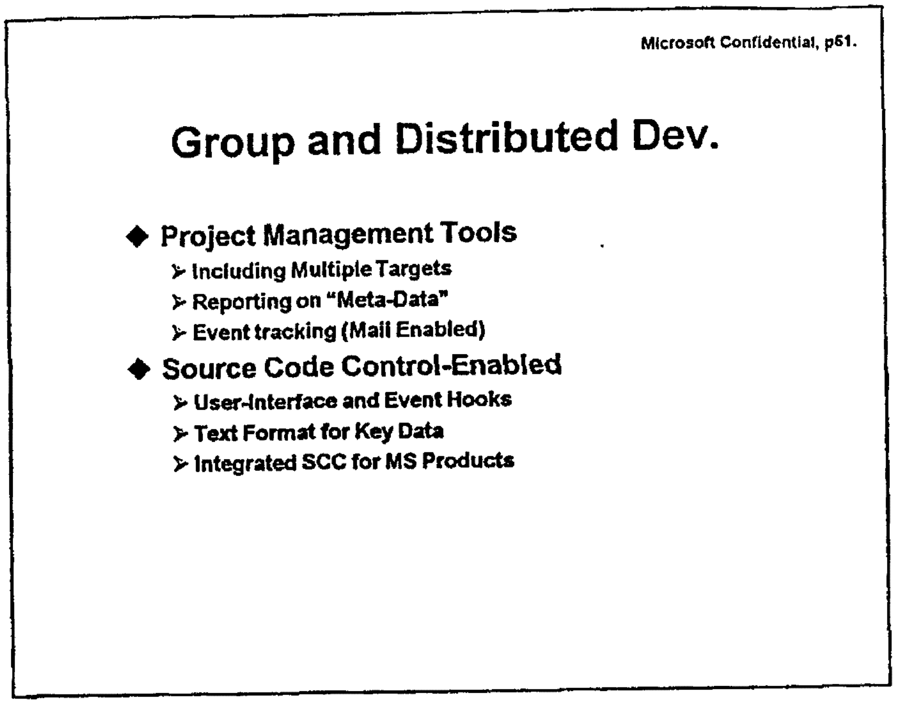
- OLE2.0 Controls (OCX)
  - Standard Events
  - Shared/Standard Container/Form/Dialog
  - Form/Dialog Editing
- OLE 2.0 Client and Server Support
  - Visual Editing, U/I Negotiation
- OLE 2.0 Compound Files
  - Standard Properties and Extensible Properties
- Object Customization
  - Add-in Architecture
  - Runtime v. Create-time

Programming Model/Support

- Interface Definition
  - COM v. OLE Automation
- COM
  - Share Design/Specification of Common "objects"
  - Reduce Number of Unique Classes/Methods
- OLE Automation (Macro Languages)
  - Share Design/Specification of Common "commands"
  - Share Top-Level Operations Code Across Applications

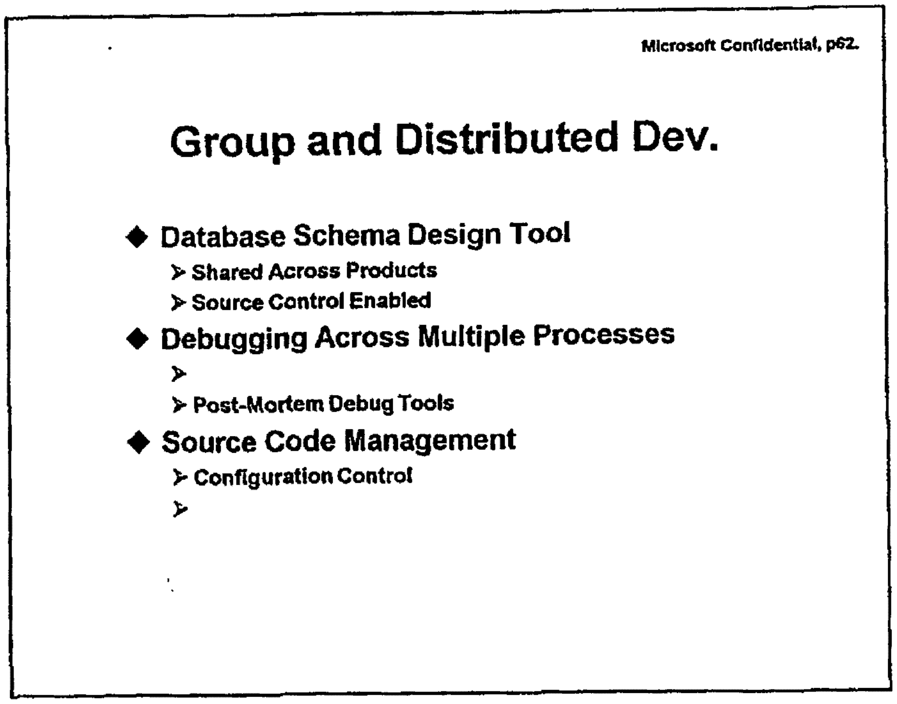
Common Objects, Code, Tools

- Chicago Text Control
- Other Chicago U/I elements
  - Property Sheets, Hierarchy Viewer, Toolbars
  - Common Dialogs
- MFC 2.5's OLE 2.0 support
- VBA hosting
- Dialogue Editor
- Forms runtime
- Other Tools and Filters (OLE 2.0 servers)
- Localization Tools for No-compile Localize
- Wizard Authoring (Internal, Use of VBA)
- Online Documentation and Indexing
- HLP Viewer and Authoring Tools



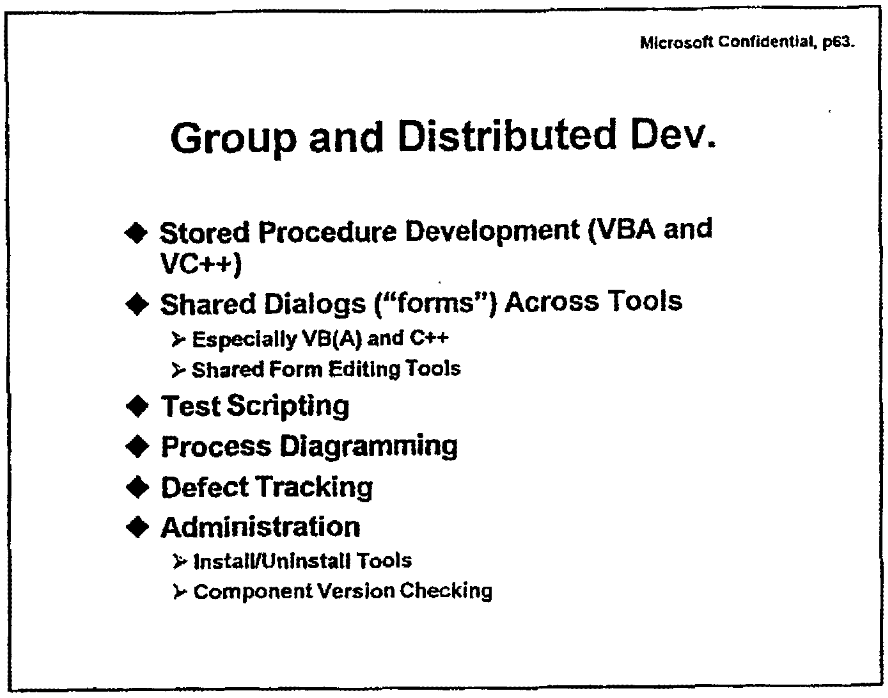
Group and Distributed Dev.

- ◆ Project Management Tools
  - ➢ Including Multiple Targets
  - ➢ Reporting on "Meta-Data"
  - ➢ Event tracking (Mail Enabled)
- ◆ Source Code Control-Enabled
  - ➢ User-Interface and Event Hooks
  - ➢ Text Format for Key Data
  - ➢ Integrated SCC for MS Products



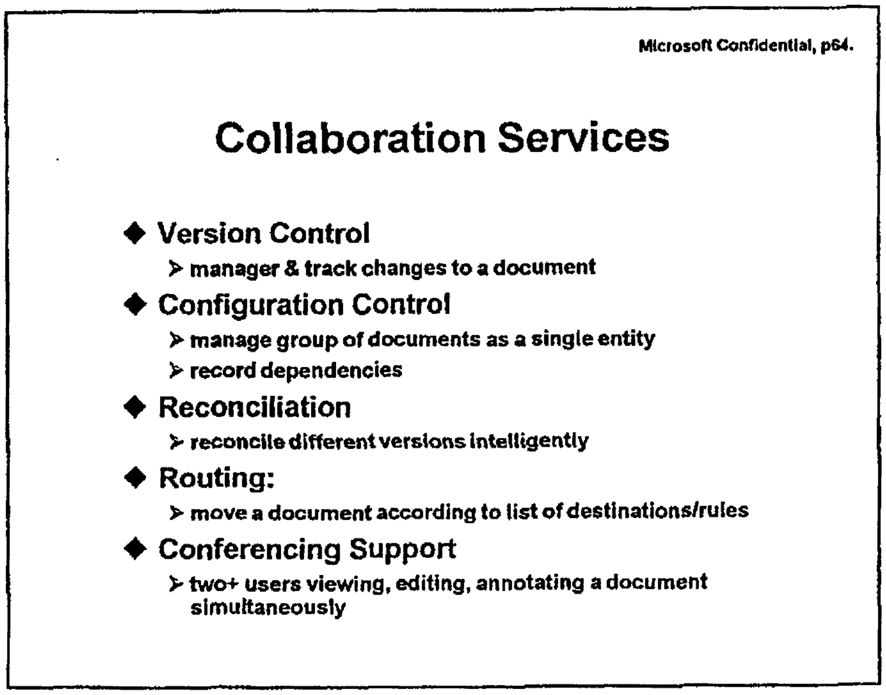
Group and Distributed Dev.

- ◆ Database Schema Design Tool
  - ➢ Shared Across Products
  - ➢ Source Control Enabled
- ◆ Debugging Across Multiple Processes
  - ➢
  - ➢ Post-Mortem Debug Tools
- ◆ Source Code Management
  - ➢ Configuration Control
  - ➢



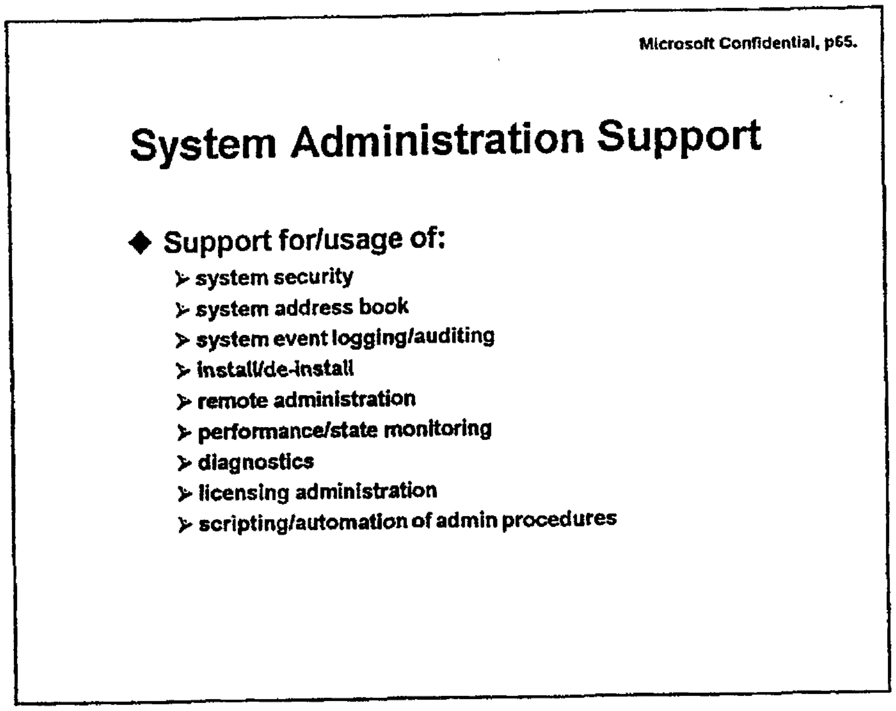
Group and Distributed Dev.

- Stored Procedure Development (VBA and VC++)
- Shared Dialogs ("forms") Across Tools
  - Especially VB(A) and C++
  - Shared Form Editing Tools
- Test Scripting
- Process Diagramming
- Defect Tracking
- Administration
  - Install/Uninstall Tools
  - Component Version Checking



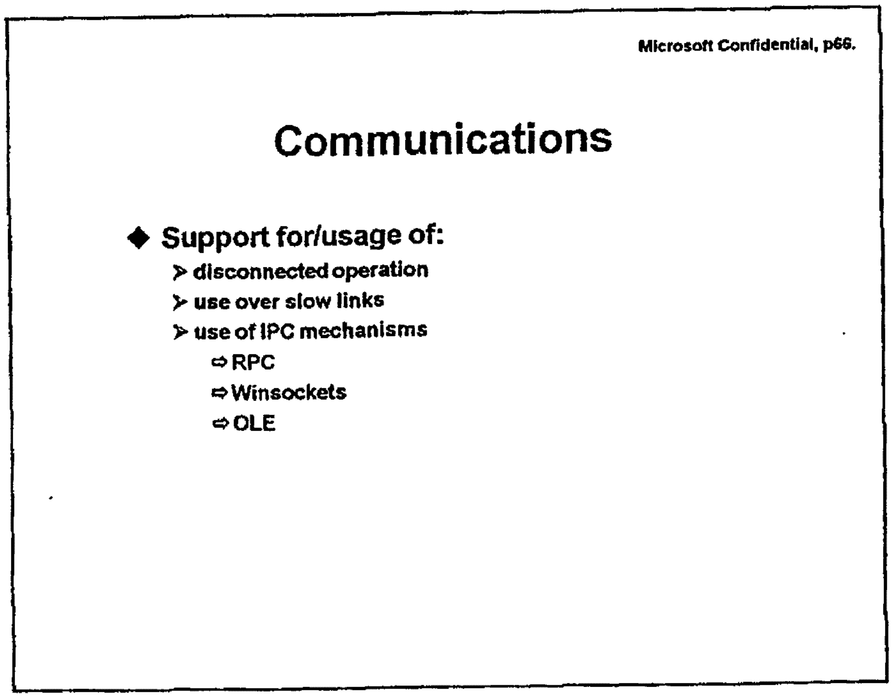
Collaboration Services

- Version Control
  - manager & track changes to a document
- Configuration Control
  - manage group of documents as a single entity
  - record dependencies
- Reconciliation
  - reconcile different versions intelligently
- Routing:
  - move a document according to list of destinations/rules
- Conferencing Support
  - two+ users viewing, editing, annotating a document simultaneously



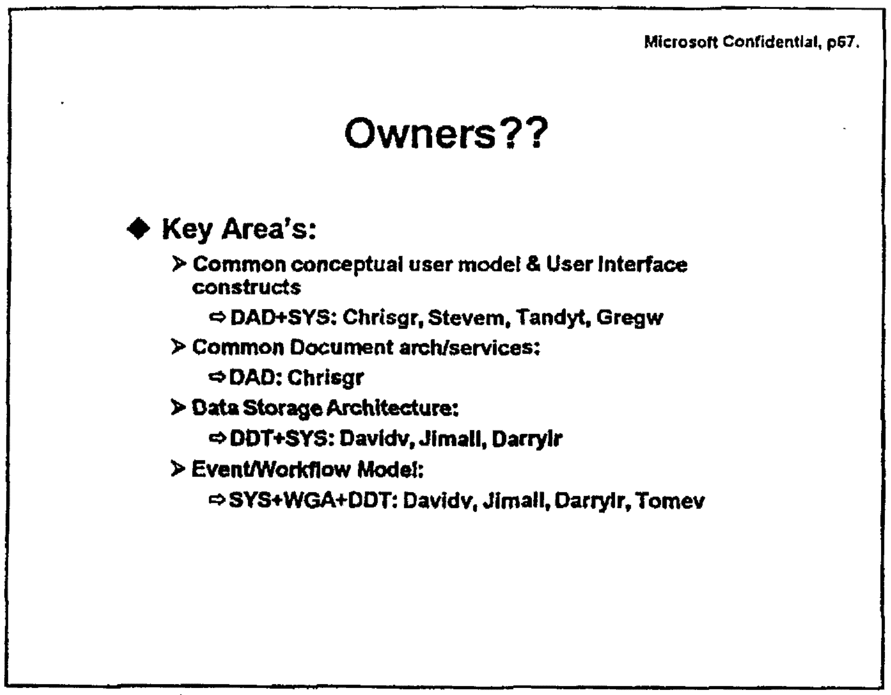
System Administration Support

- Support for/usage of:
  - system security
  - system address book
  - system event logging/auditing
  - install/de-install
  - remote administration
  - performance/state monitoring
  - diagnostics
  - licensing administration
  - scripting/automation of admin procedures



Communications

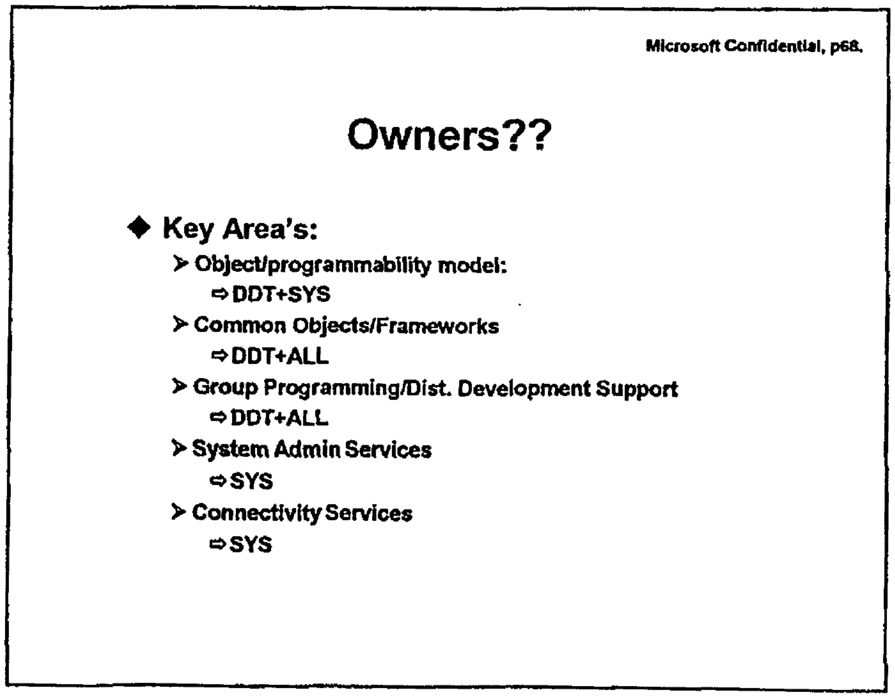
- Support for/usage of:
  - disconnected operation
  - use over slow links
  - use of IPC mechanisms
    - RPC
    - Winsockets
    - OLE



Microsoft Confidential, p67.

Owners??

- **Key Area's:**
  - Common conceptual user model & User Interface constructs
    - ⇨ DAD+SYS: Chrisgr, Stevem, Tandyt, Gregw
  - Common Document arch/services:
    - ⇨ DAD: Chrisgr
  - Data Storage Architecture:
    - ⇨ DDT+SYS: Davidv, Jimall, Darrylr
  - Event/Workflow Model:
    - ⇨ SYS+WGA+DDT: Davidv, Jimall, Darrylr, Tomev



Microsoft Confidential, p68.

Owners??

- **Key Area's:**
  - Object/programmability model:
    - ⇨ DDT+SYS
  - Common Objects/Frameworks
    - ⇨ DDT+ALL
  - Group Programming/Dist. Development Support
    - ⇨ DDT+ALL
  - System Admin Services
    - ⇨ SYS
  - Connectivity Services
    - ⇨ SYS

Follow-up

- List of current efforts/issues from 3 year plans
- Owners to work with Archiecture group to:
  - document preferred architecture and direction
  - schedule reviews
  - develop steps of objectives for product units