COM on

Developing Component based Information Systems

with tools supporting the Microsoft Component Object Model

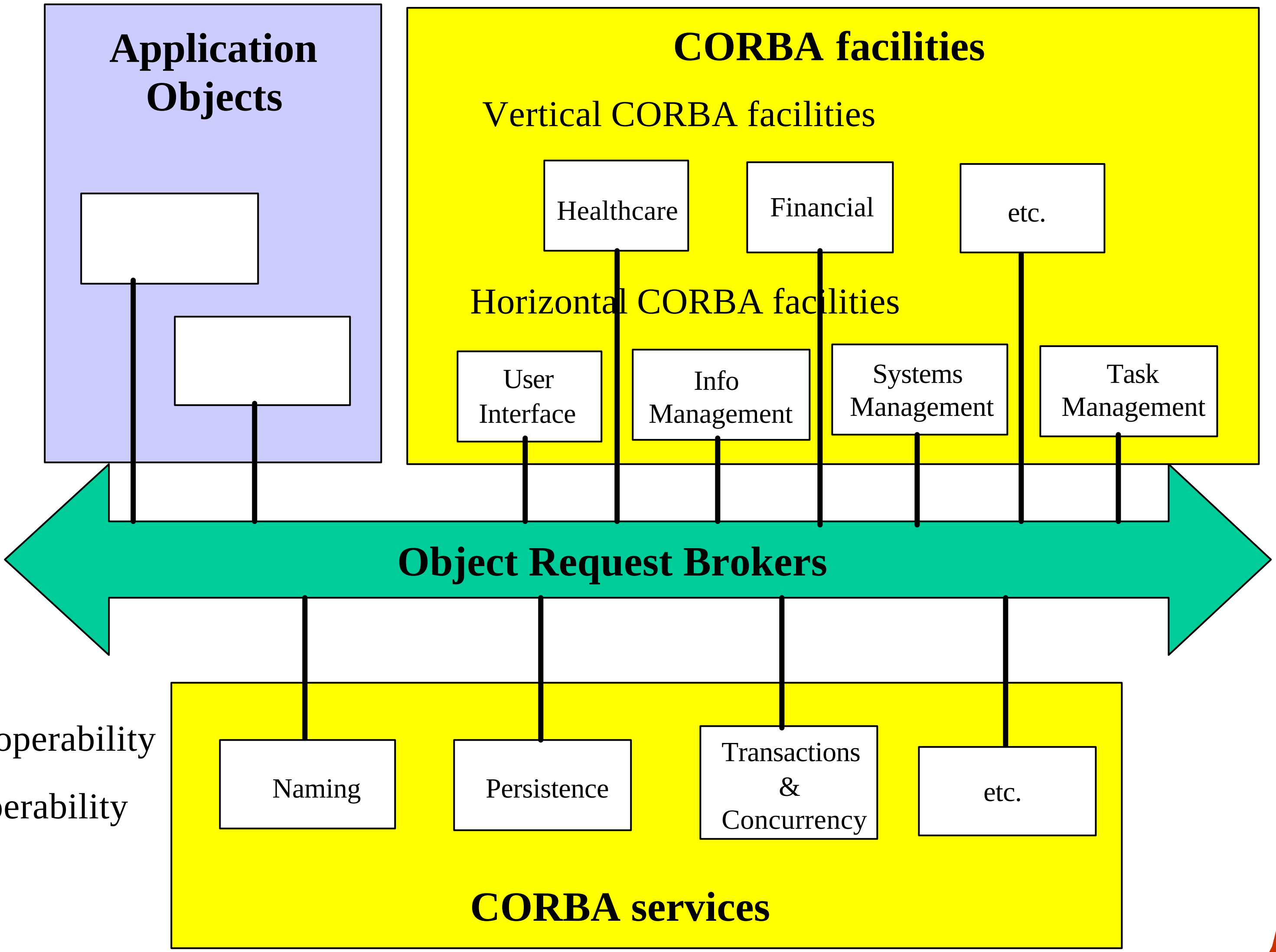
Egil P.Andersen
Norwegian Computing Center
P.O.Box 114, Blindern, 0314 Oslo, Norway
Tel: +47 22 85 25 94, Fax: +47 22 69 76 60
Egil.Paulin.Andersen@nr.no

Outline

Some tools and technologies for the development of Component based Information Systems

- Object Management Architecture, CORBA (Common Object Request Broker Architecture)
- Microsoft COM (Component Object Model), DCOM (Distributed COM)
- Component Object Models, IDL (Interface Description Language)
- Layered Architectures, 3-tier/n-tier, Business Objects
- Rational Rose/UML (Unified Modelling Language)
- Visual Basic, Visual C++, ATL (Active Template Library), J++ (MS Java)
- Compound Documents, ActiveX Controls, ActiveX Documents
- Universal Data Access, OLE DB, ADO (ActiveX Data Objects)
- MTS (Microsoft Transaction Server)
- IIS (Internet Information Server), ASP (Active Server Pages), Scripting
- SOAP - Simple Object Access Protocol
- XML, DTD (Document Type Definition), DOM (Document Object Model), XSL (eXstensible Stylesheet Language)
- Microsoft Repository, Visual Component Manager (VCM)
- Microsoft Message Queue Server (MSMQ)
- OODB (??)
- …..and more….puh….

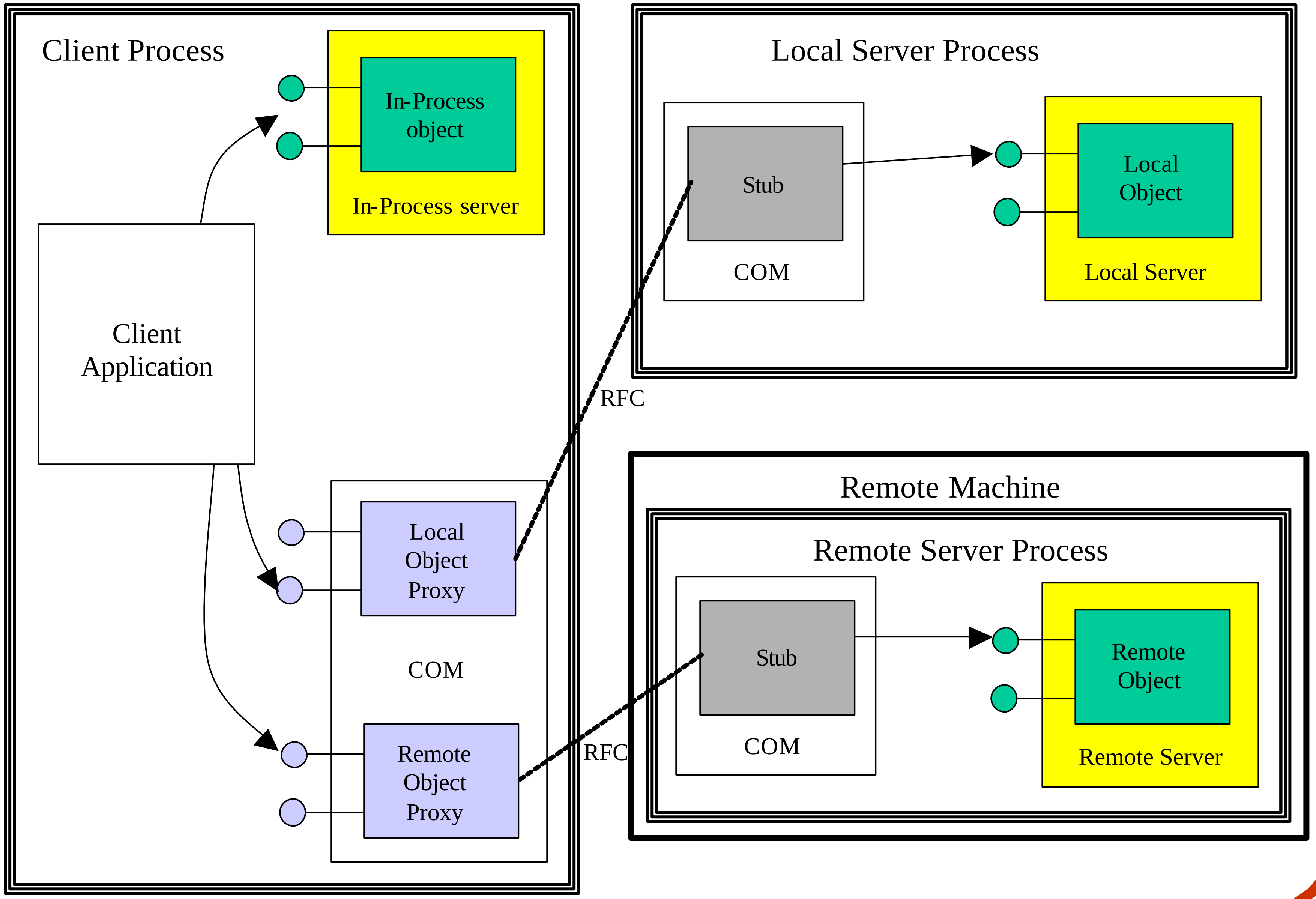
OMG's Object Management Architecture



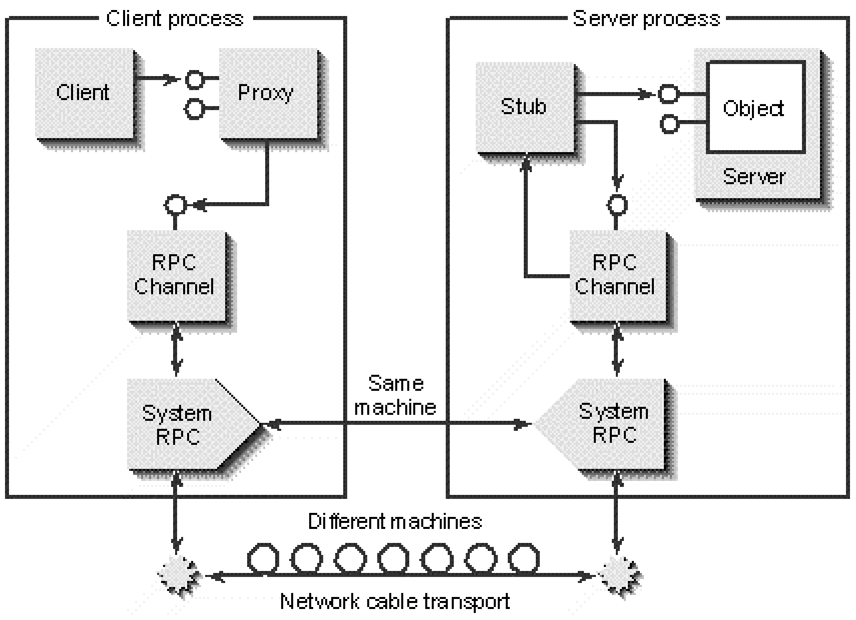
- Syntactical interoperability
- Semantic interoperability

OMG - Object Management Group CORBA - Common Object Request Broker Architecture

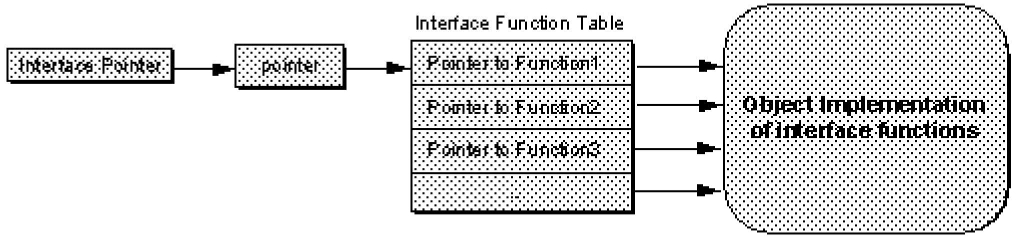
Local in-process, Local out-of-process, Remote



Marshalling for Out-of-Process Components



Basic COM (Component Object Model)



- **VTable interfaces -** a binary standard with interfaces based on a memory layout corresponding to that of abstract classes in C++

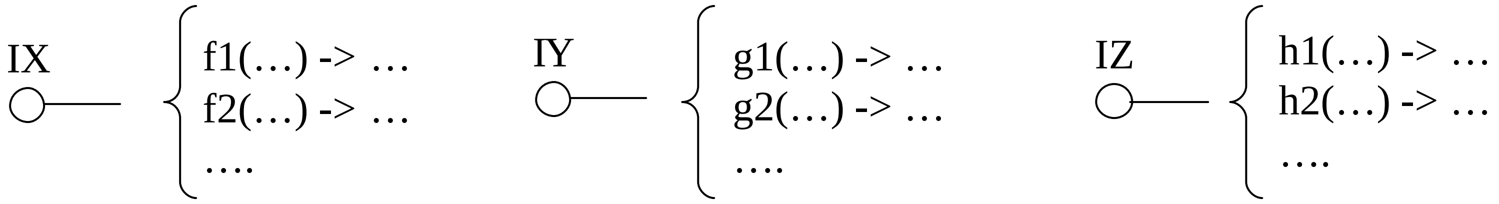
  A COM interface and its functions is similar to an abstract base class with a set of virtual functions in C++

  The extra level of indirection provides flexibility with respect to how interfaces are implemented.

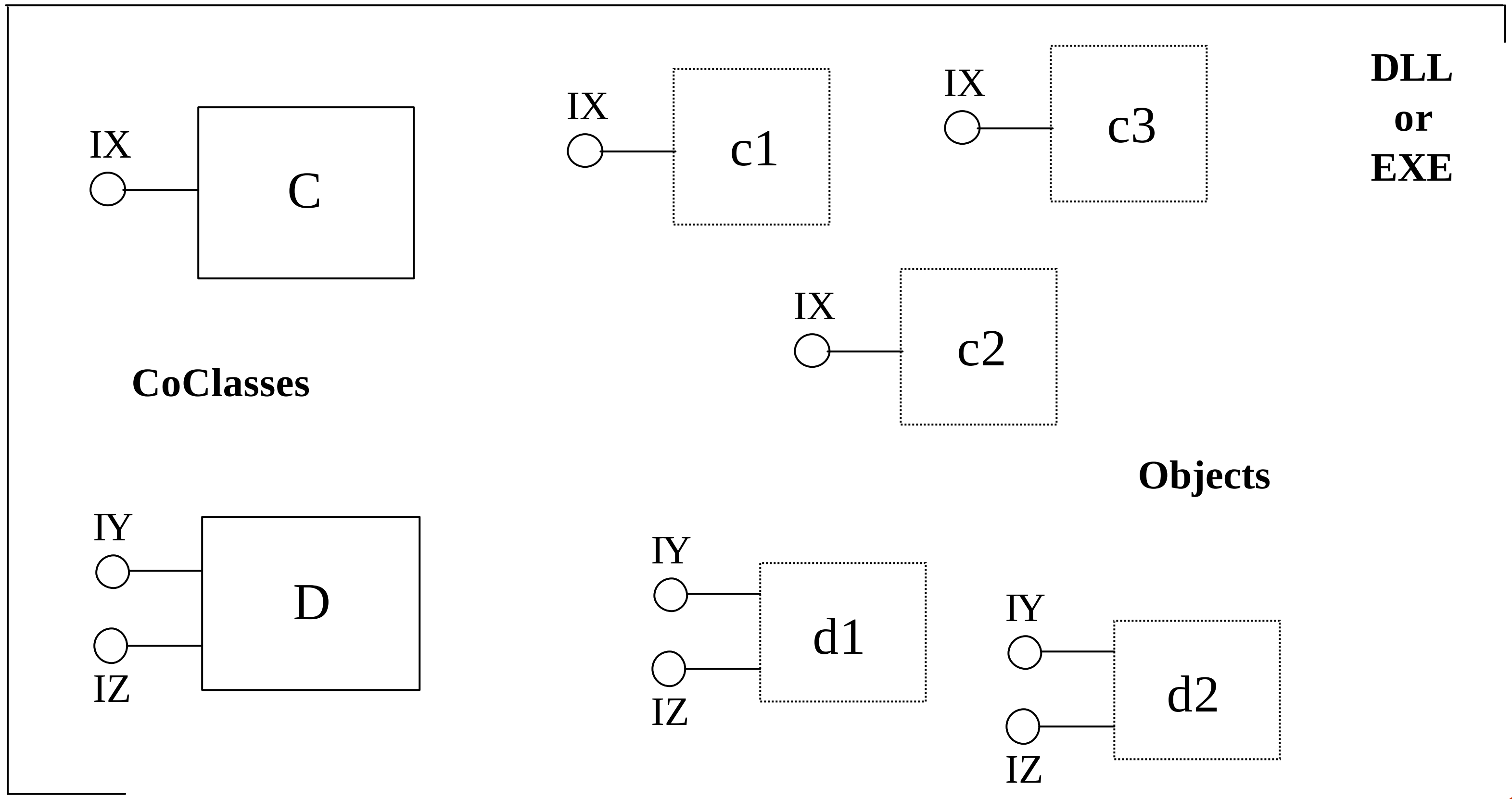
- **Dispatch interfaces -** query the interface for its functions and their signatures

- **Dual interfaces -** available both for efficient vtable access and for scripting languages

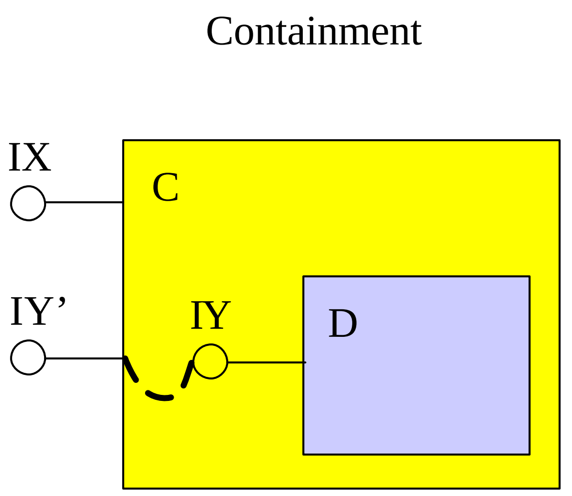
Interfaces, Components/CoClasses, Objects, GUID (Globally Unique Identifiers), CLSID, IID

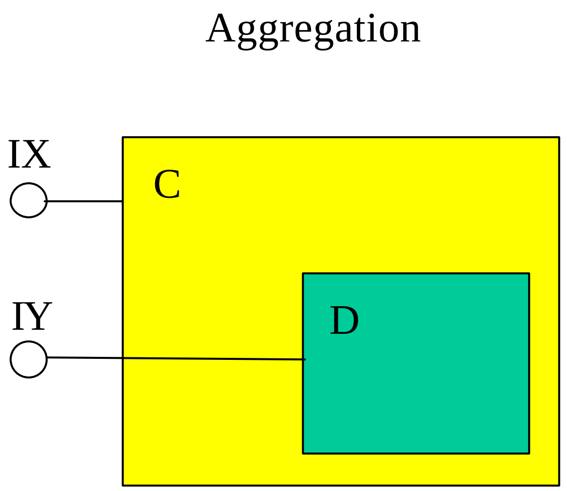


Interfaces: Versioning - Multiple interfaces - Single inheritance - IUnknown



Integrating COM Components via Containment vs Aggregation

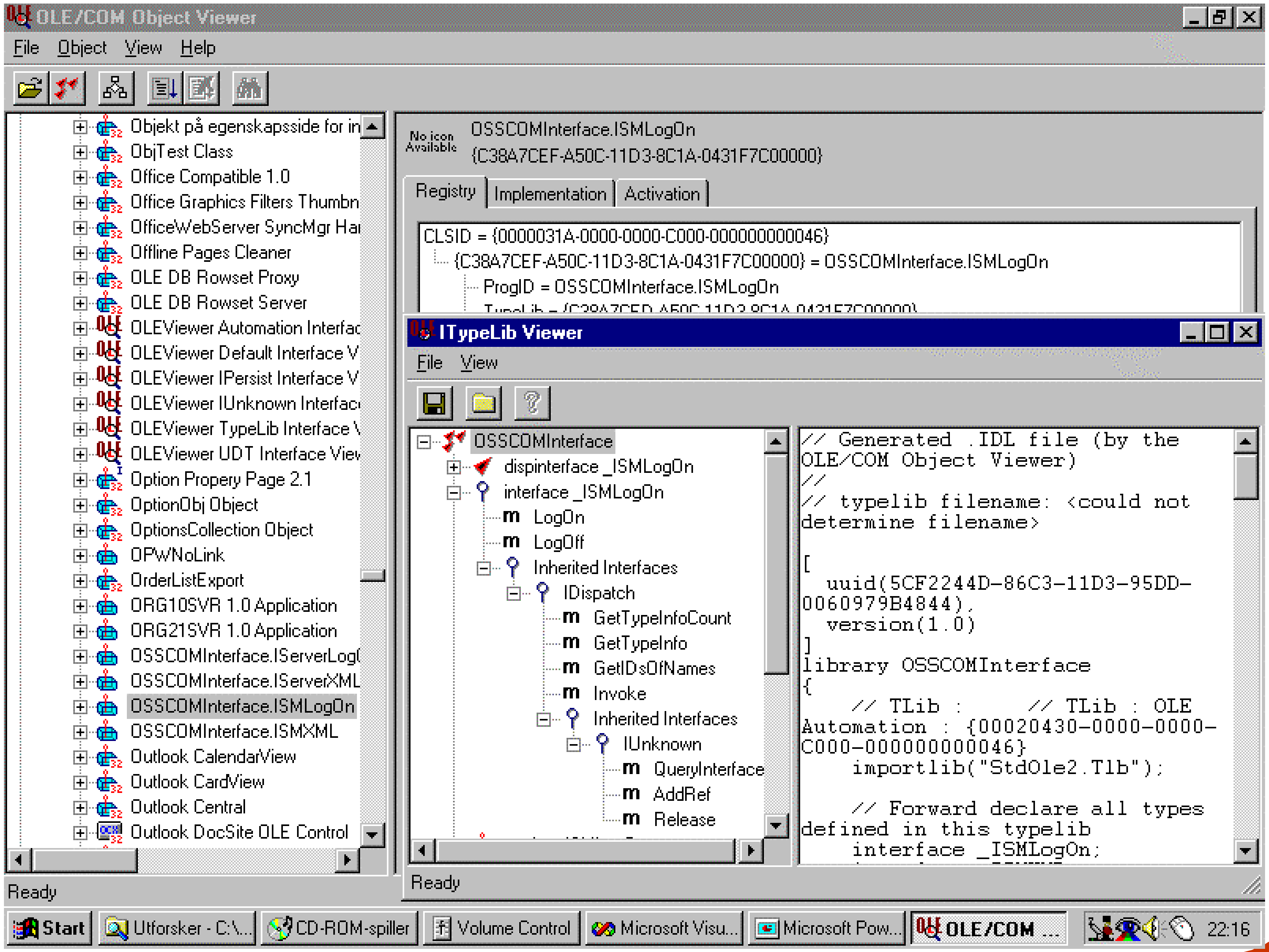




IDL - Interface Definition Language

```
[ object,
  uuid(EA762187-A99A-11d3-95F4-0060979B4844),
   oleautomation,
   dual,
   …..]
interface IOSSSMLogin : IDispatch
{[id(1), helpstring("Function LogOn")]
     HRESULT LogOn([in] BSTR user, [in] BSTR pwd, [out] VARIANT_BOOL* okLogOn);

  [id(2), helpstring("Function LogOff")]
     HRESULT LogOff([out] VARIANT_BOOL* okLogOff);
};

[ object,
   uuid(EA762188-A99A-11d3-95F4-0060979B4844),
   oleautomation,
   dual,
   ….]
interface IOSSSMXML : IDispatch
{ [id(1), helpstring("Function GetRecordInfo")]
     HRESULT GetRecordInfo([in] long recordID, [in] short retrievalMode,
                                              [in] VARIANT_BOOL getHTML, [out] BSTR* XMLString);
    ….. };
```

Information on Interfaces and Components


OLE/COM Object Viewer
File Object View Help
Registry Implementation Activation
ITypeLib Viewer
File View
Ready
Start Utforsker - C:\... CD-ROM-spiller Volume Control Microsoft Visu... Microsoft Pow... OLE/COM ... 22:16

```
Objekt på egenskapsside for in
ObjTest Class
Office Compatible 1.0
Office Graphics Filters Thumbn
OfficeWebServer SyncMgr Han
Offline Pages Cleaner
OLE DB Rowset Proxy
OLE DB Rowset Server
OLEViewer Automation Interfac
OLEViewer Default Interface V
OLEViewer IPersist Interface V
OLEViewer IUnknown Interfac
OLEViewer TypeLib Interface V
OLEViewer UDT Interface View
Option Propery Page 2.1
OptionObj Object
OptionsCollection Object
OPWNoLink
OrderListExport
ORG10SVR 1.0 Application
ORG21SVR 1.0 Application
OSSCOMInterface.IServerLogO
OSSCOMInterface.IServerXML
OSSCOMInterface.ISMLogOn
OSSCOMInterface.ISMXML
Outlook CalendarView
Outlook CardView
Outlook Central
Outlook DocSite OLE Control
```

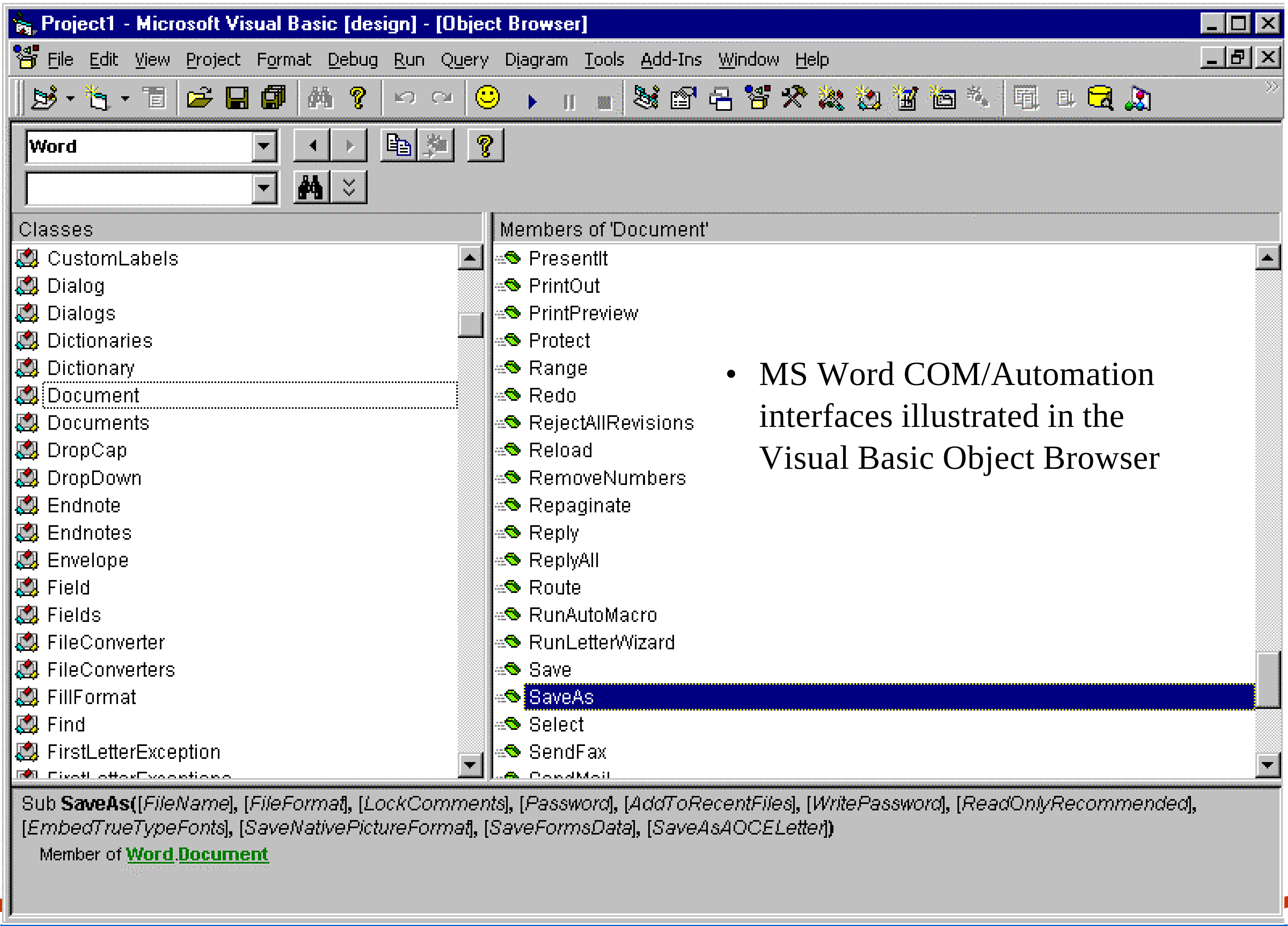
```
No icon Available   OSSCOMInterface.ISMLogOn
                    {C38A7CEF-A50C-11D3-8C1A-0431F7C00000}

CLSID = {0000031A-0000-0000-C000-000000000046}
  {C38A7CEF-A50C-11D3-8C1A-0431F7C00000} = OSSCOMInterface.ISMLogOn
    ProgID = OSSCOMInterface.ISMLogOn
```

```
OSSCOMInterface
  dispinterface _ISMLogOn
  interface _ISMLogOn
    m LogOn
    m LogOff
    Inherited Interfaces
      IDispatch
        m GetTypeInfoCount
        m GetTypeInfo
        m GetIDsOfNames
        m Invoke
        Inherited Interfaces
          IUnknown
            m QueryInterface
            m AddRef
            m Release
```

```
// Generated .IDL file (by the
OLE/COM Object Viewer)
//
// typelib filename: <could not
determine filename>

[
  uuid(5CF2244D-86C3-11D3-95DD-
0060979B4844),
  version(1.0)
]
library OSSCOMInterface
{
    // TLib :     // TLib : OLE
Automation : {00020430-0000-0000-
C000-000000000046}
    importlib("StdOle2.Tlb");

    // Forward declare all types
defined in this typelib
    interface _ISMLogOn;
```

Component Object Models

- In component based systems an object model consists of classes, interfaces, functions, etc, typically specified by an IDL (interface definition language).


Project1 - Microsoft Visual Basic [design] - [Object Browser]
File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help
Word

Classes

- CustomLabels
- Dialog
- Dialogs
- Dictionaries
- Dictionary
- Document
- Documents
- DropCap
- DropDown
- Endnote
- Endnotes
- Envelope
- Field
- Fields
- FileConverter
- FileConverters
- FillFormat
- Find
- FirstLetterException

Members of 'Document'

- PresentIt
- PrintOut
- PrintPreview
- Protect
- Range
- Redo
- RejectAllRevisions
- Reload
- RemoveNumbers
- Repaginate
- Reply
- ReplyAll
- Route
- RunAutoMacro
- RunLetterWizard
- Save
- SaveAs
- Select
- SendFax

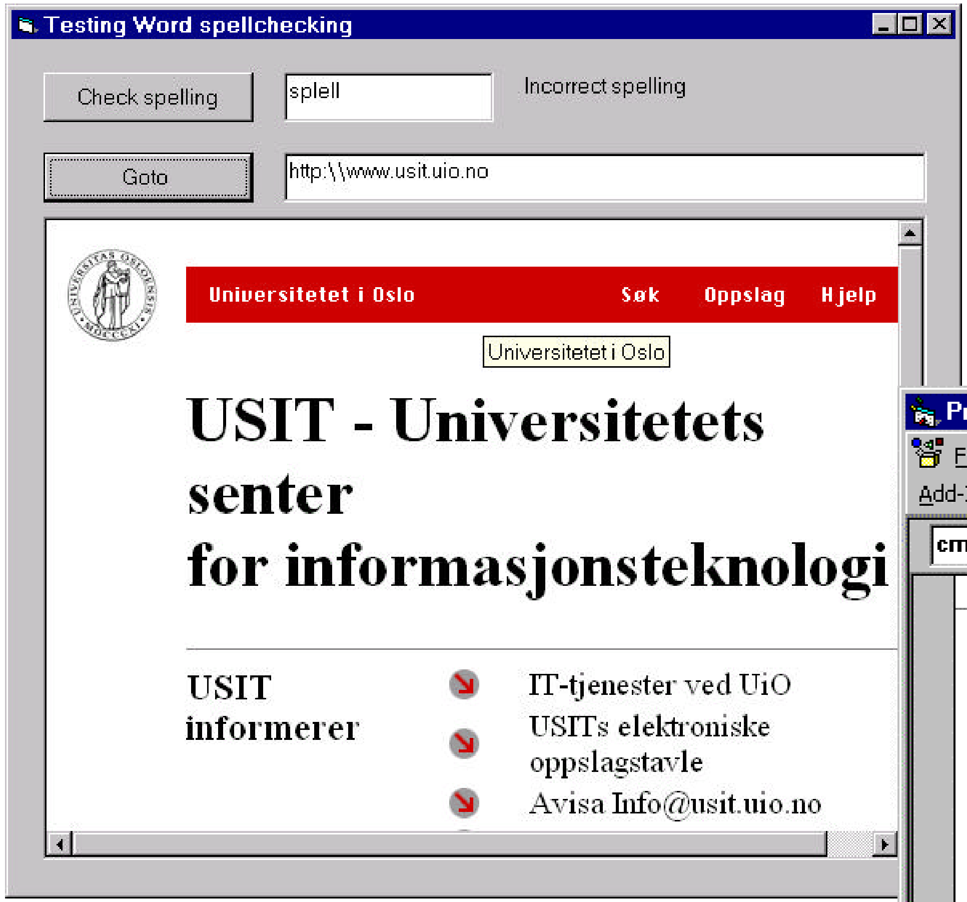
Sub **SaveAs**([*FileName*], [*FileFormat*], [*LockComments*], [*Password*], [*AddToRecentFiles*], [*WritePassword*], [*ReadOnlyRecommended*], [*EmbedTrueTypeFonts*], [*SaveNativePictureFormat*], [*SaveFormsData*], [*SaveAsAOCELetter*])
Member of Word.Document

- MS Word COM/Automation interfaces illustrated in the Visual Basic Object Browser

Programming Languages and Development Environment

- **Microsoft Visual Studio** - an elaborate development environment
- **Visual Basic** - very(!) easy to learn and use - inflexible - performance
- **Visual C++** - powerful and flexible - complex - wizzardmania….
- **Visual J++** - no experience with it…..
- **ATL (Active Template Library)** - utility for creating COM components in VC++

Compound Documents
with ActiveX Controls and ActiveX Documents



splell

Incorrect spelling

http:\\www.usit.uio.no

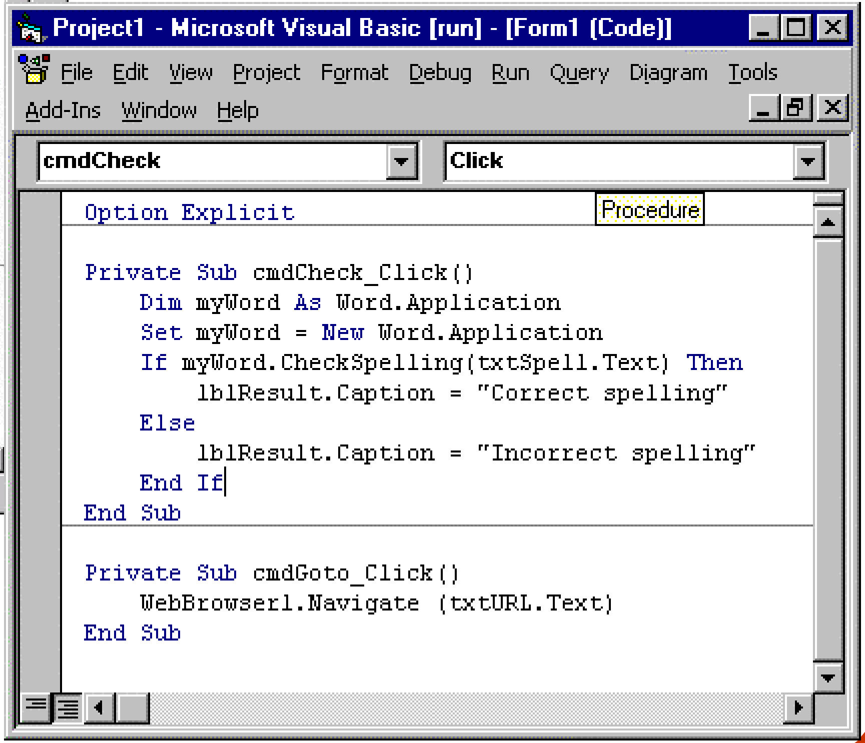
USIT - Universitetets senter for informasjonsteknologi

USIT informerer

IT-tjenester ved UiO

USITs elektroniske oppslagstavle

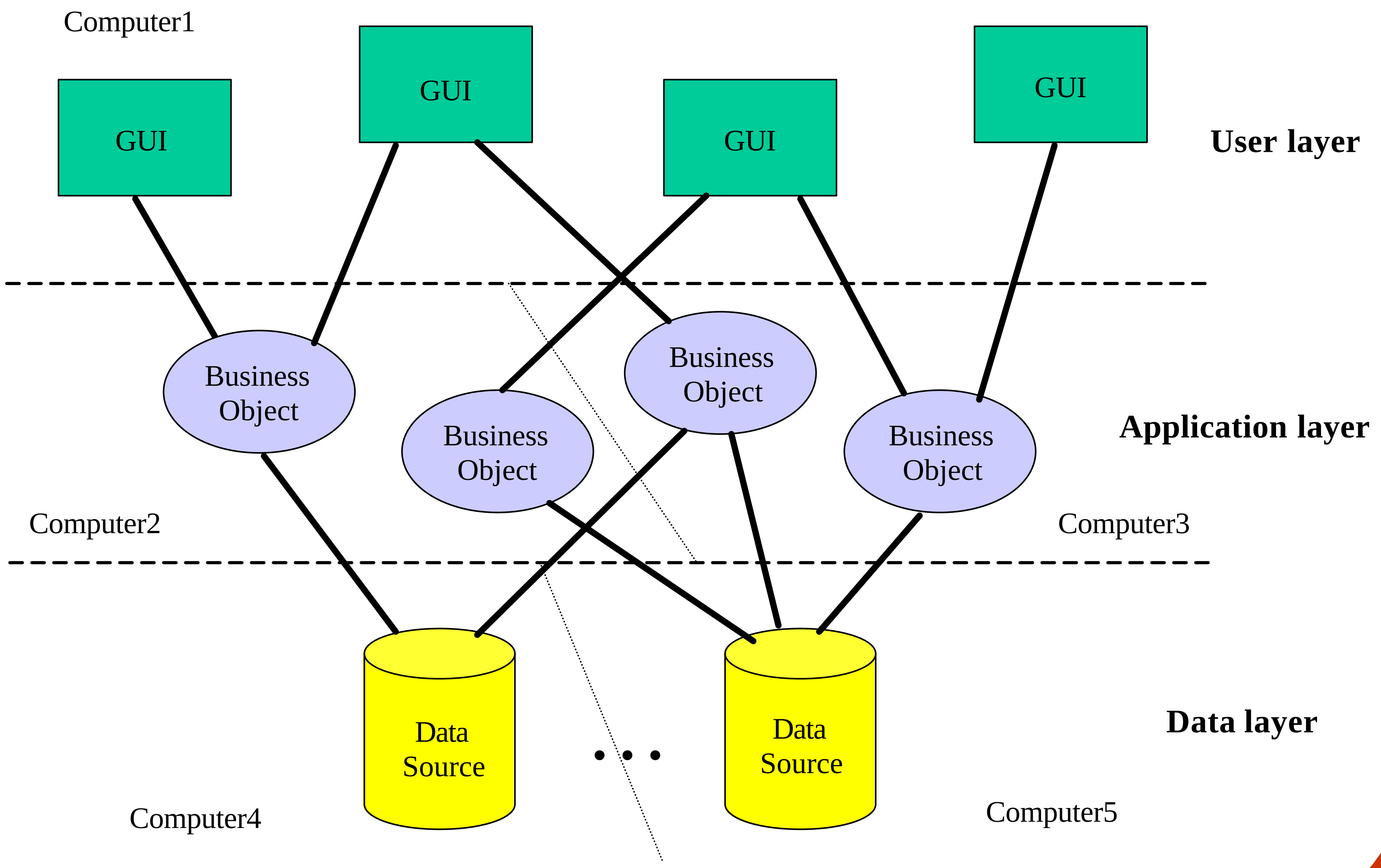
Avisa Info@usit.uio.no



Project1 - Microsoft Visual Basic [run] - [Form1 (Code)]
File Edit View Project Format Debug Run Query Diagram Tools
Add-Ins Window Help
cmdCheck
Click
Procedure

```
Option Explicit

Private Sub cmdCheck_Click()
    Dim myWord As Word.Application
    Set myWord = New Word.Application
    If myWord.CheckSpelling(txtSpell.Text) Then
        lblResult.Caption = "Correct spelling"
    Else
        lblResult.Caption = "Incorrect spelling"
    End If
End Sub

Private Sub cmdGoto_Click()
    WebBrowser1.Navigate (txtURL.Text)
End Sub
```

Layered Architectures - 3-tier/n-tier



Characteristics of Rational Rose/UML

+ “Mainstream” - well-known and seen as a standard
+ Information modelling and explicit object interaction modelling
+ Object model available via COM/automation - it can be extended and customised
+ Code generation (but **not** production code…)
+ Informal (is this a plus??)

÷ Business rules and behaviour other than explicit object interaction

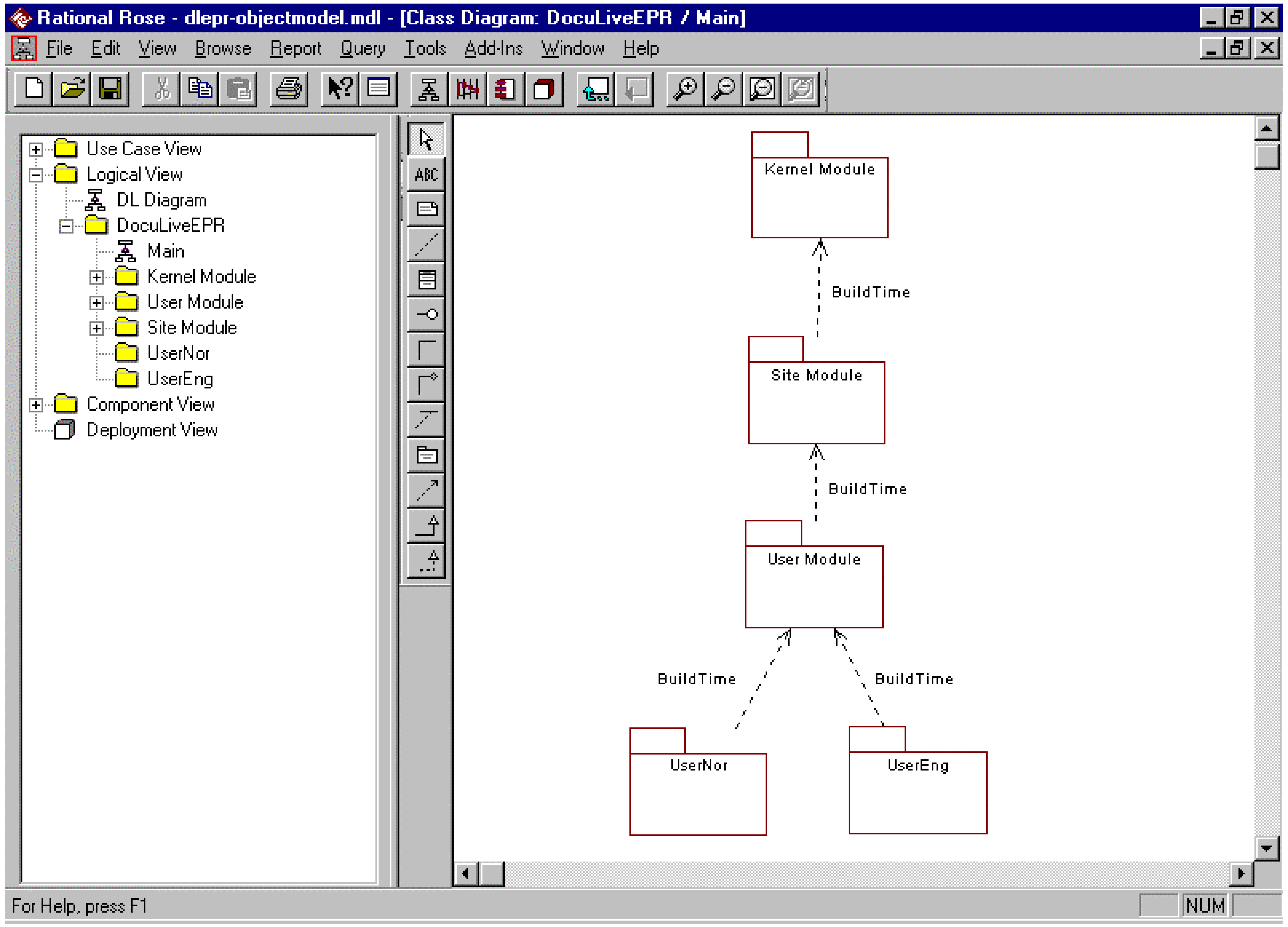
÷ Conceptual errors cannot be detected - models are not correct/incorrect - no modelling tool can distinguish good from bad models (and this is difficult also for experienced modellers)

÷ Incomplete

÷ Slightly confusing organization (at least at first…)

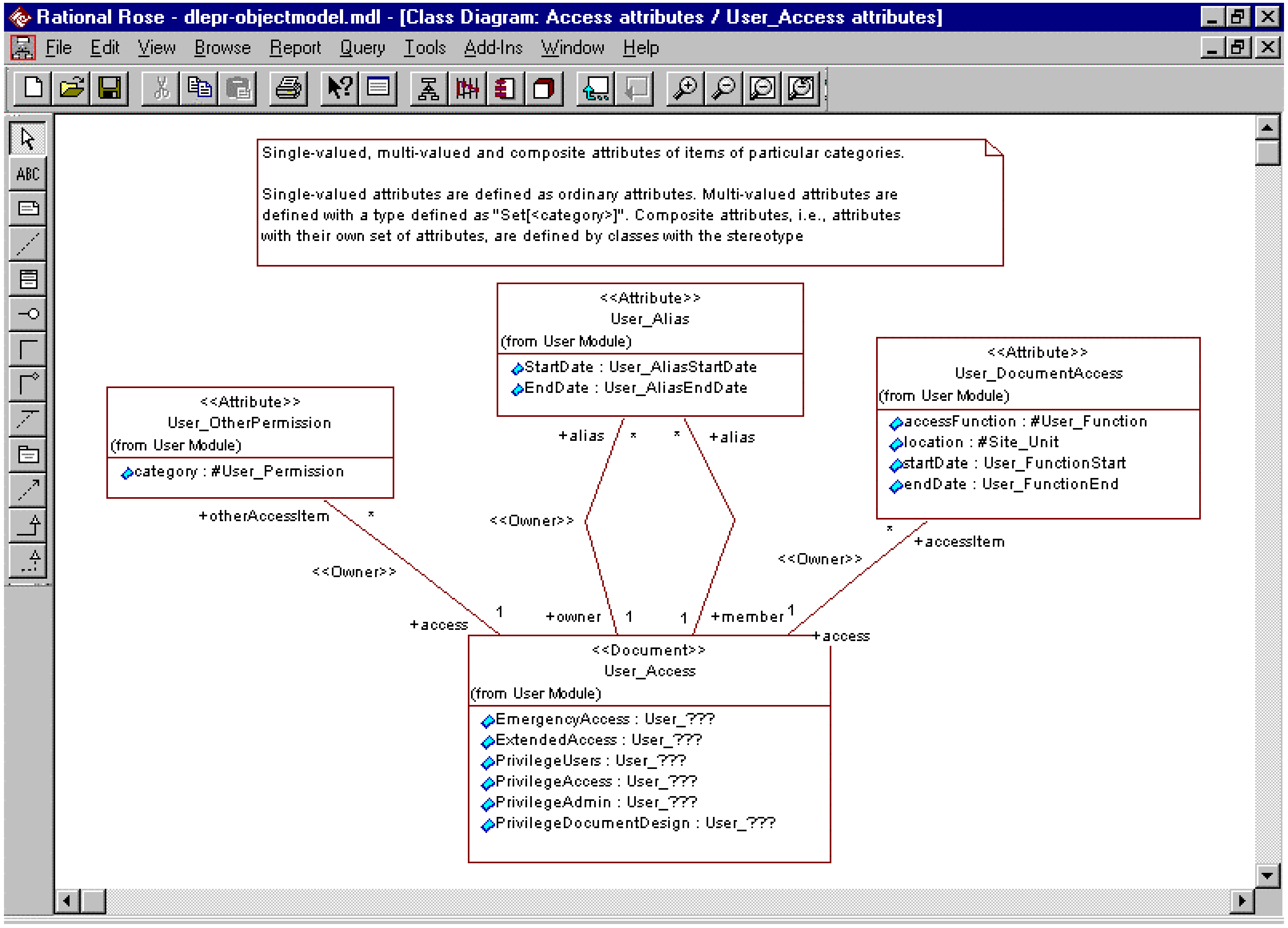
- Consider it mainly as a drawing tool and as a model repository
- Use only those parts that are well understood/agreed upon, and use it consistently - do **not** “over-model”
- Modelling syntax is not essential, but you are not likely to do e.g. Class Diagrams any better...
- Assuming that analysis/design is essential to large-scale software development, then a modelling tool can be useful to establish good routines for planning and documentation, and as a means for unambigous communication internally and externally.

Rose (cont.) - a model is organized into a set of logical packages



NR

Rose (cont.) - Class Diagrams - Information Models


Rational Rose - dlepr-objectmodel.mdl - [Class Diagram: Access attributes / User_Access attributes]
File Edit View Browse Report Query Tools Add-Ins Window Help

Single-valued, multi-valued and composite attributes of items of particular categories.

Single-valued attributes are defined as ordinary attributes. Multi-valued attributes are defined with a type defined as "Set[<category>]". Composite attributes, i.e., attributes with their own set of attributes, are defined by classes with the stereotype

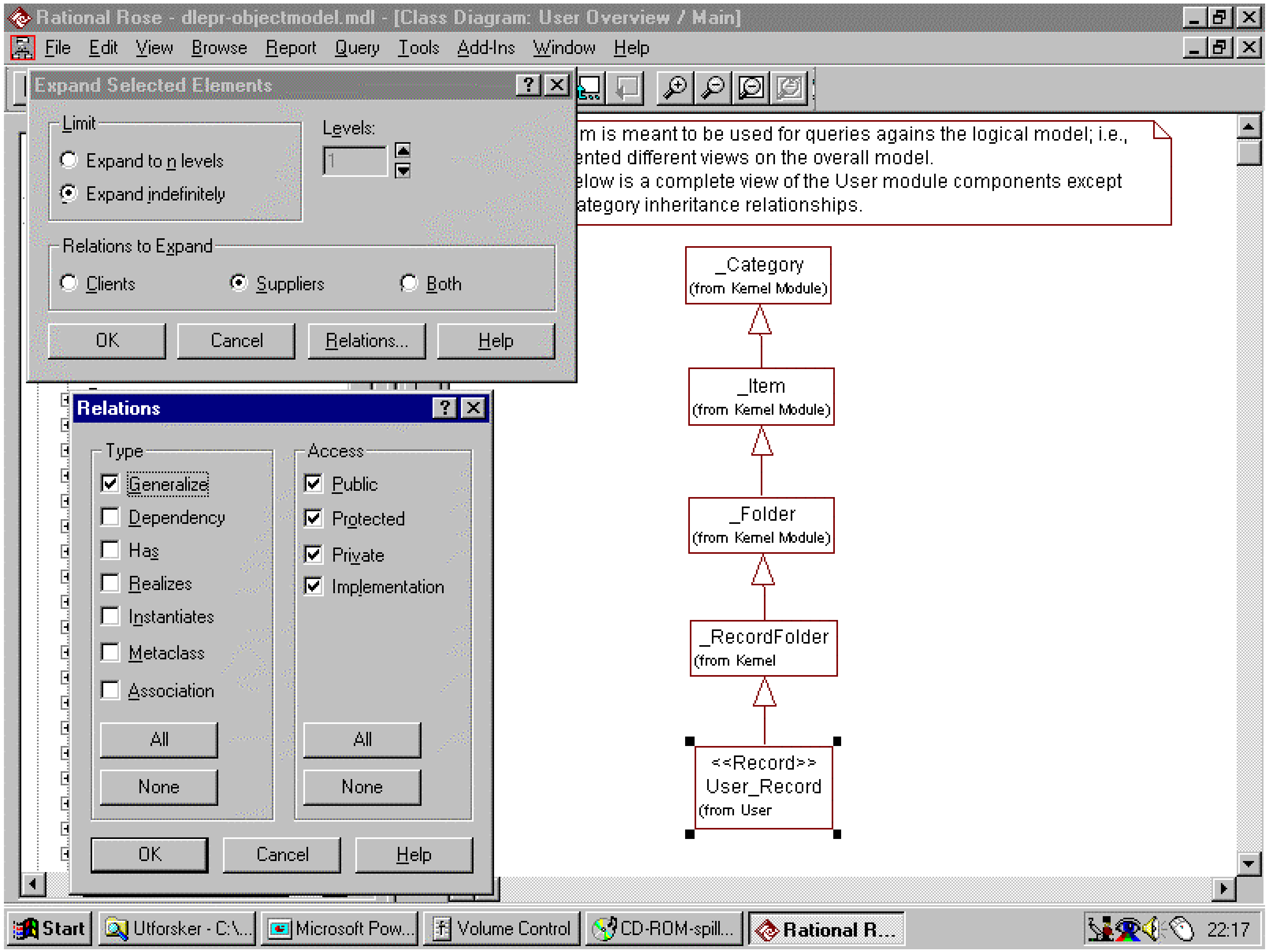
<<Attribute>>
User_Alias
(from User Module)
StartDate : User_AliasStartDate
EndDate : User_AliasEndDate

<<Attribute>>
User_OtherPermission
(from User Module)
category : #User_Permission

<<Attribute>>
User_DocumentAccess
(from User Module)
accessFunction : #User_Function
location : #Site_Unit
startDate : User_FunctionStart
endDate : User_FunctionEnd

<<Document>>
User_Access
(from User Module)
EmergencyAccess : User_???
ExtendedAccess : User_???
PrivilegeUsers : User_???
PrivilegeAccess : User_???
PrivilegeAdmin : User_???
PrivilegeDocumentDesign : User_???

Rose (cont.) - customised views on the model



Rational Rose - dlepr-objectmodel.mdl - [Class Diagram: User Overview / Main]

Expand Selected Elements

Limit
- ○ Expand to n levels
- ● Expand indefinitely

Levels: 1

Relations to Expand: ○ Clients [● Suppliers] ○ Both

OK | Cancel | Relations... | Help

Relations

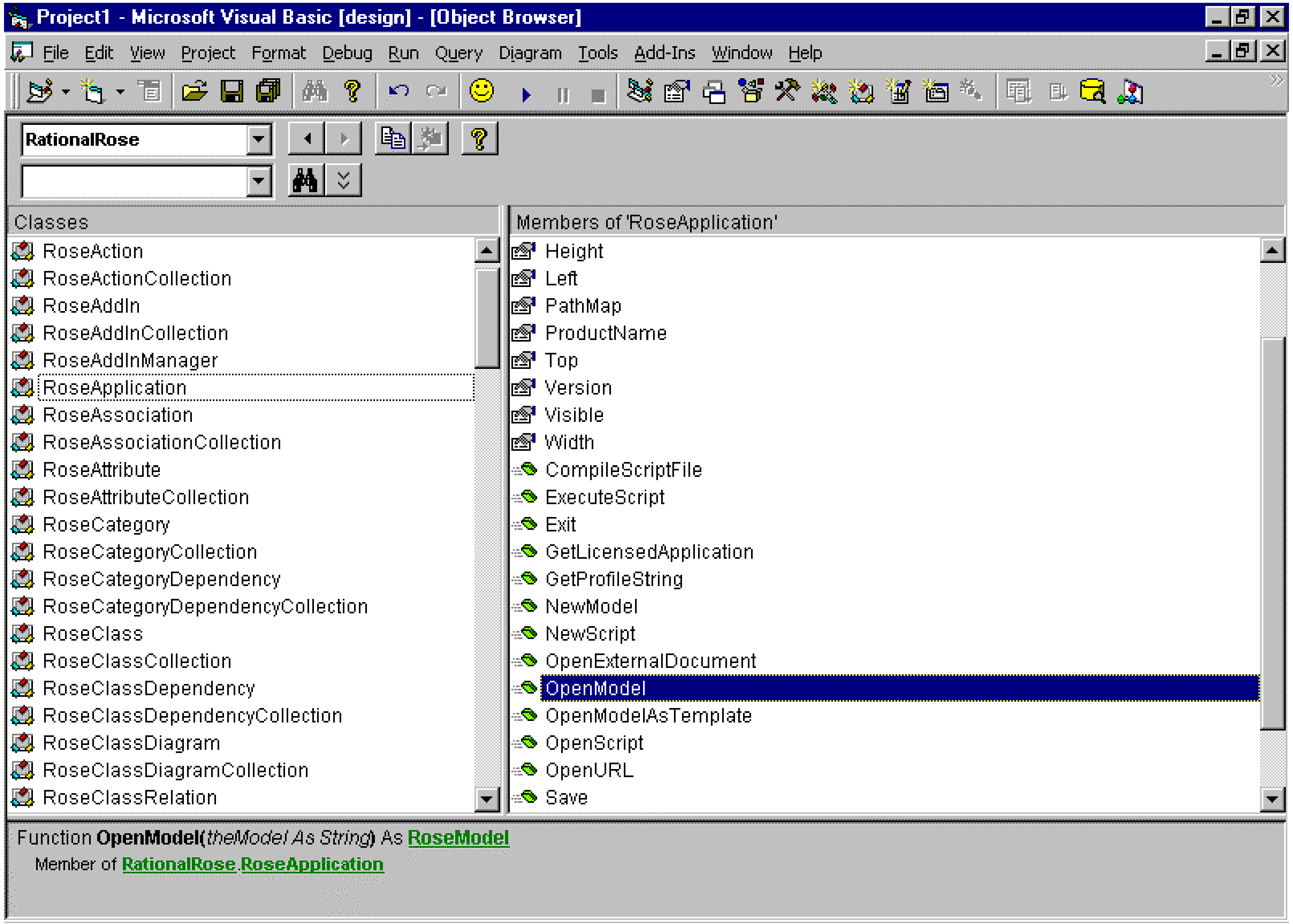
Type	Access
☑ Generalize	☑ Public
☐ Dependency	☑ Protected
☐ Has	☑ Private
☐ Realizes	☑ Implementation
☐ Instantiates	
☐ Metaclass	
☐ Association	

OK | Cancel | Help

m is meant to be used for queries agains the logical model; i.e., ented different views on the overall model. elow is a complete view of the User module components except ategory inheritance relationships.

File
Edit
View
Browse
Report
Query
Tools
Add-Ins
Window
Help
_Category (from Kernel Module)
_Item (from Kernel Module)
_Folder (from Kernel Module)
_RecordFolder (from Kernel
<<Record>> User_Record (from User
Start
Utforsker - C:\...
Microsoft Pow...
Volume Control
CD-ROM-spill...
Rational R...
22:17

Rose (cont.) - COM/Automation - the Rational Rose Object Model

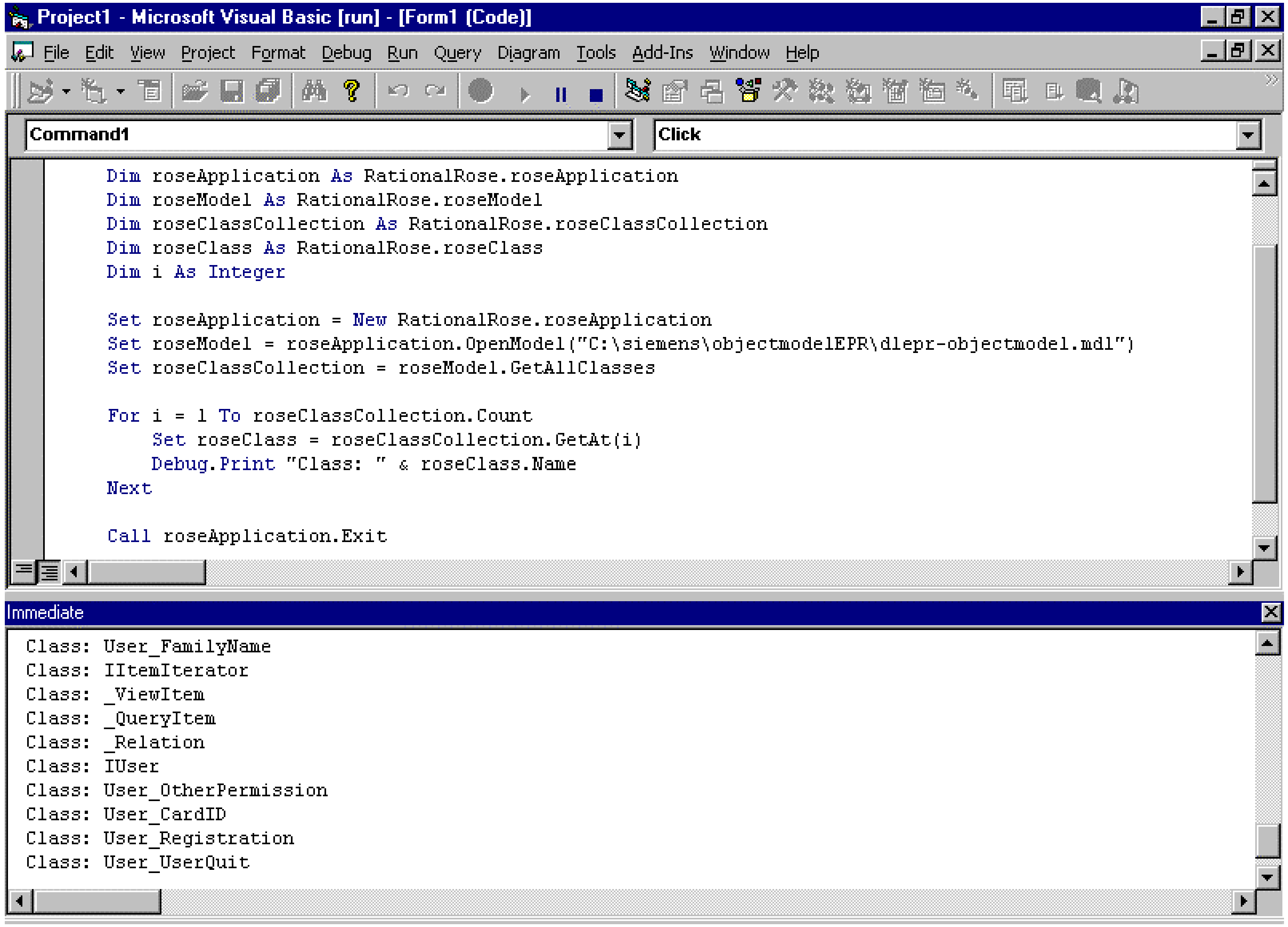


Project1 - Microsoft Visual Basic [design] - [Object Browser]
File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help
RationalRose

Classes	Members of 'RoseApplication'
RoseAction	Height
RoseActionCollection	Left
RoseAddIn	PathMap
RoseAddInCollection	ProductName
RoseAddInManager	Top
RoseApplication	Version
RoseAssociation	Visible
RoseAssociationCollection	Width
RoseAttribute	CompileScriptFile
RoseAttributeCollection	ExecuteScript
RoseCategory	Exit
RoseCategoryCollection	GetLicensedApplication
RoseCategoryDependency	GetProfileString
RoseCategoryDependencyCollection	NewModel
RoseClass	NewScript
RoseClassCollection	OpenExternalDocument
RoseClassDependency	OpenModel
RoseClassDependencyCollection	OpenModelAsTemplate
RoseClassDiagram	OpenScript
RoseClassDiagramCollection	OpenURL
RoseClassRelation	Save

Function **OpenModel**(*theModel As String*) As RoseModel
Member of RationalRose.RoseApplication

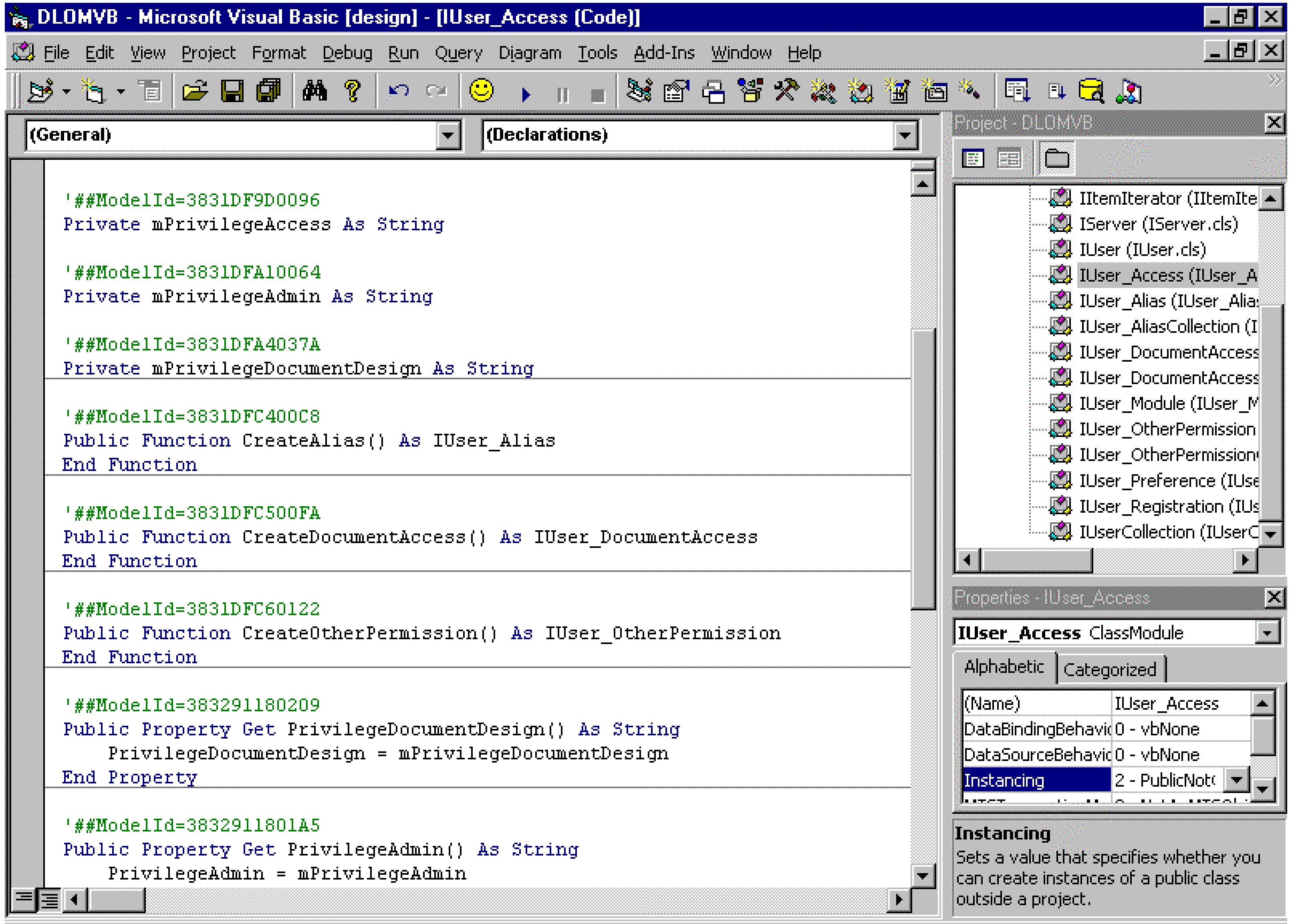
Rose (cont.) - retrieving the classes defined in a particular model


Project1 - Microsoft Visual Basic [run] - [Form1 (Code)]
File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help
Command1
Click
Immediate

```
Dim roseApplication As RationalRose.roseApplication
Dim roseModel As RationalRose.roseModel
Dim roseClassCollection As RationalRose.roseClassCollection
Dim roseClass As RationalRose.roseClass
Dim i As Integer

Set roseApplication = New RationalRose.roseApplication
Set roseModel = roseApplication.OpenModel("C:\siemens\objectmodelEPR\dlepr-objectmodel.mdl")
Set roseClassCollection = roseModel.GetAllClasses

For i = 1 To roseClassCollection.Count
    Set roseClass = roseClassCollection.GetAt(i)
    Debug.Print "Class: " & roseClass.Name
Next

Call roseApplication.Exit
```

```
Class: User_FamilyName
Class: IItemIterator
Class: _ViewItem
Class: _QueryItem
Class: _Relation
Class: IUser
Class: User_OtherPermission
Class: User_CardID
Class: User_Registration
Class: User_UserQuit
```

Rose (cont.) - Generating Visual Basic from a Model

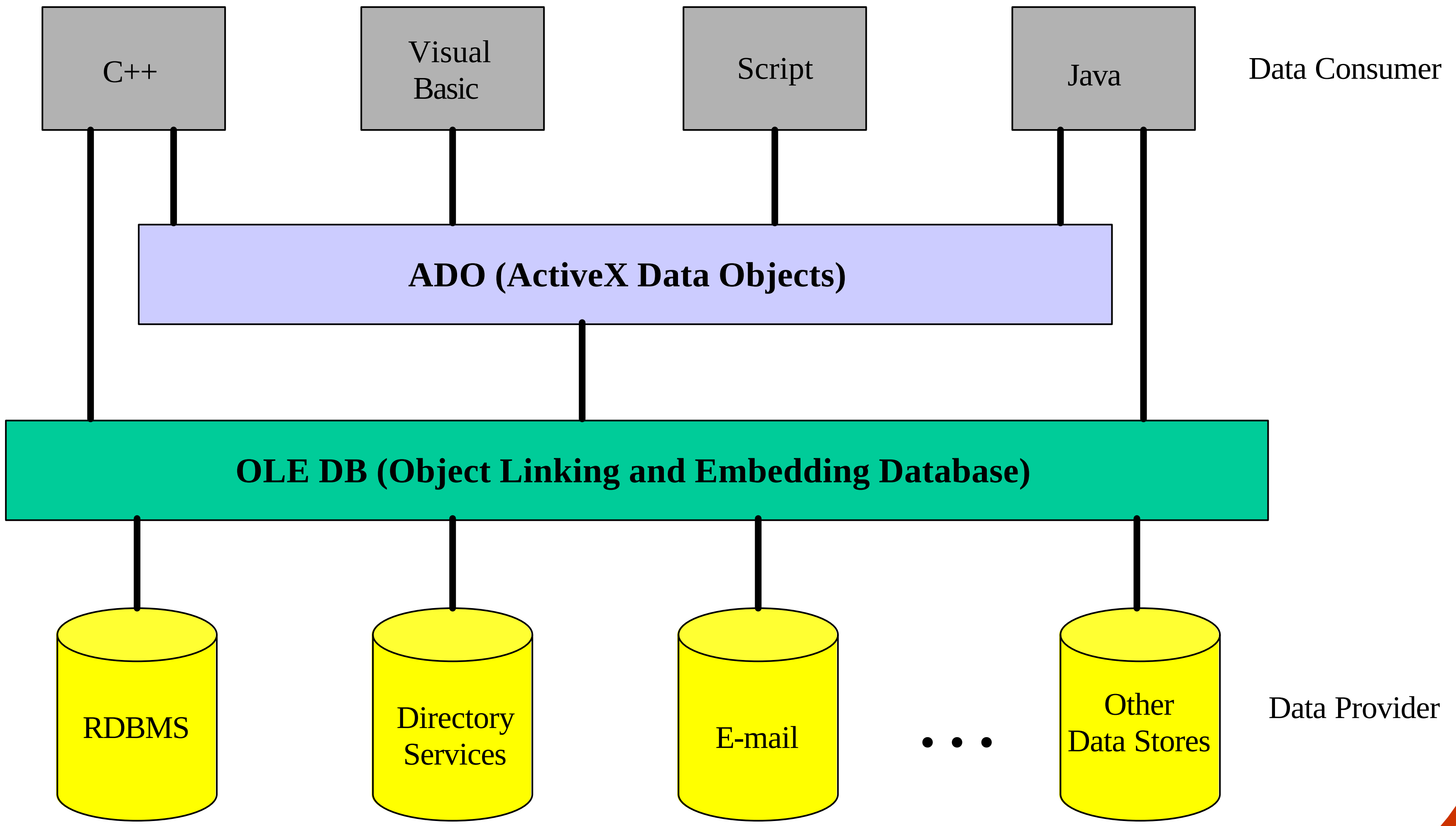


```
'##ModelId=3831DF9D0096
Private mPrivilegeAccess As String

'##ModelId=3831DFA10064
Private mPrivilegeAdmin As String

'##ModelId=3831DFA4037A
Private mPrivilegeDocumentDesign As String

'##ModelId=3831DFC400C8
Public Function CreateAlias() As IUser_Alias
End Function

'##ModelId=3831DFC500FA
Public Function CreateDocumentAccess() As IUser_DocumentAccess
End Function

'##ModelId=3831DFC60122
Public Function CreateOtherPermission() As IUser_OtherPermission
End Function

'##ModelId=383291180209
Public Property Get PrivilegeDocumentDesign() As String
    PrivilegeDocumentDesign = mPrivilegeDocumentDesign
End Property

'##ModelId=3832911801A5
Public Property Get PrivilegeAdmin() As String
    PrivilegeAdmin = mPrivilegeAdmin
```

(Name)	IUser_Access
DataBindingBehavi	0 - vbNone
DataSourceBehavi	0 - vbNone
Instancing	2 - PublicNot

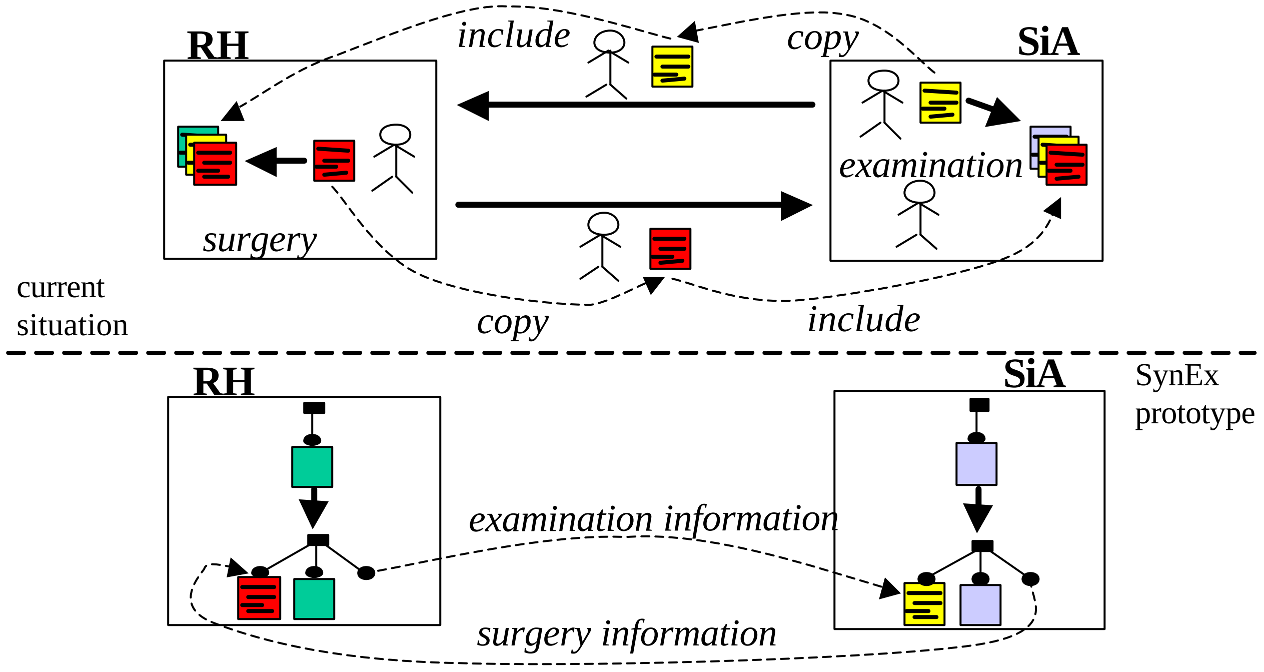
Instancing
Sets a value that specifies whether you can create instances of a public class outside a project.

Universal Data Access (UDA) with OLE DB and ADO

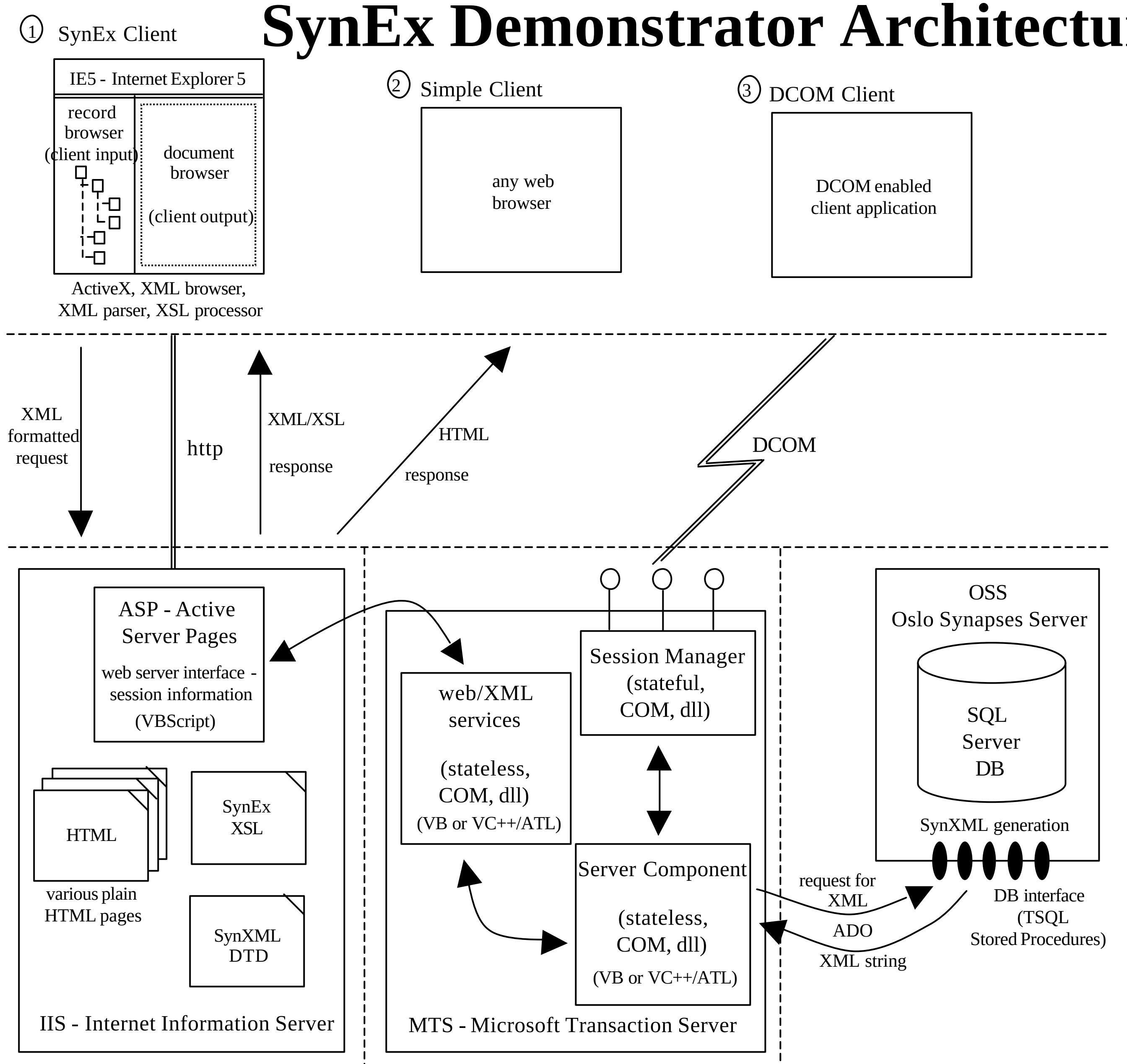


SynEx - Synergy on the Extranet

Shared, Federated Electronic Healthcare Records



SynEx Demonstrator Architecture



SOAP - Simple Object Access Protocol

XML formatted Server Requests via http

Server request:

http:\\www.nr.no\synexdemo\oss.asp?<OSSrequest>
<RecordInfo RecordID="12082373463" Retrieval="all"/>
</OSSrequest>

The parameter **"<OSSrequest>…</OSSrequest>"** is received by the Active Server Page **"oss.asp"**, at the specified address, for further server-side processing.

Benefit:

Enable access to server-side business objects via http - less problems with Firewalls

PS: SOAP (Simple Object Access Protocol) uses a different XML format.

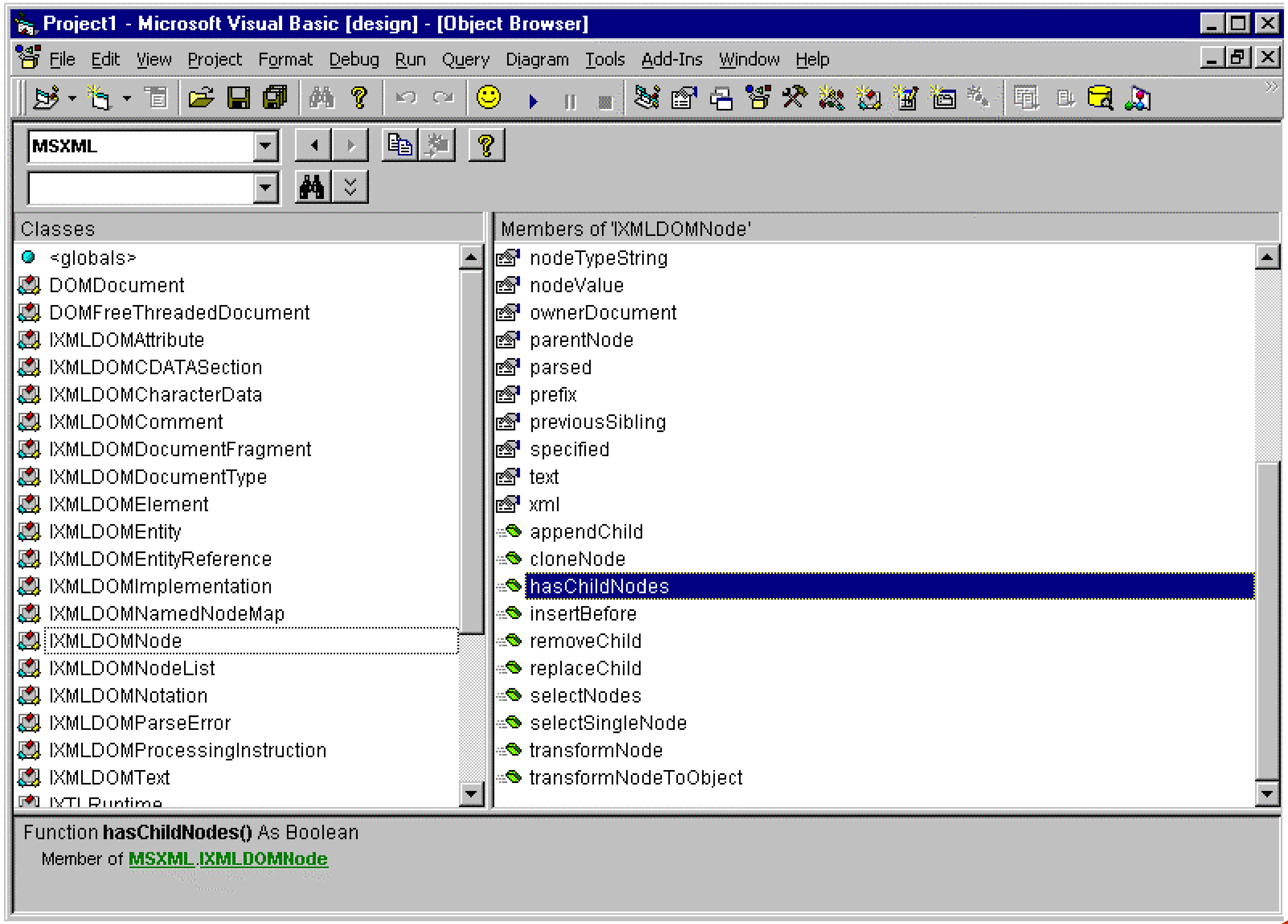
Active Server Pages (ASP)

- Avoid the use of scripting languages (e.g. VBscript) except as ”glue” between COM components.

  VB, VC++ or J++ offers better development environments

```
<%@ Language=VBScript %>
<%
'--| The ASP script for retrieval of record and document XML from the Oslo Synapses Server.
'--|
'--| The script creates a COSSASPServer object and forwards its parameter client request to this
'--| object's PerformSynXMLRequest function. The COSSASPServer object will handle the request
'--| and forward the requested XML information, or an error message, back to the client via the
'--| Response object of this ASP script.

On Error Resume Next

Set objServer = Server.CreateObject("OSSSynExDemo.COSSASPServer")
objServer.PerformSynXMLRequest(Request.ServerVariables("QUERY_STRING"))

If Err.Number <> 0 Then
   Response.Write(”…error message to client - e.g. XML formatted...")
   Err.Clear
End If
%>
```

Document Object Model (DOM) of the MS XML Parser


Project1 - Microsoft Visual Basic [design] - [Object Browser]
File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help
MSXML

Classes

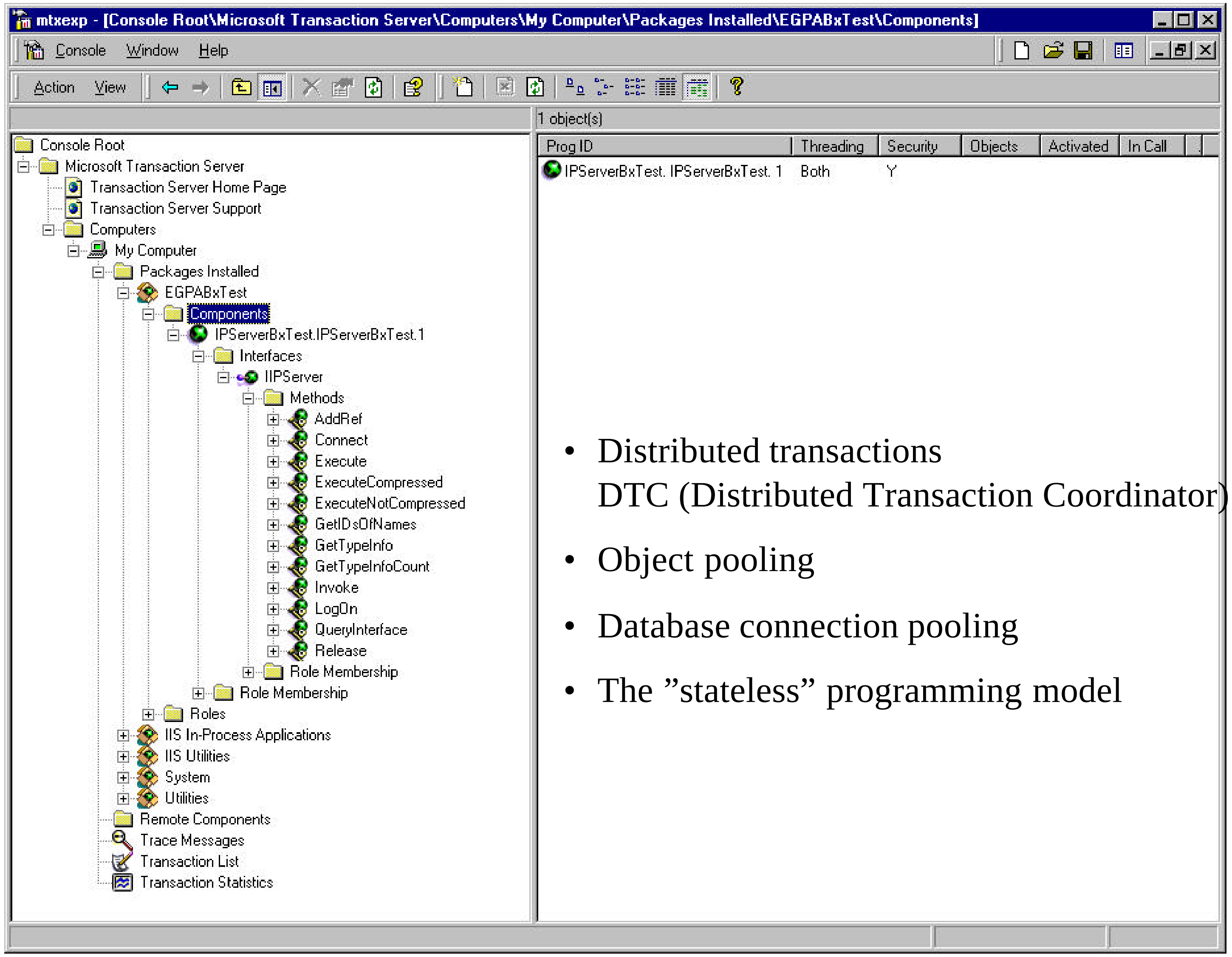
- <globals>
- DOMDocument
- DOMFreeThreadedDocument
- IXMLDOMAttribute
- IXMLDOMCDATASection
- IXMLDOMCharacterData
- IXMLDOMComment
- IXMLDOMDocumentFragment
- IXMLDOMDocumentType
- IXMLDOMElement
- IXMLDOMEntity
- IXMLDOMEntityReference
- IXMLDOMImplementation
- IXMLDOMNamedNodeMap
- IXMLDOMNode
- IXMLDOMNodeList
- IXMLDOMNotation
- IXMLDOMParseError
- IXMLDOMProcessingInstruction
- IXMLDOMText
- IXTLRuntime

Members of 'IXMLDOMNode'

- nodeTypeString
- nodeValue
- ownerDocument
- parentNode
- parsed
- prefix
- previousSibling
- specified
- text
- xml
- appendChild
- cloneNode
- hasChildNodes
- insertBefore
- removeChild
- replaceChild
- selectNodes
- selectSingleNode
- transformNode
- transformNodeToObject

Function **hasChildNodes()** As Boolean
Member of MSXML.IXMLDOMNode

MTS - Microsoft Transaction Server



mtxexp - [Console Root\Microsoft Transaction Server\Computers\My Computer\Packages Installed\EGPABxTest\Components]

- Console Root
  - Microsoft Transaction Server
    - Transaction Server Home Page
    - Transaction Server Support
    - Computers
      - My Computer
        - Packages Installed
          - EGPABxTest
            - Components
              - IPServerBxTest.IPServerBxTest.1
                - Interfaces
                  - IIPServer
                    - Methods
                      - AddRef
                      - Connect
                      - Execute
                      - ExecuteCompressed
                      - ExecuteNotCompressed
                      - GetIDsOfNames
                      - GetTypeInfo
                      - GetTypeInfoCount
                      - Invoke
                      - LogOn
                      - QueryInterface
                      - Release
                    - Role Membership
                - Role Membership
            - Roles
          - IIS In-Process Applications
          - IIS Utilities
          - System
          - Utilities
        - Remote Components
        - Trace Messages
        - Transaction List
        - Transaction Statistics

Prog ID	Threading	Security	Objects	Activated	In Call
IPServerBxTest. IPServerBxTest. 1	Both	Y			

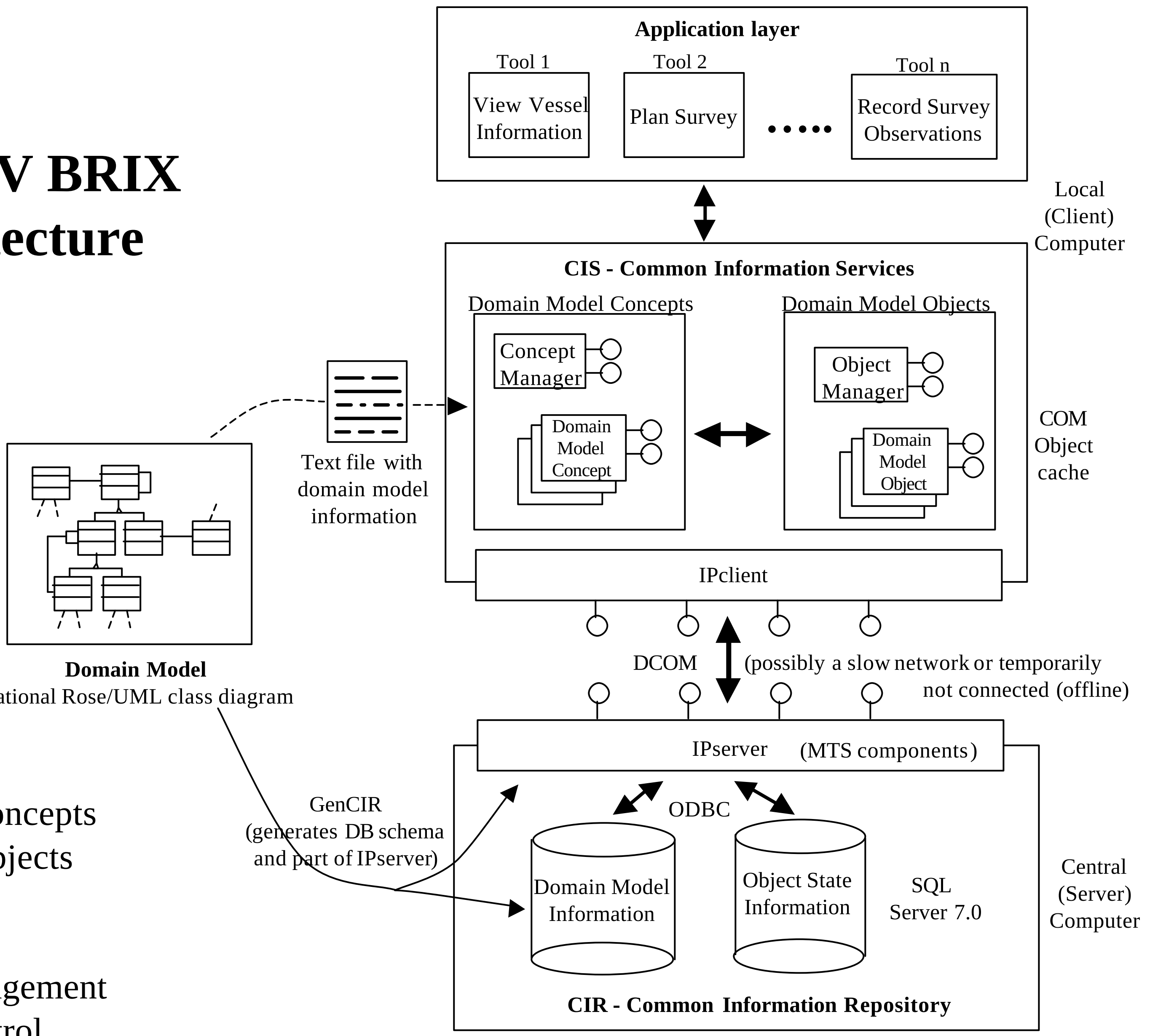
- Distributed transactions
  DTC (Distributed Transaction Coordinator)
- Object pooling
- Database connection pooling
- The ”stateless” programming model

Microsoft Message Queue Server (MSMQ)

- Message - “piece” of information sent between two applications
- Messages can be formatted in e.g. XML
- MSMQ allow different applications to communicate with each other using “store-and-forward”
- MSMQ is similar to E-mail servers -
  more mechanisms for assuring the reception of messages sent

The DNV BRIX Architecture

- Domain model
- CIR - Relational database
- IPserver
- CIR generation
- CIS object cache
- Domain model concepts
- Domain model objects
- IPclient
- CIS generation
- Transaction management
- Concurrency control



Application layer
Tool 1
View Vessel Information
Tool 2
Plan Survey
.....
Tool n
Record Survey Observations
Local (Client) Computer
CIS - Common Information Services
Domain Model Concepts
Concept Manager
Domain Model Concept
Domain Model Objects
Object Manager
Domain Model Object
COM Object cache
Text file with domain model information
Domain Model
Rational Rose/UML class diagram
IPclient
DCOM
(possibly a slow network or temporarily not connected (offline)
IPserver (MTS components)
GenCIR (generates DB schema and part of IPserver)
ODBC
Domain Model Information
Object State Information
SQL Server 7.0
Central (Server) Computer
CIR - Common Information Repository

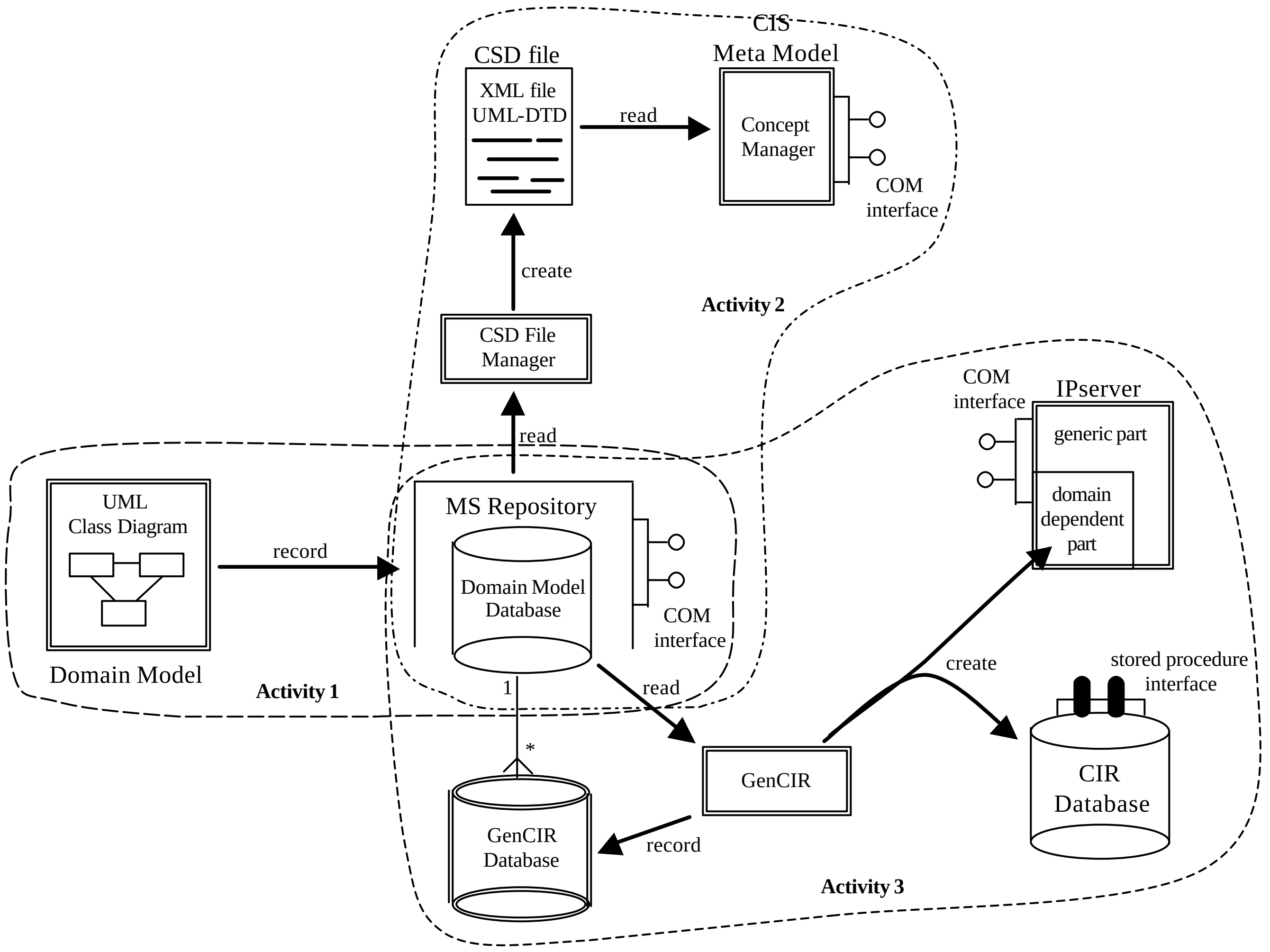
Motivation for BRIX

- Uniform access to a shared domain model
- Application developers only need to know the domain model to know how to operate on persistent objects instantiated from classes defined in the model
- Genericity is achieved without a generic domain model
- Services offered and transactions executed cannot be entirely predefined
- Change control - being able to handle changes in a flexible manner Different parts of the architecture provides relatively good encapsulation such that major changes or revisions do not affect the entire architecture
- Model independent CIS - easier to change the server centrally than to change all the distributed clients

Problems and Challenges for BRIX

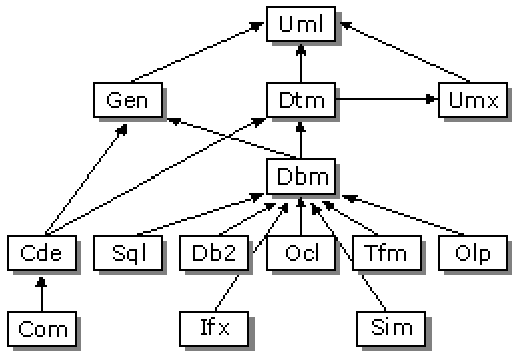
- Performance
- Transaction handling and Concurrency control
- Caching strategies
- Enforcing general business rules
- Managing the software development process
- Roles and role modeling
  The generic caching services and the transaction and concurrency control mechanisms does not utilize knowledge on how the objects involved will be used and operated on by the application tools.
  Instead of creating a single overall domain model, create several smaller sub-domain models that each model a more specific and narrow domain; e.g. a single task or activity.

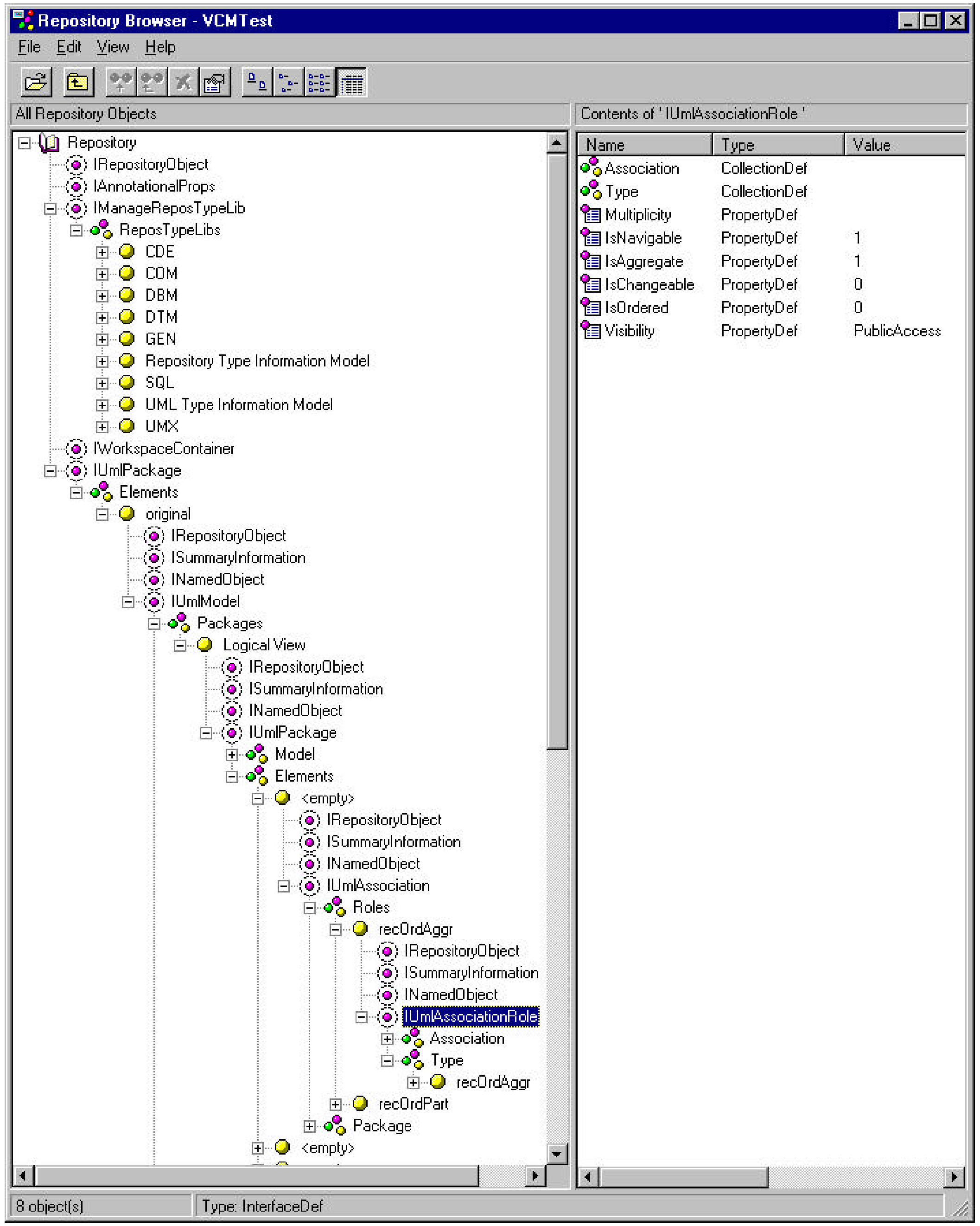
Microsoft Repository in the BRIX Architecture



Microsoft Repository

- **Meta-information management**
- **Object Information Model**
- **Extendable Subject Areas**
- **COM/Automation access**





All Repository Objects

```
Repository
  IRepositoryObject
  IAnnotationalProps
  IManageReposTypeLib
    ReposTypeLibs
      CDE
      COM
      DBM
      DTM
      GEN
      Repository Type Information Model
      SQL
      UML Type Information Model
      UMX
  IWorkspaceContainer
  IUmlPackage
    Elements
      original
        IRepositoryObject
        ISummaryInformation
        INamedObject
        IUmlModel
          Packages
            Logical View
              IRepositoryObject
              ISummaryInformation
              INamedObject
              IUmlPackage
                Model
                Elements
                  <empty>
                    IRepositoryObject
                    ISummaryInformation
                    INamedObject
                    IUmlAssociation
                      Roles
                        recOrdAggr
                          IRepositoryObject
                          ISummaryInformation
                          INamedObject
                          IUmlAssociationRole
                            Association
                            Type
                              recOrdAggr
                        recOrdPart
                      Package
                  <empty>
```

Contents of ' IUmlAssociationRole '

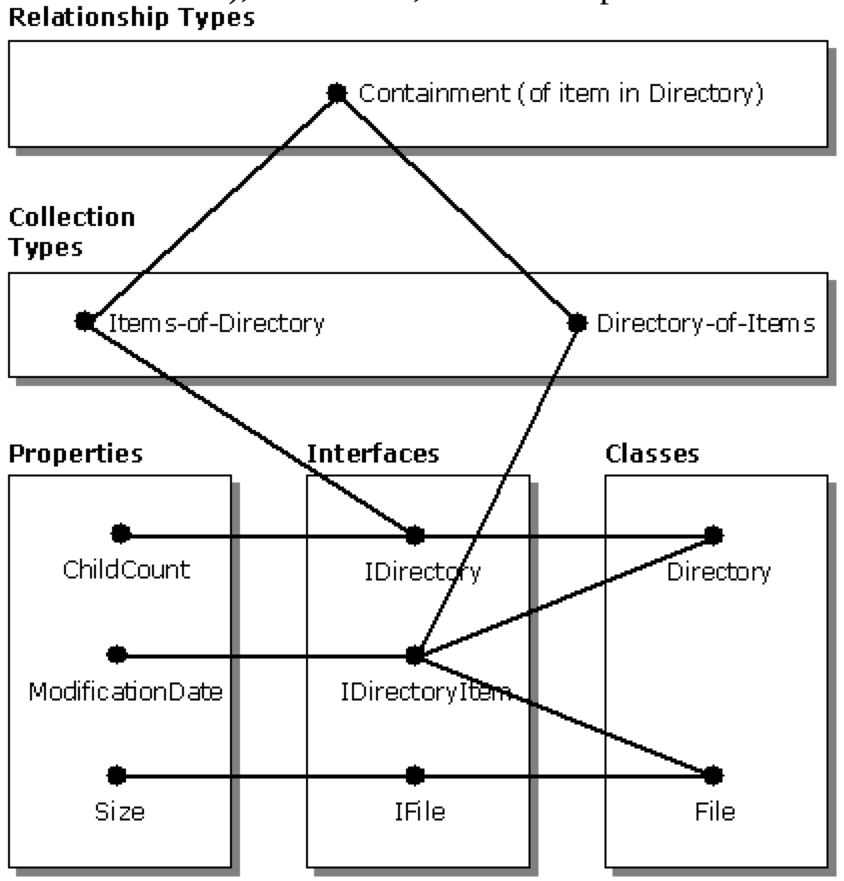
Name	Type	Value
Association	CollectionDef	
Type	CollectionDef	
Multiplicity	PropertyDef	
IsNavigable	PropertyDef	1
IsAggregate	PropertyDef	1
IsChangeable	PropertyDef	0
IsOrdered	PropertyDef	0
Visibility	PropertyDef	PublicAccess

8 object(s) | Type: InterfaceDef

Object Information Models of Microsoft Repository

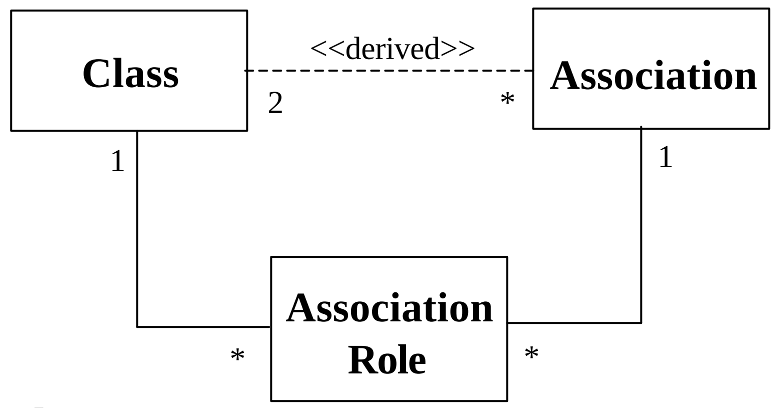
RTIM - Repository Type Information Model

- A domain independent information model
- Made to record and retrievemeta-information on a variety of domains (e.g. UML, DB Schemas, Components, Datatypes, and more)
- Basic concepts: Class, Interface, Property (attribute and method), Collection, Relationship



Repository UML Information Model

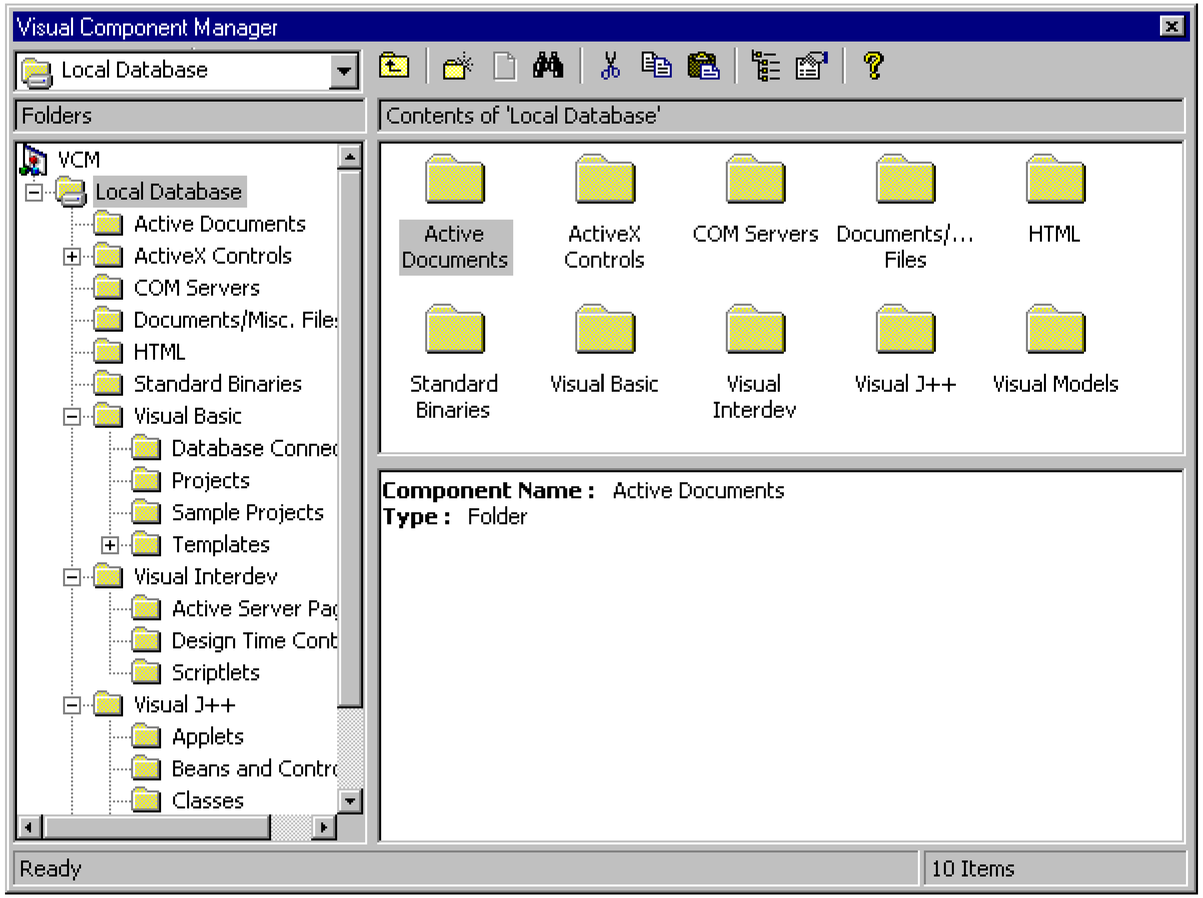
- A domain dependent information model
- Made to record and retrieve meta-information on UML models (e.g. fromRational Rose models)
- Accompanies COM/Automation interfaces
- Implemented by RTIM
- Currently “too normalized” - should allow for “redundancy”; e.g. Class-Association-Role relationships



```
IClass ==
    GetRoles()->IAssociationRoleColl
  + [GetAssociations()->IAssociationColl]
IAssociation ==
    GetRoles()->IAssociationRoleColl
  + [GetClasses()->IClassColl]
IAssociationRole ==
    GetAssociation()->IAssociation
    GetClass()->IClass
```

Visual Component Manager (VCM)

A repository for organizing and storing information on components, models, projects, and more, to make them readily available to the development organization.



Component Name : Active Documents
Type : Folder

NR

What about Object-Oriented Databases?

- Main benefit

  Avoids mismatch between relational data and object-oriented applications;
  e.g. inheritance relationships, recursive structures, ….

- Do they scale well, do they perform well - may be - but many in industry considers it an added risk to rely on this for large enterprise information systems

- Developers will be happy with them - but will this reduce development cost enough to outweight the “risk” (real or perceived)?

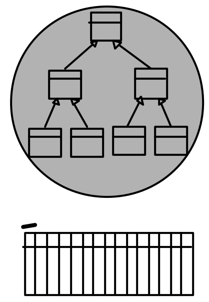
- What about customers - do they benefit from it?

- Main problem

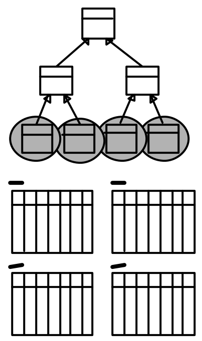
  No “killer application” - there seems to be no undisputable need for it

Object-Oriented to Relational

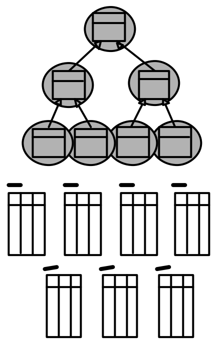
a) Single table



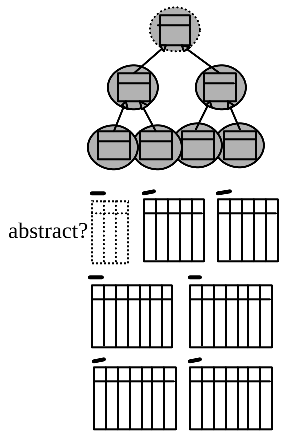
b) Leaf tables only



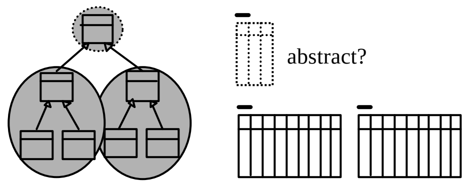
c) One partial table per class



d) One full table per class



e) Logical split in the inheritance hierarchy



“Componentifying” FS?