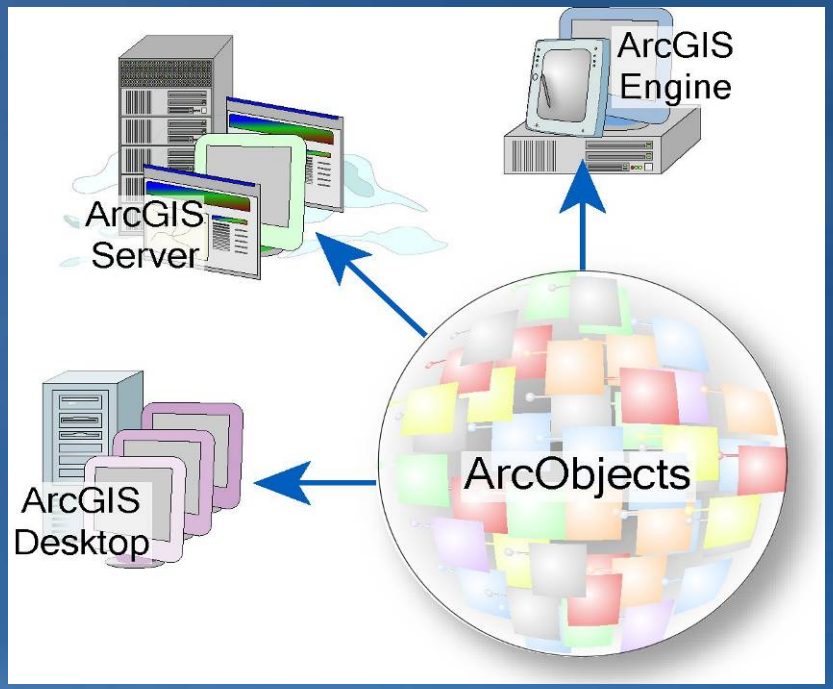
Getting to Know ArcObjects



Presenters

- Rob Burke
  - Instructor
  - ESRI-Redlands
- Don Kemlage
  - .NET SDK Team
  - Product Engineer
  - ESRI-Redlands

Intended audience

- Anyone wanting to learn about ArcObjects
- We start with the ArcObjects basics
- For programmers and non-programmers
- Maybe you know
  - Some ArcGIS
  - Some programming
- Not language specific

Lecture Path

What are ArcObjects

GIS Parts

Diagrams

Interfaces & COM

Implement ESRI Interfaces

Extend the Applications

Extend the Geodatabase

Engine

Server

Questions

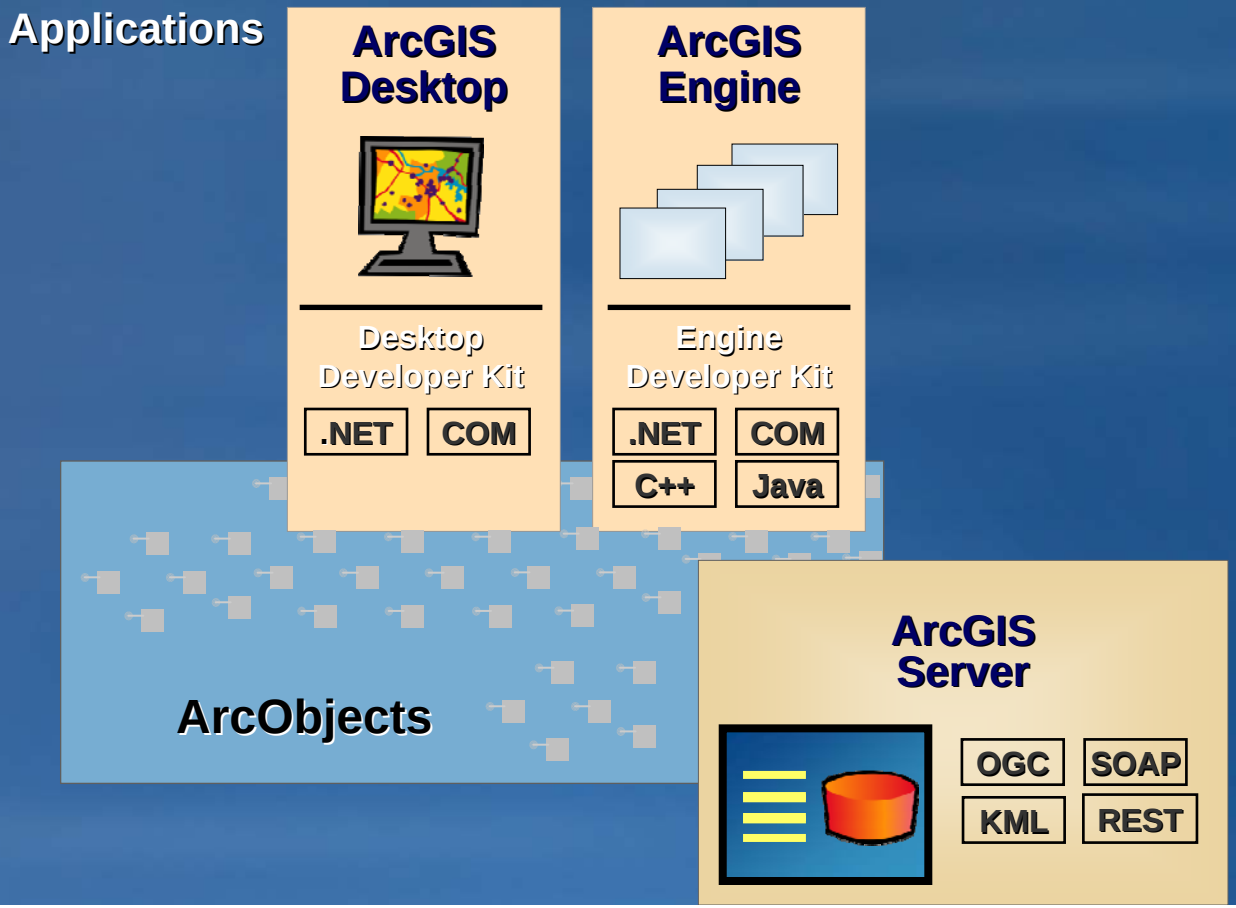
Wrap-up

How do you get ArcObjects

- ArcObjects is not a product, by itself
- You can't buy just ArcObjects
  - You buy an ArcGIS Desktop product
  - You buy ArcGIS Engine
  - You buy ArcGIS Server
- ArcObjects-related files, folders, Dlls, libraries, and assemblies are installed when you install one of the three products

ArcGIS: The diagram

- ESRI development team creates the ArcObjects
- ESRI uses ArcObjects to build ArcMap and other desktop apps
- You use ArcObjects to build Engine apps
- You use ArcObjects to enhance Server apps



Parts for GIS

- Building blocks for ArcGIS software products and yours
- The most basic GIS parts are programmable objects

Map

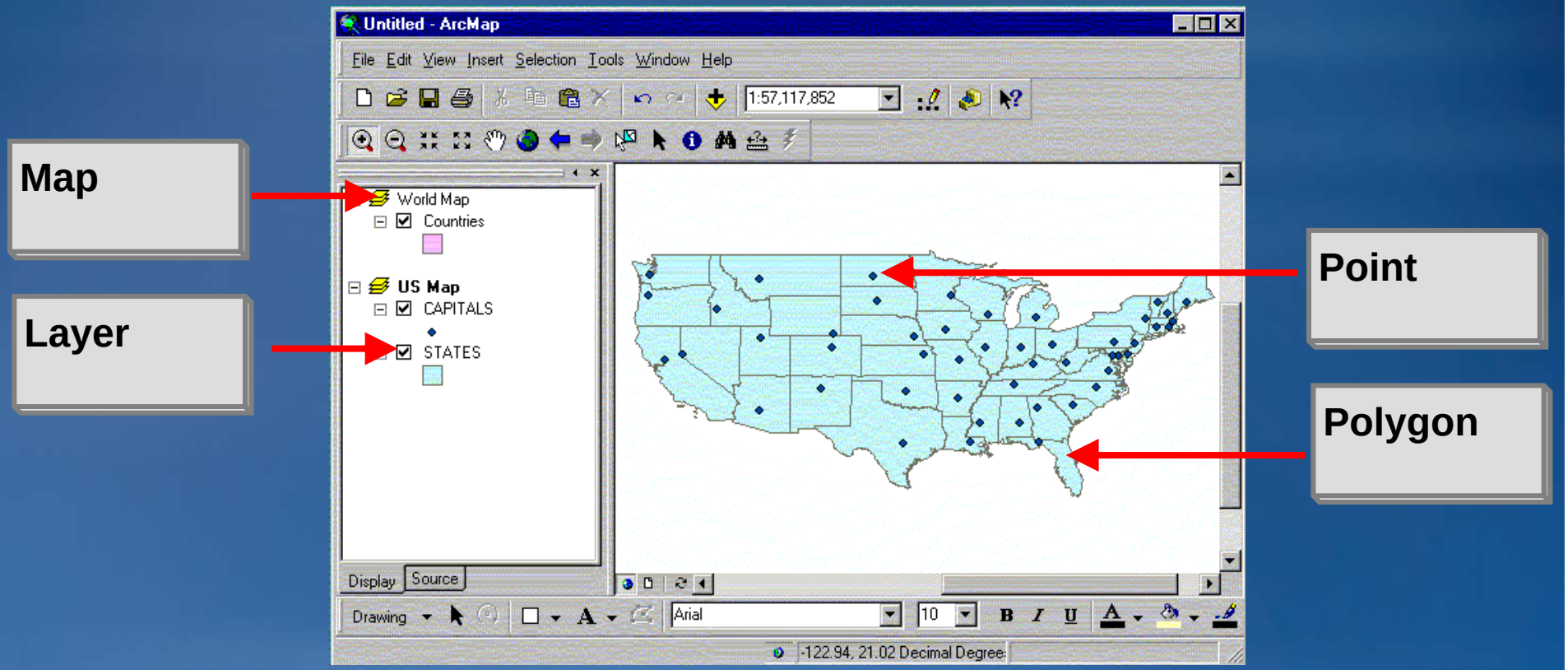
Layer

Point

Polygon

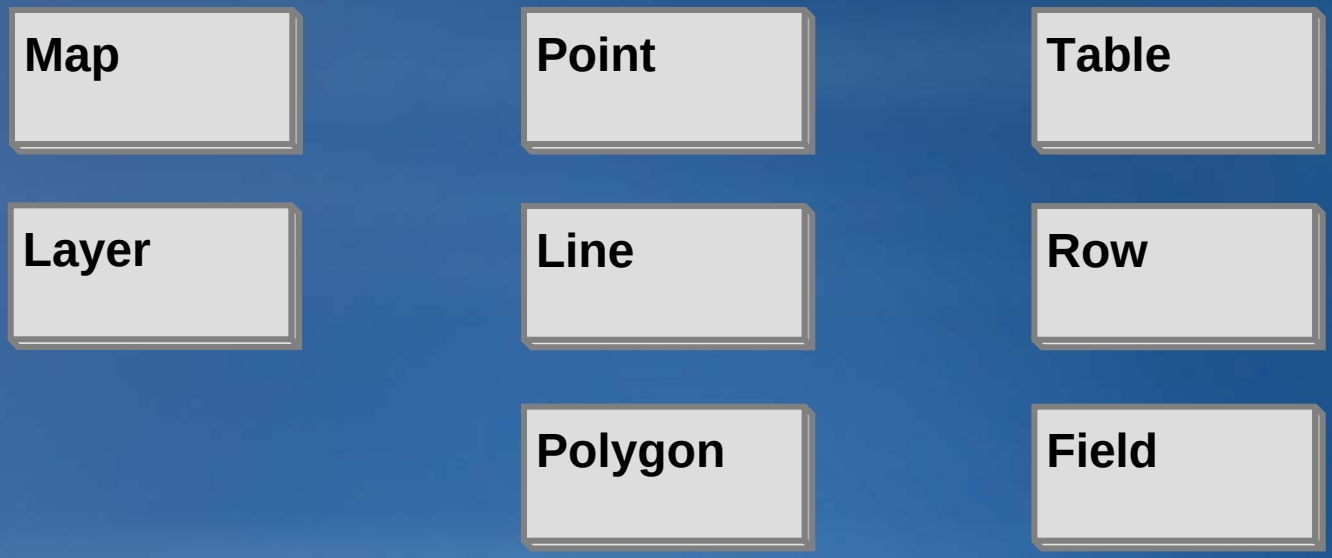
Parts for GIS

- ArcMap and ArcCatalog are built with ArcObjects
- Their most basic parts are programmable GIS objects
- Users and programmers interact with the same objects
- Everyone uses the same parts
  - ESRI developers, users, you, and I



GIS parts: Classes, blueprints, templates, or components

- Technically the ‘parts’ are called classes (or components)
  - Over 4,000 ArcObjects classes
  - Over 21,000 properties and methods
  - Grouped into over 7 0 logical libraries
    - (not including ArcGIS Server)
- Each class corresponds to a basic GIS part



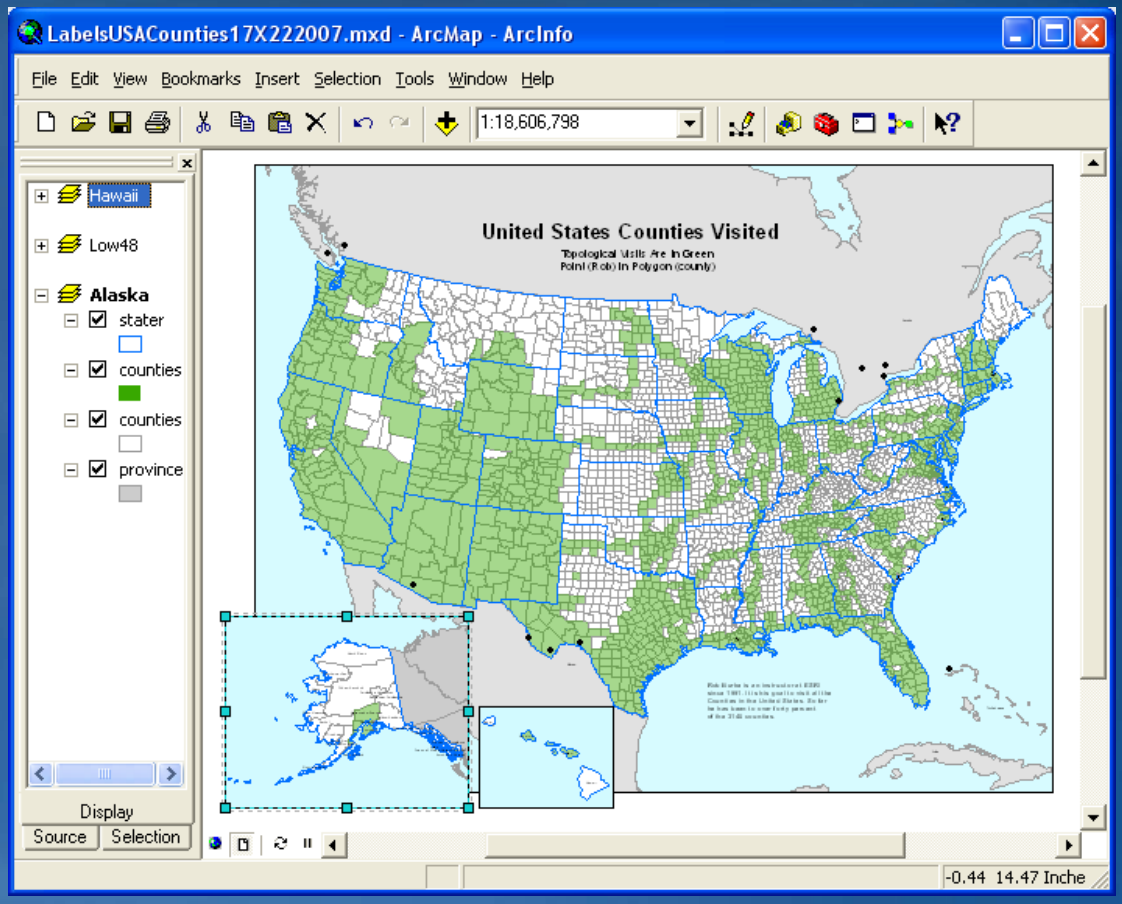
Map class compared to a map object

- There is one Map class
- You make many map objects: Hawaii, Low48, and Alaska
- A class is code behind an object's properties and methods
- Objects live in memory and take on your settings

The Map class

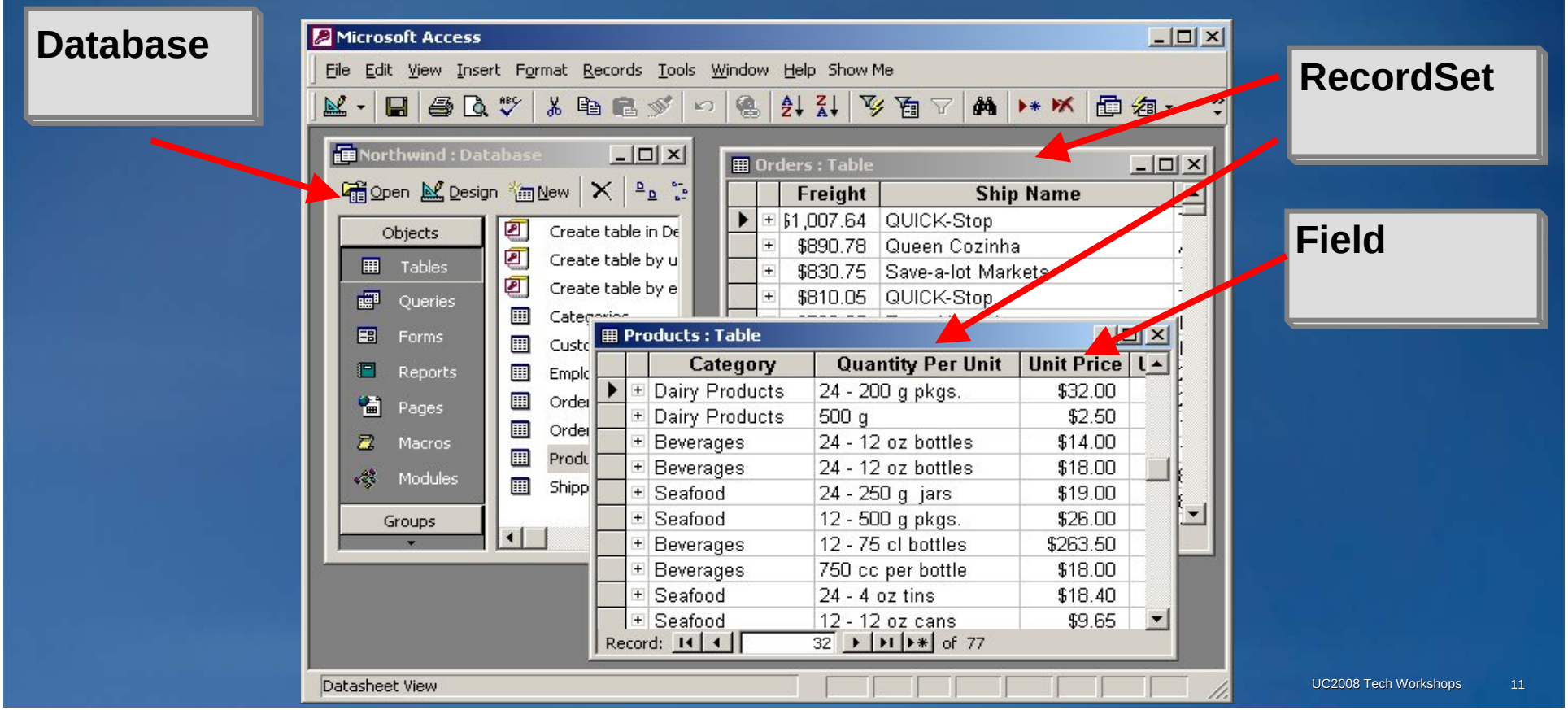
Map

Three map objects



Microsoft Access data objects

- Access built from a about a hundred objects
- Users and programmers use same objects
- Same classes and objects idea

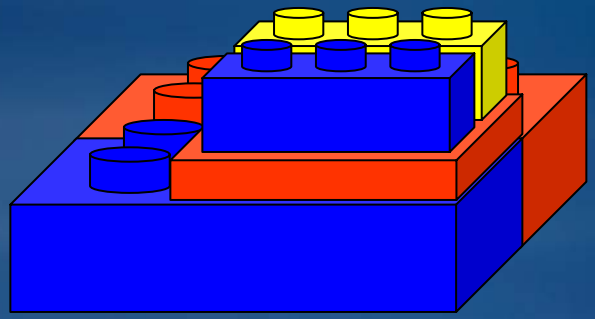


Use COM compatible languages

- COM – component object model
  - A style of programming classes in any language for use by any other com compatible language. ArcObjects are all C++ COM classes.
- VBA is built into the ArcGIS desktop applications
  - Visual Basic for Applications
  - VBA code is stored in map document files (mxd files)
- Buy a language and IDE (Development environment)
  - Visual Basic or C# (Visual Studio)
  - C++
  - Java
- Use a free open source language and IDE
  - Like Python

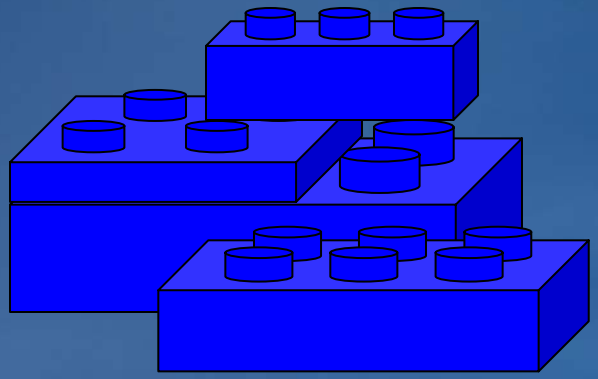
Class libraries

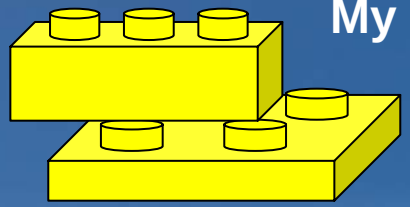
- COM classes can be mixed and matched
- Make your own library of COM classes
- Language independent



My Application

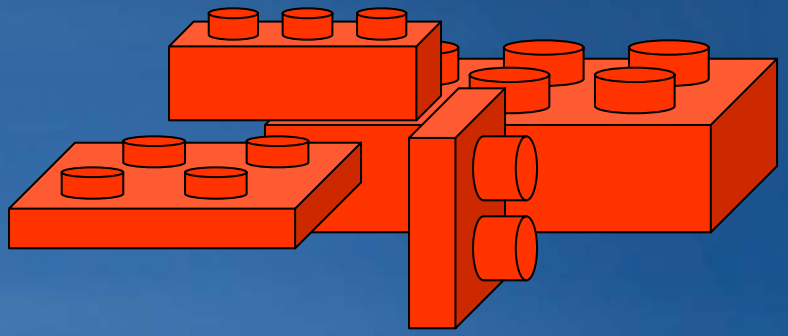
ArcObjects classes



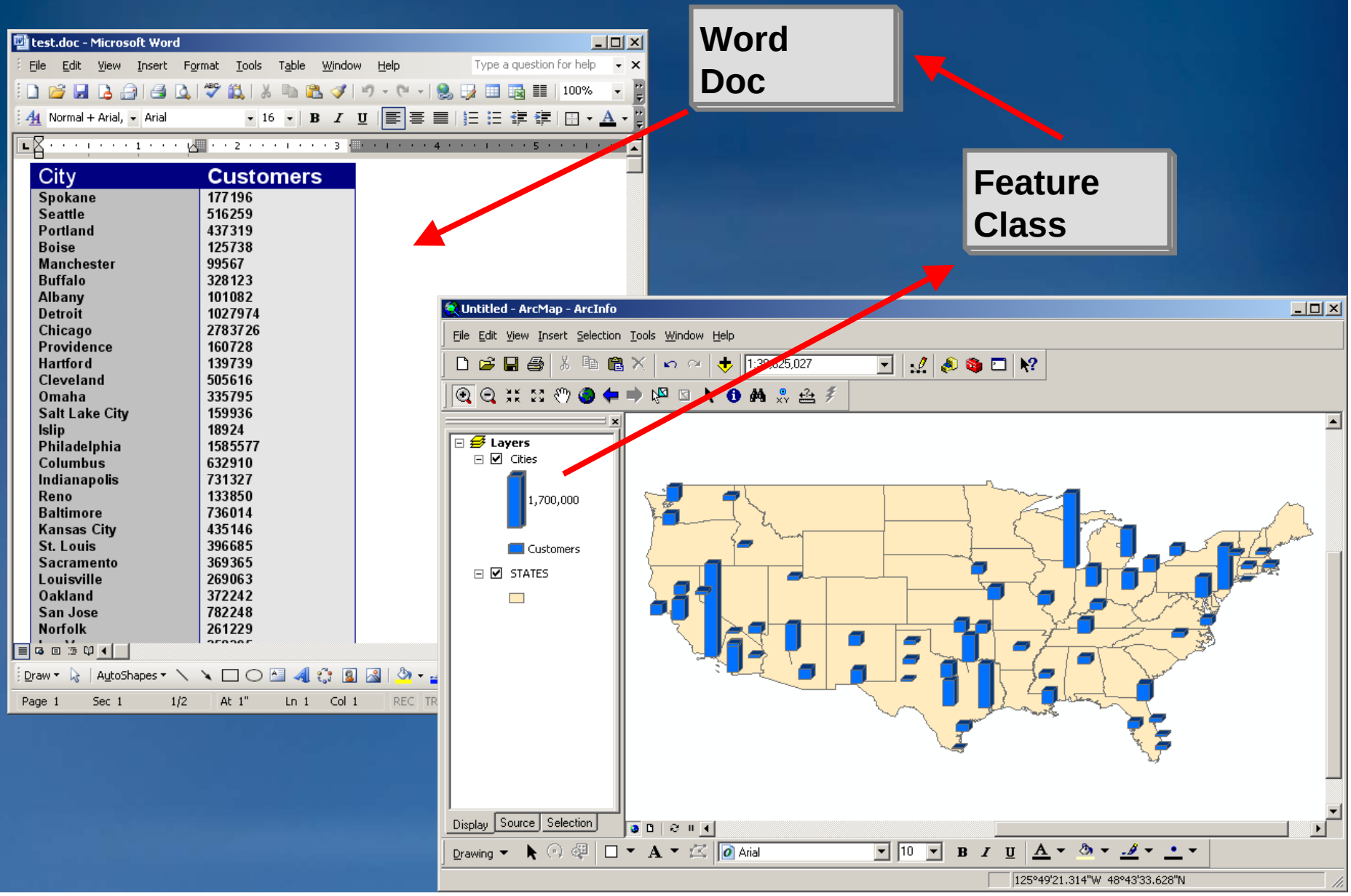


My classes

Word classes

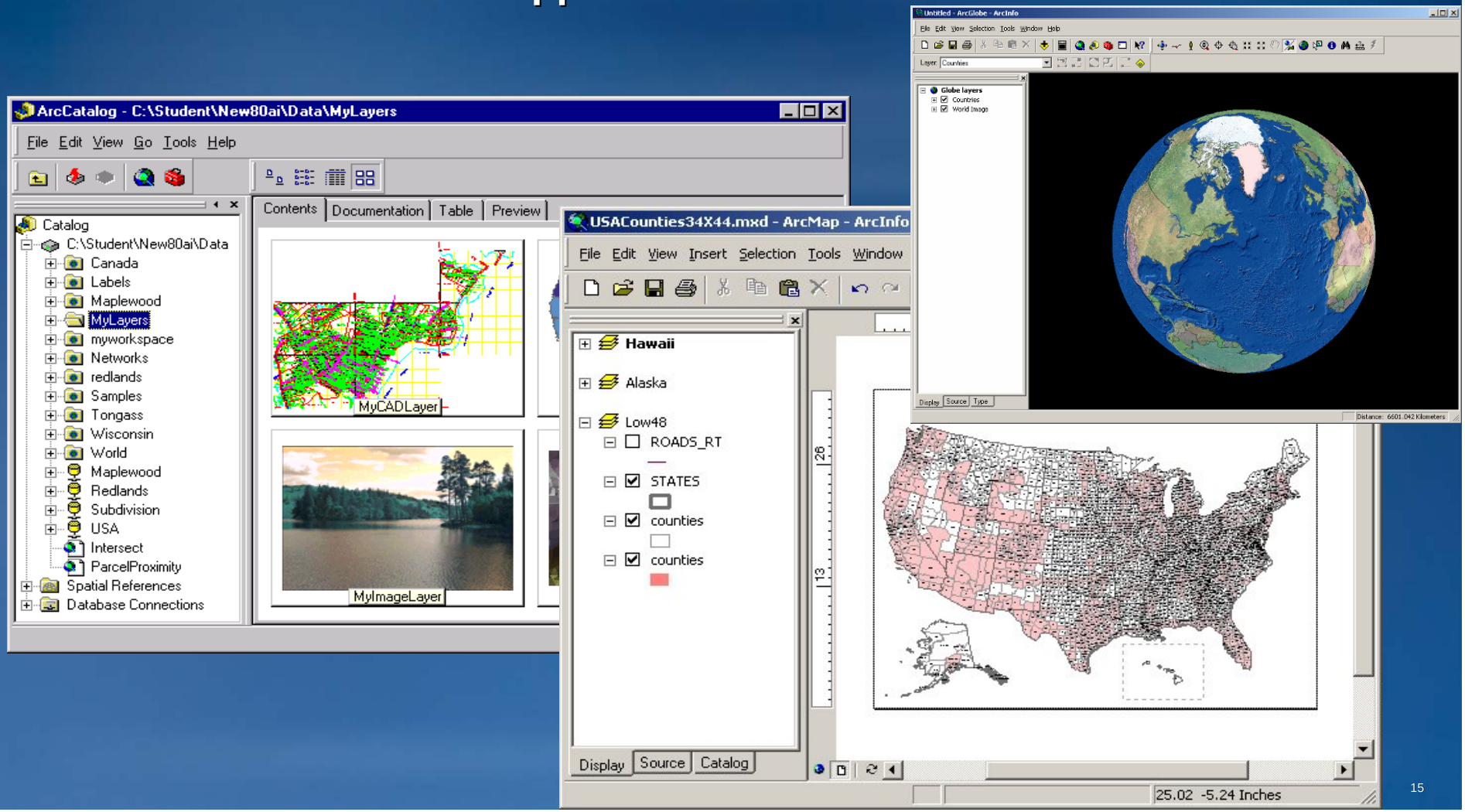


Mixing libraries: shapefile to word doc



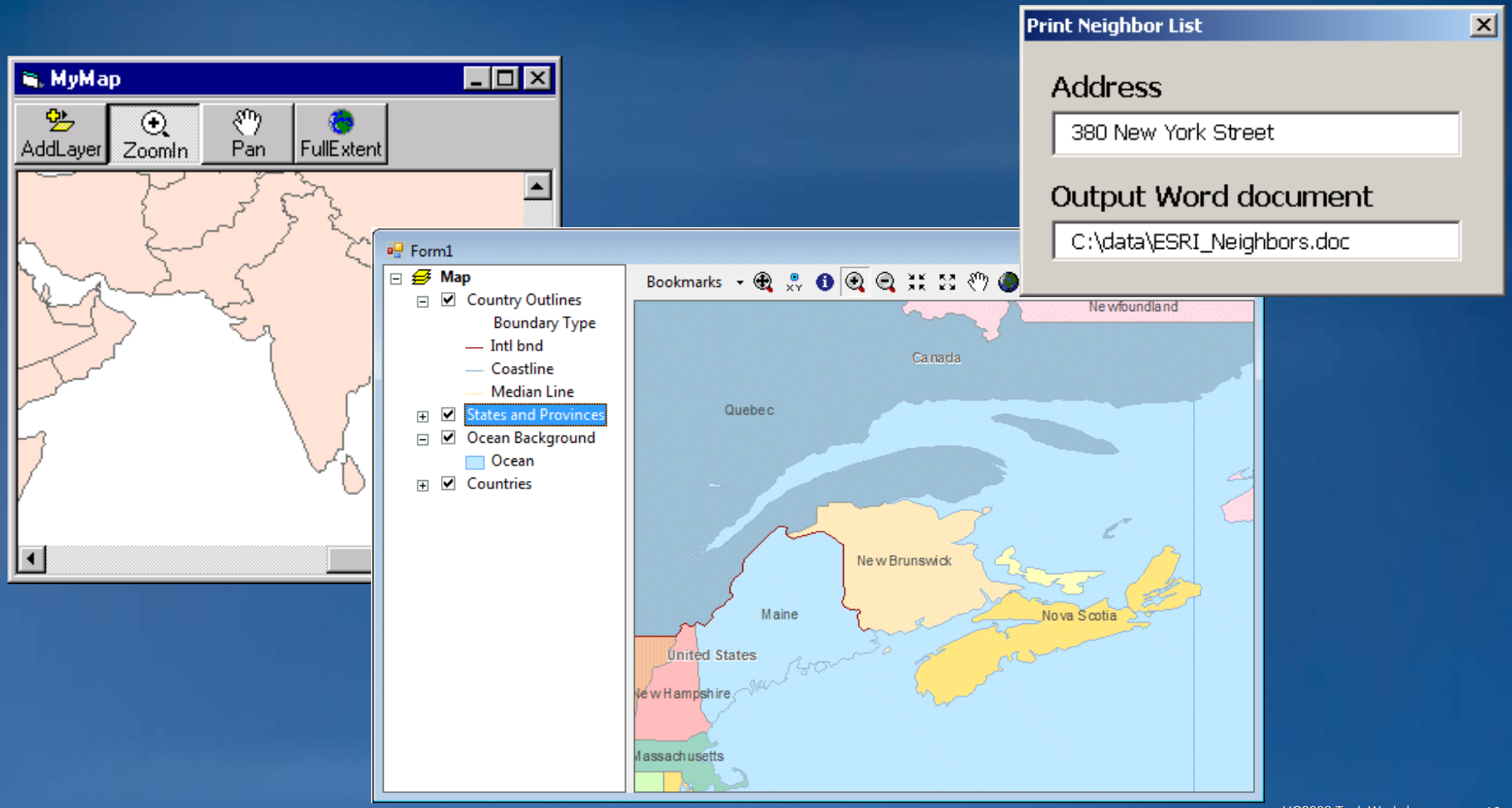
ArcObjects are building blocks: Desktop

- ArcGIS Desktop applications are created with ArcObjects
- ArcMap, ArcCatalog, ArcScene, ArcGlobe…
- You can extend these applications



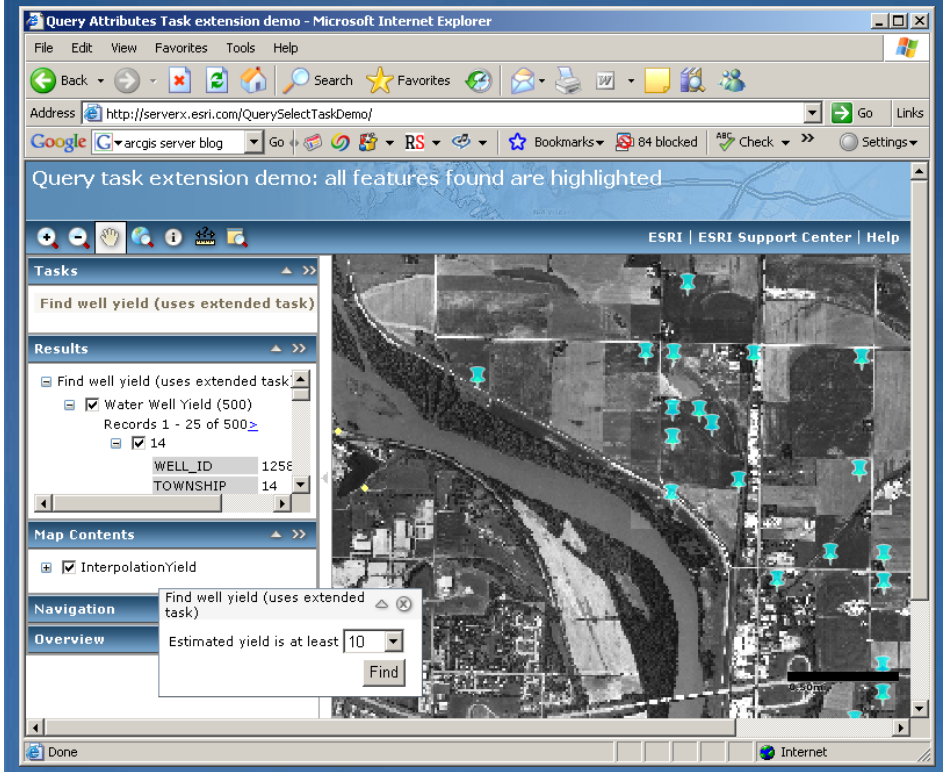
ArcObjects are building blocks: Engine

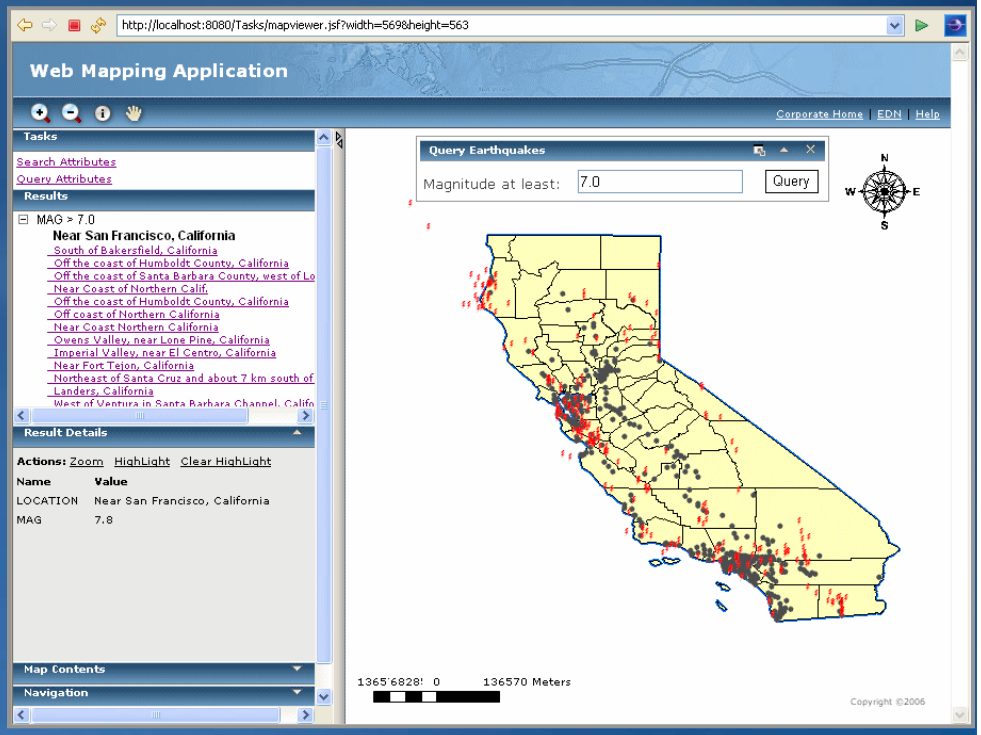
- ArcGIS Engine applications are created with ArcObjects
- You create stand alone GIS applications
- Use ArcObjects in non-GIS-centric applications



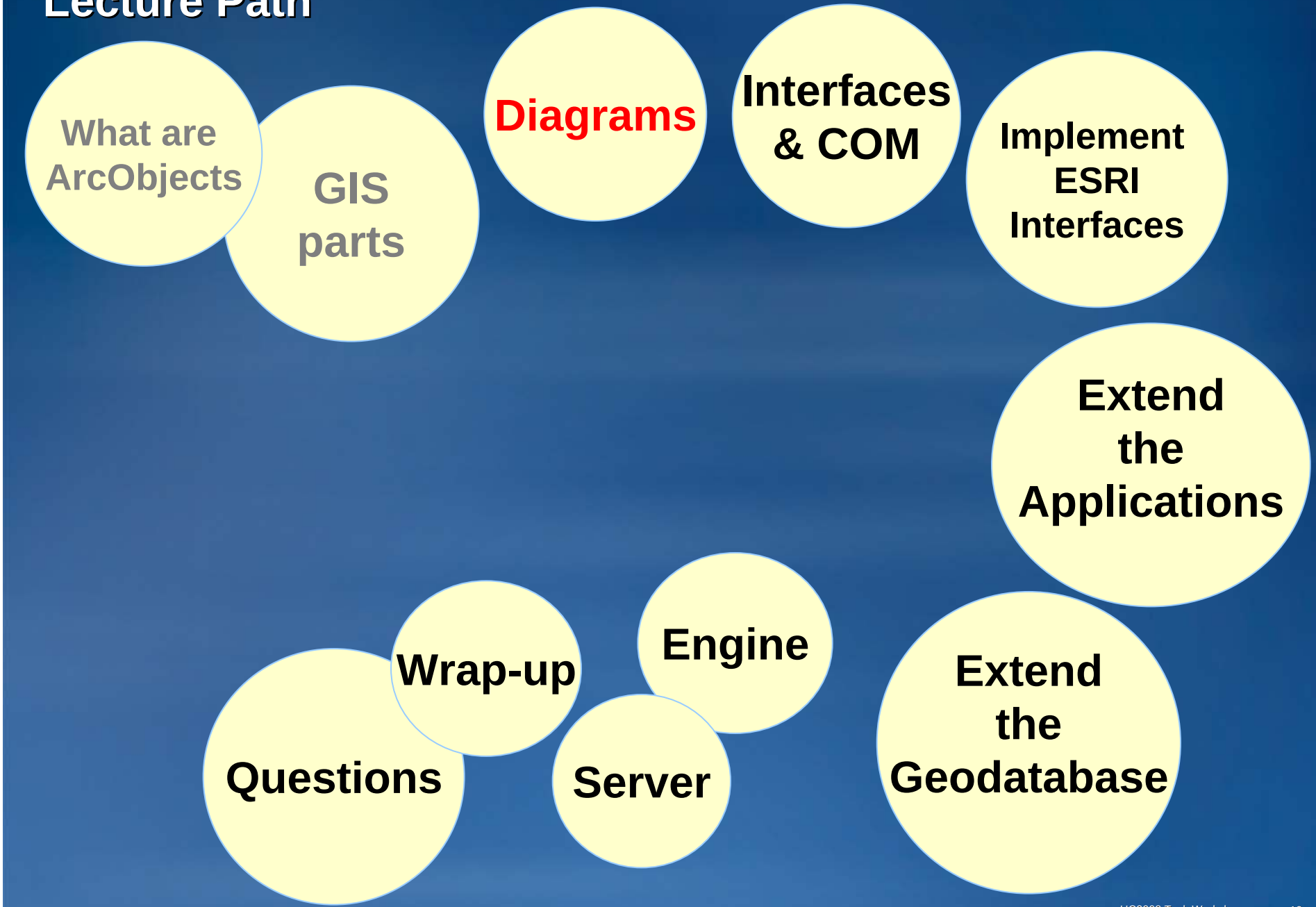
ArcObjects are building blocks: Server

- ArcGIS Server Web mapping applications are created with ArcObjects and special Web ADF GIS objects
- Author, publish, or consume
  - GIS data, functionality, and geoprocessing
  - Query, edit, address match, make custom, and more



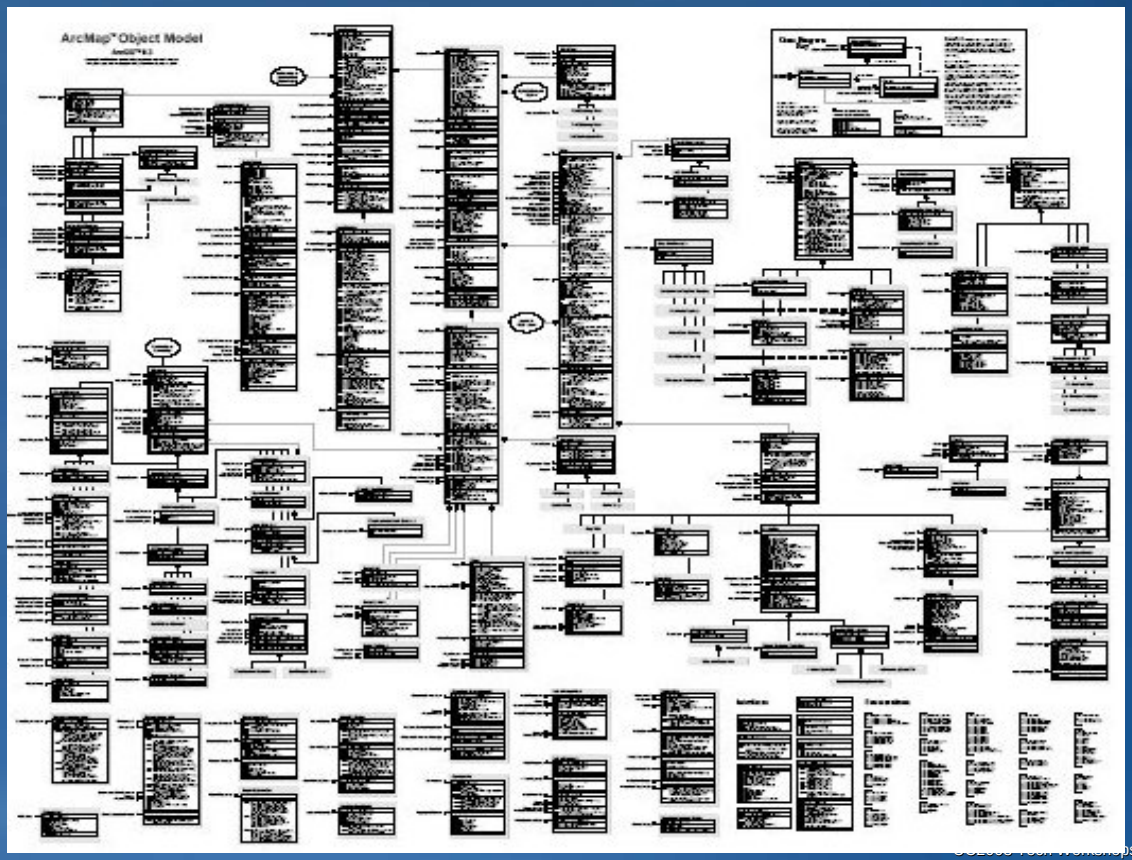


Lecture Path



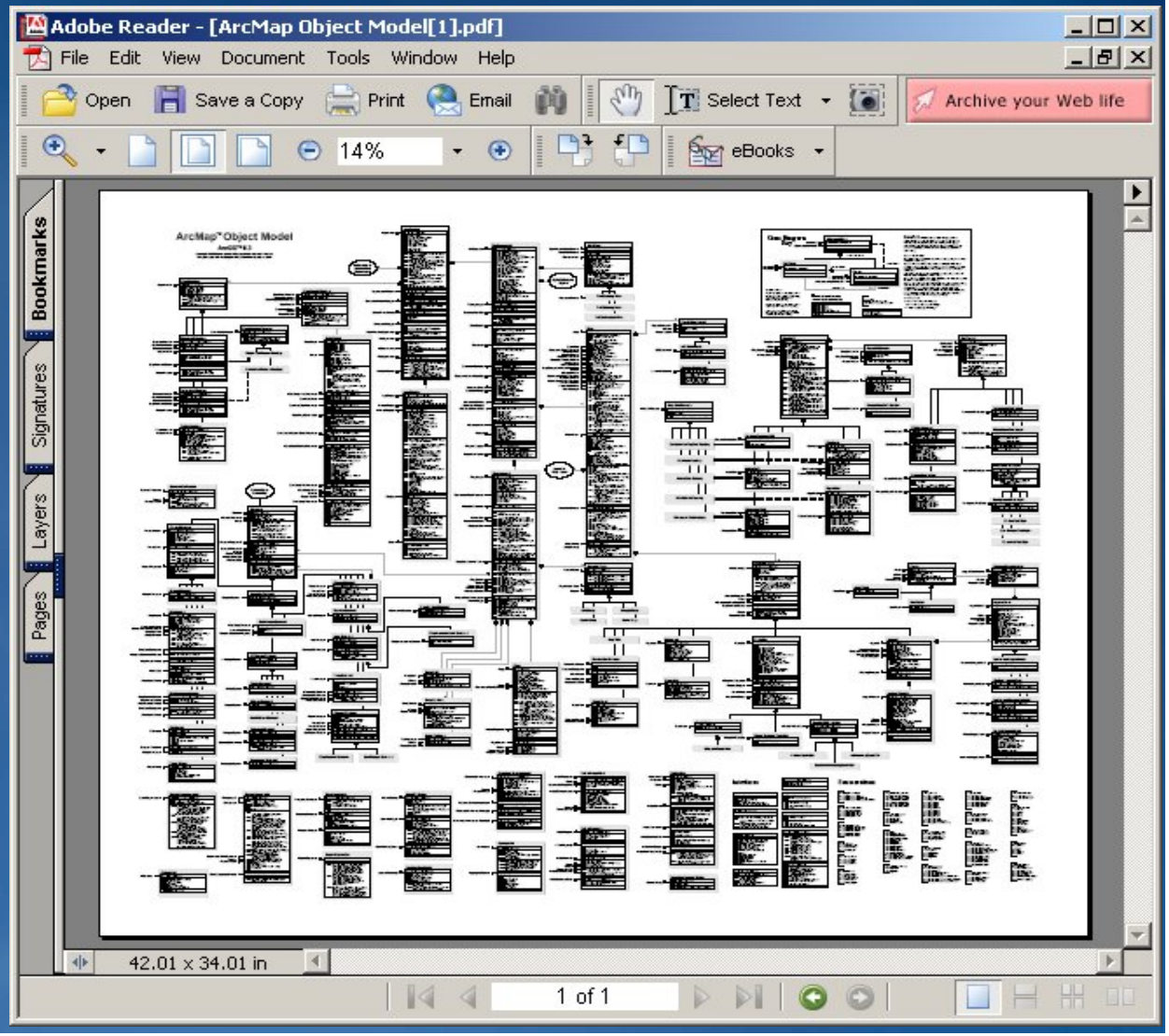
Object Model Diagrams

- Road maps to the ArcObjects classes
- Help you write code
- Based on UML – Unified Modeling Language
  - Symbols show relationships, connections, properties, and methods
- > 70 libraries
- > 110 posters

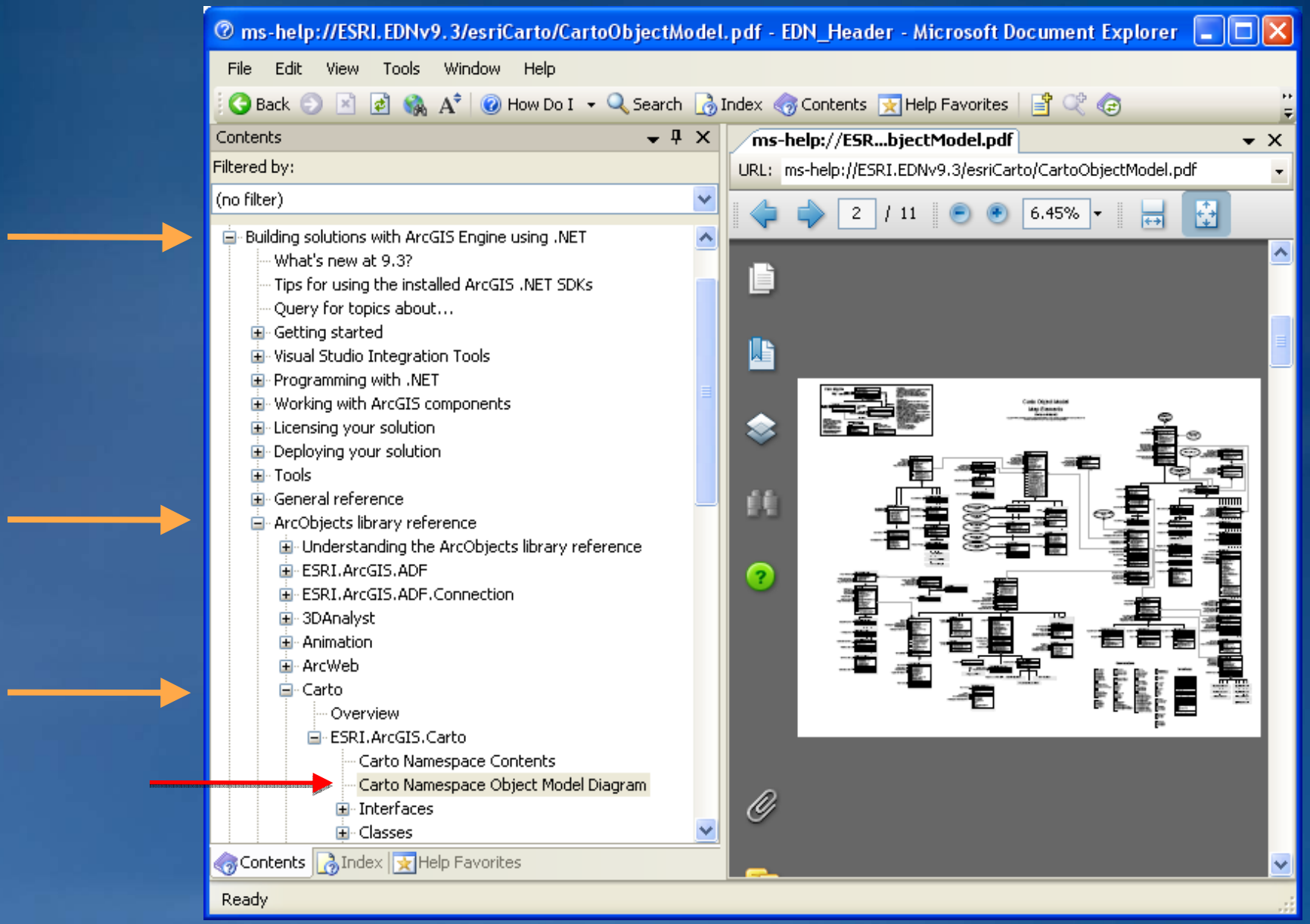


Where do you get the diagrams?

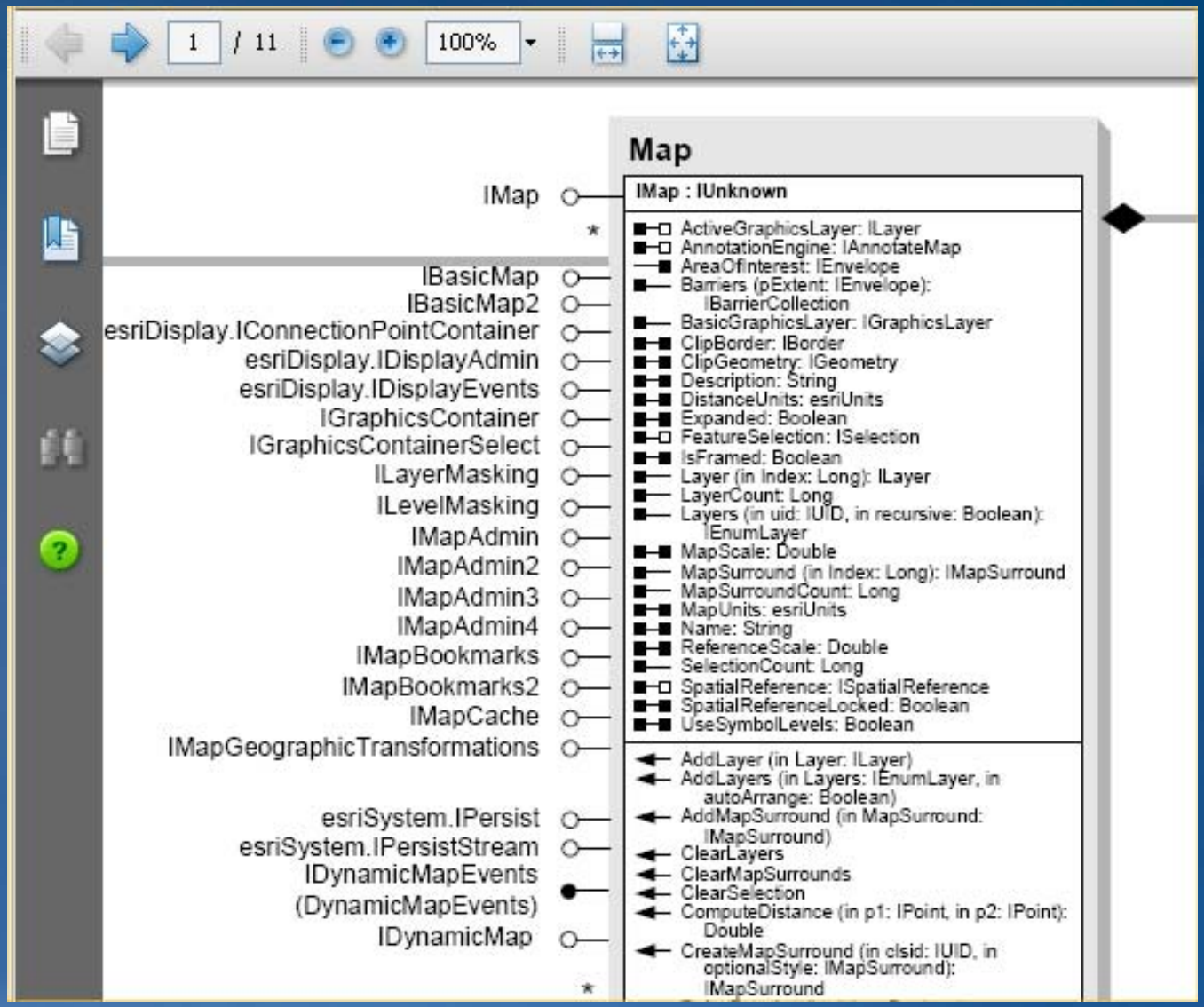
- Install the ArcObjects developer kit
  - C:\Program Files\ArcGIS\DeveloperKit\Diagrams*PDF
- Help
  - Per library
- Online
  - EDN Web site
  - Per library



Navigating the help: Carto library, diagram, and classes



Carto's Map class with lots of symbols
The symbols help you write code



The twelve UML Symbols

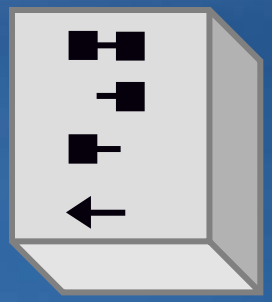
- Relationships



- Classes



- Properties and methods



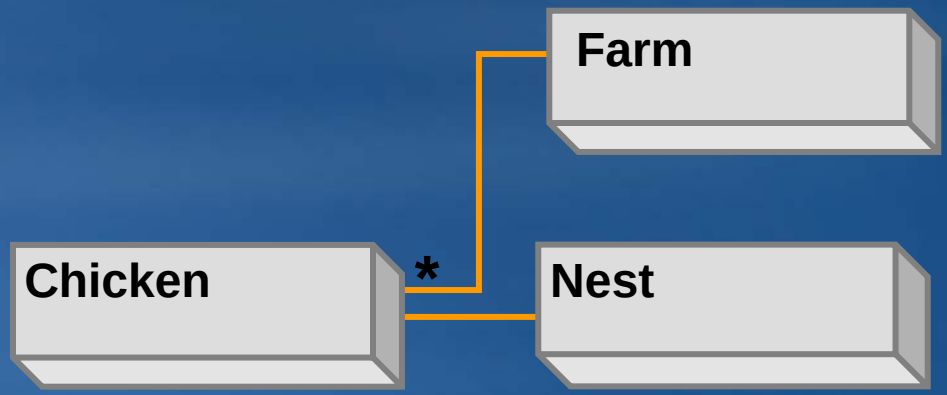
UML symbols

- **Association**



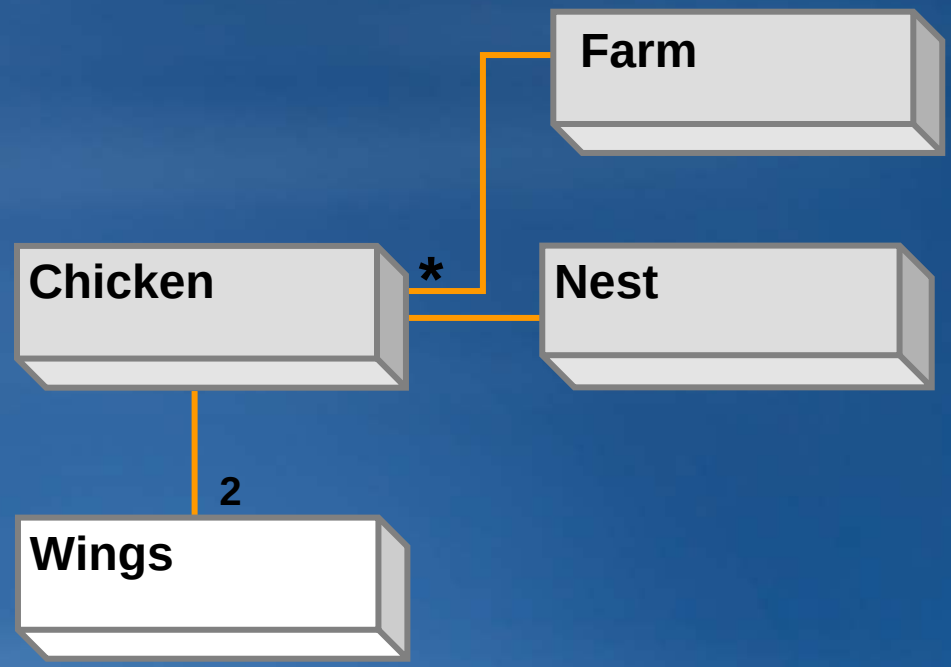
UML symbols

- **Multiplicity** *
- **Association** ——



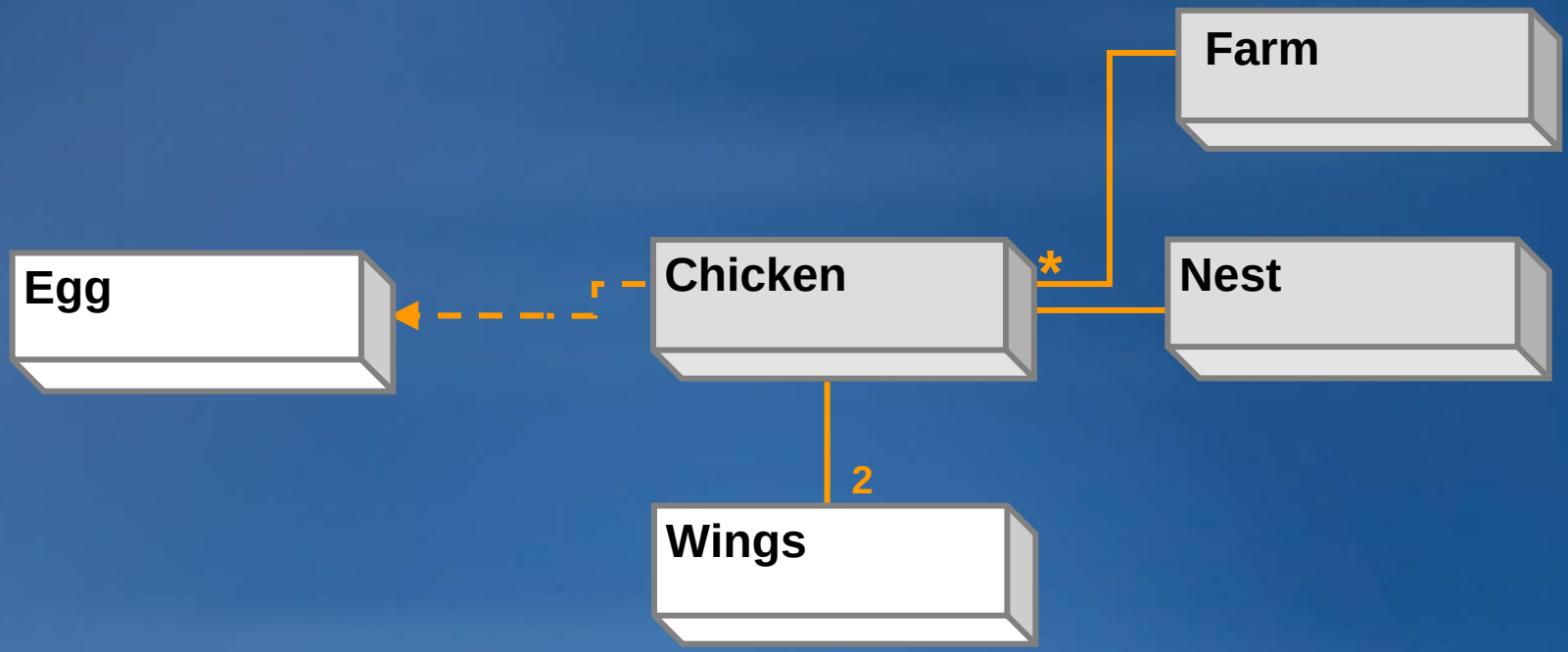
UML symbols

- Multiplicity
- Association



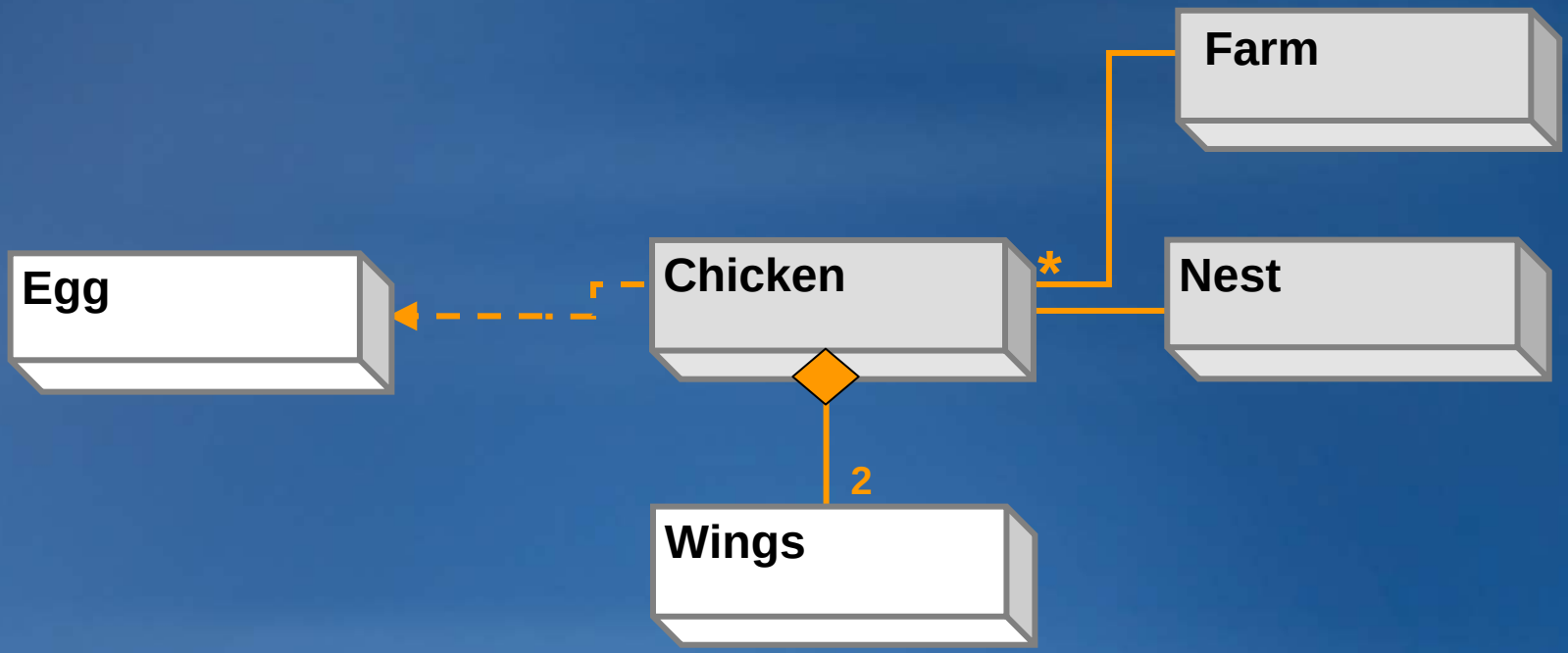
UML symbols

- Creates a
- Multiplicity
- Association



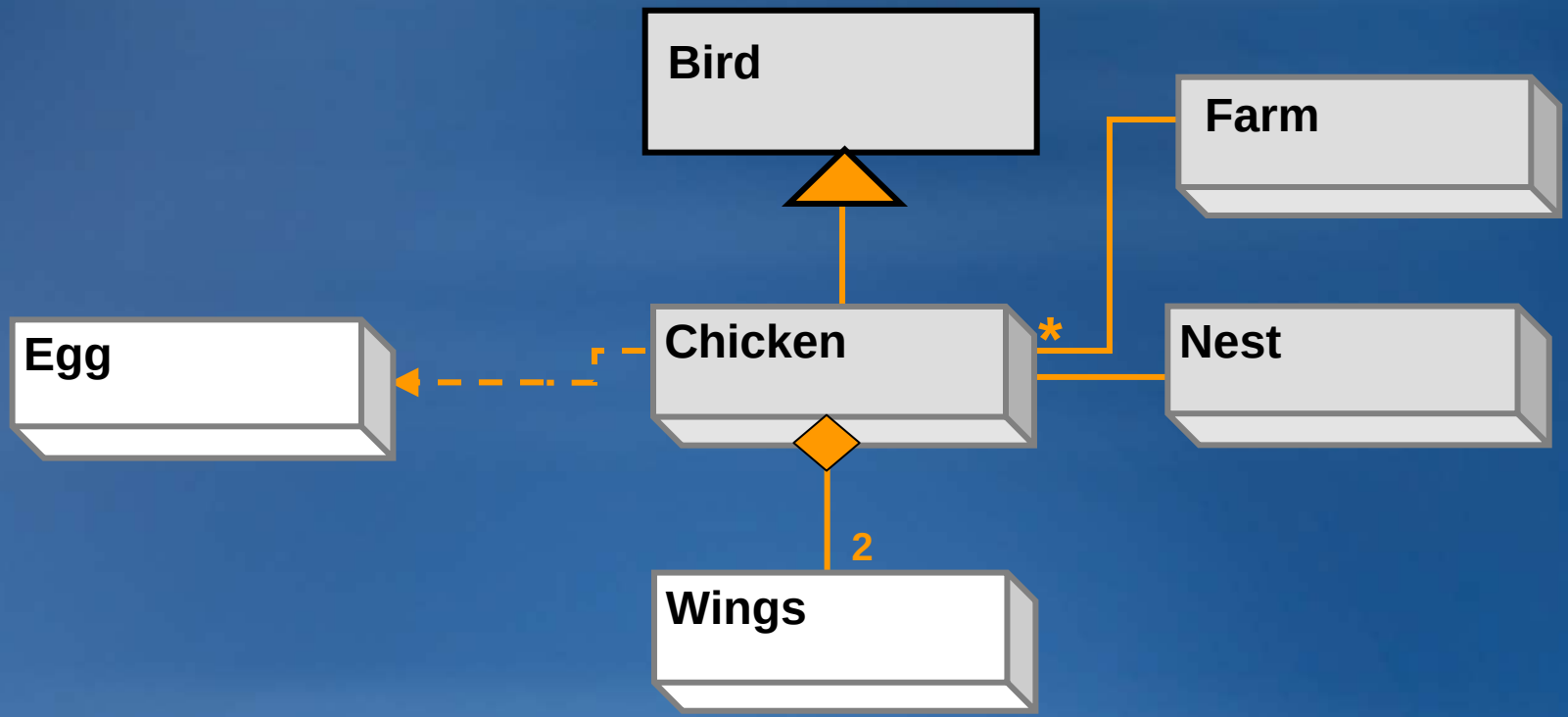
UML symbols

- Is composed of
- Creates a
- Multiplicity
- Association



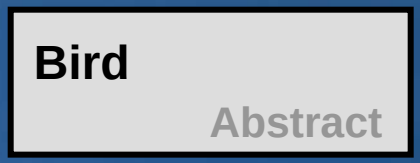
UML symbols

- Is a type of
- Is composed of
- Creates a
- Multiplicity
- Association



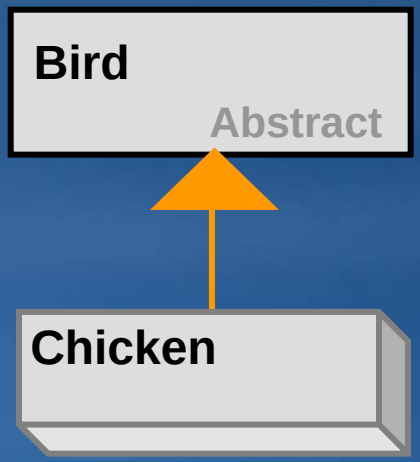
UML class symbols: Abstract class

- 2D and not shaded
- Objects can not be created from it



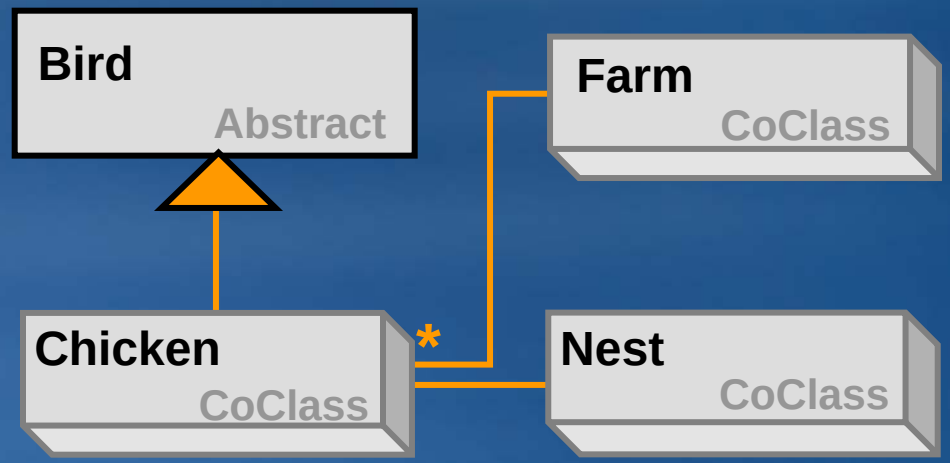
UML class symbols: Abstract class

- 2D and not shaded
- Objects can not be created from it
- Holds properties and methods that subclasses inherit



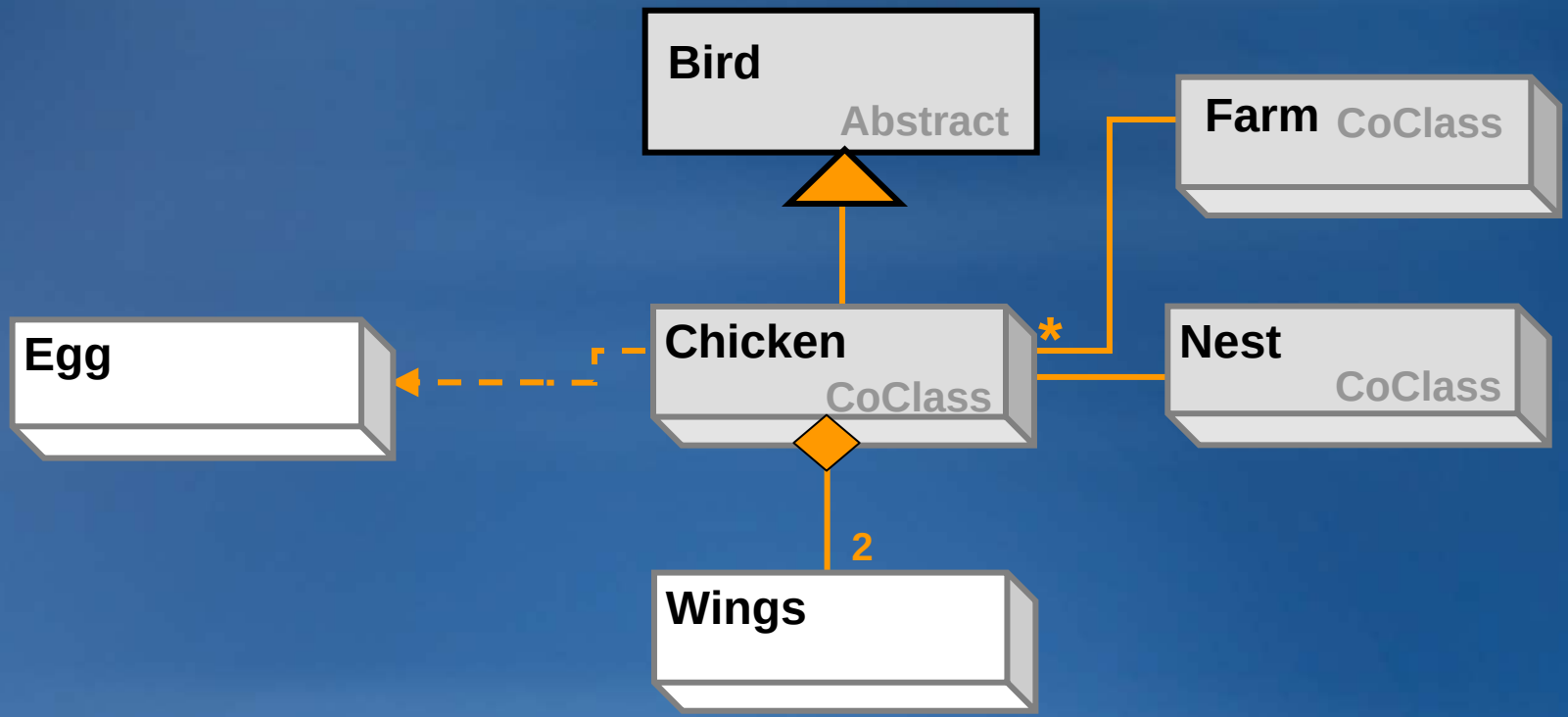
UML class symbols: CoClass

- 3D and shaded
- You create objects out of them
  - Declare a variable
  - Instantiate an object using the New keyword



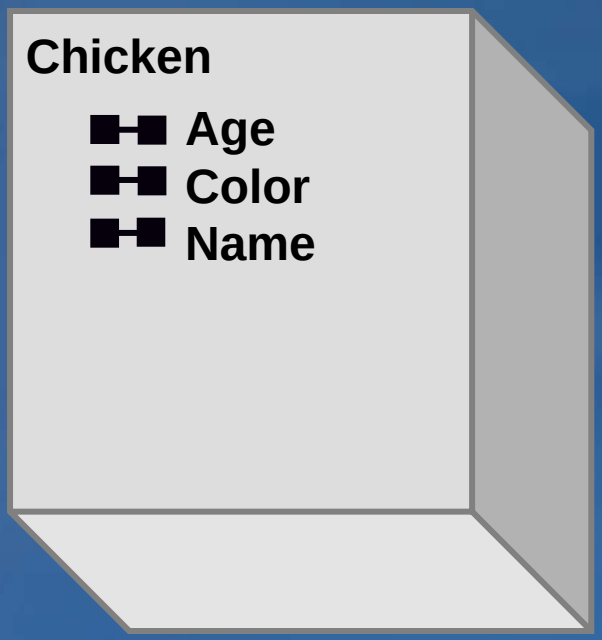
UML class symbols: Class

- Other classes create or return these objects
- You write code with another object to create or get
  - You can't create an egg without a chicken
  - You can't get a wing without an chicken



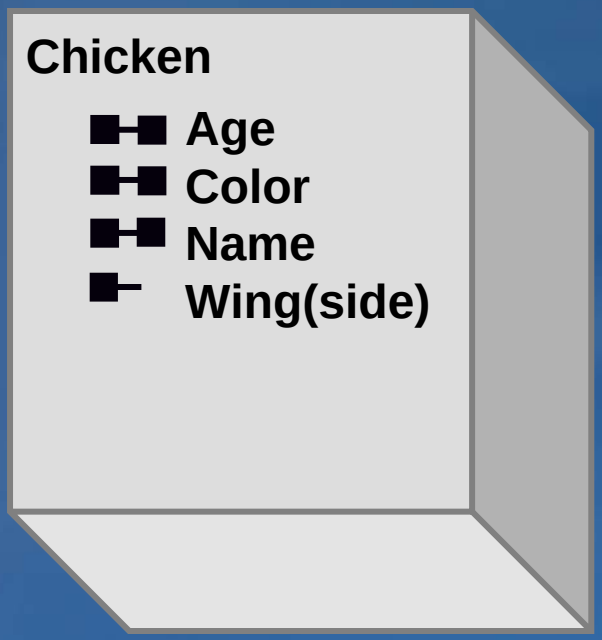
UML property symbols

- Read and write property 
  - These are attributes stored about the object
  - You can either get or set these properties



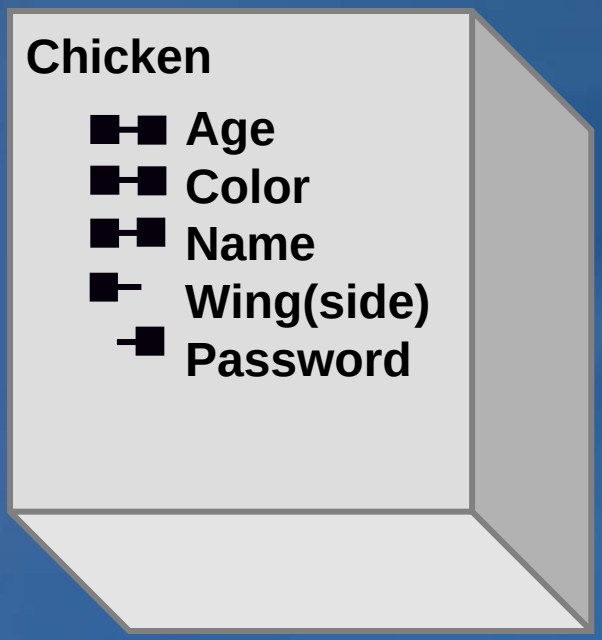
UML property symbols

- Read only property 
  - Left half barbell symbol
  - You can get this property's value
  - But you can't change it



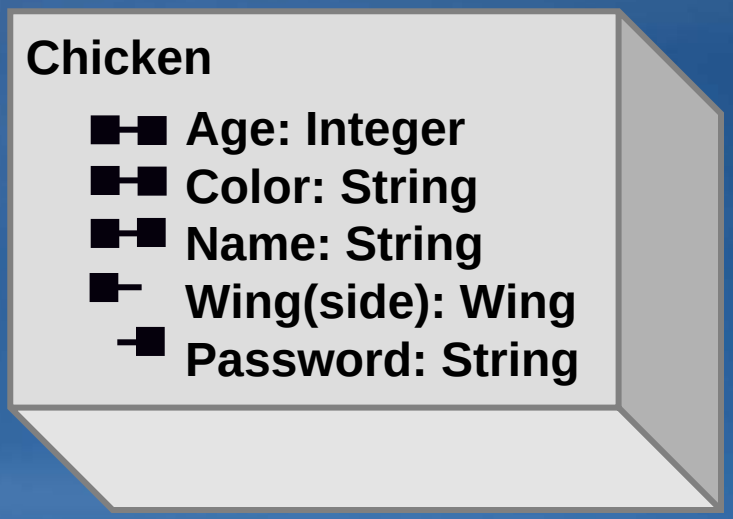
UML property symbols

- Write only property
  - Right half barbell symbol
  - Usually an edit property or like a password
  - You can change the value, but you can’t get it



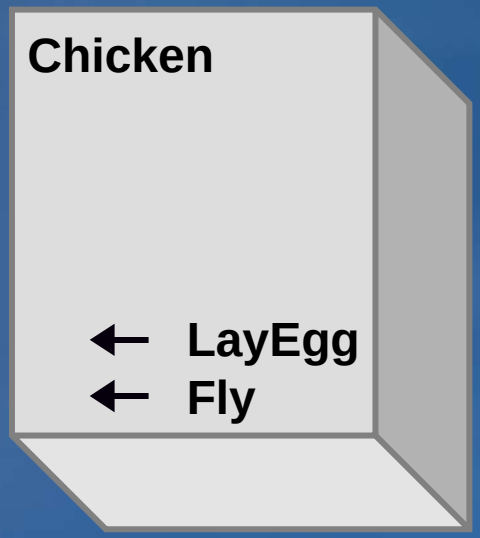
Property values

- Each property holds a value
- The values are of a certain type: Number, string, date, Boolean, object …
- The type appears to the right of the property name
  - Property name, a colon, and type
  - Name: String - means that the Name property holds a text string
  - The Wing property holds Wing objects



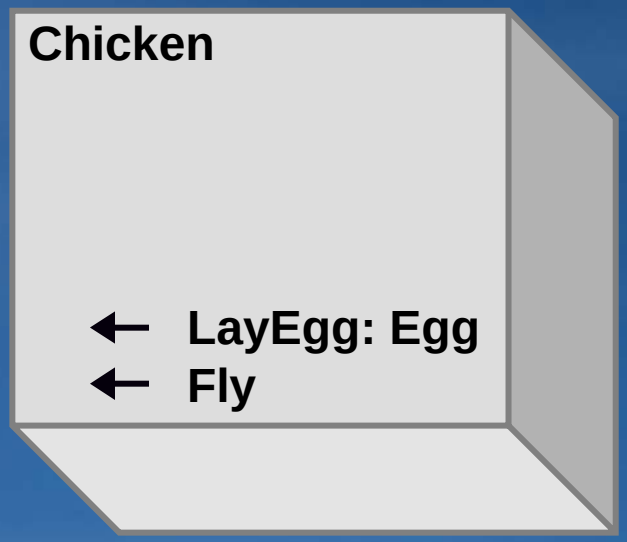
UML method symbol

- Method
  - Arrow symbol
  - Methods are actions the object can perform
  - Sometimes called behaviors



Methods return values

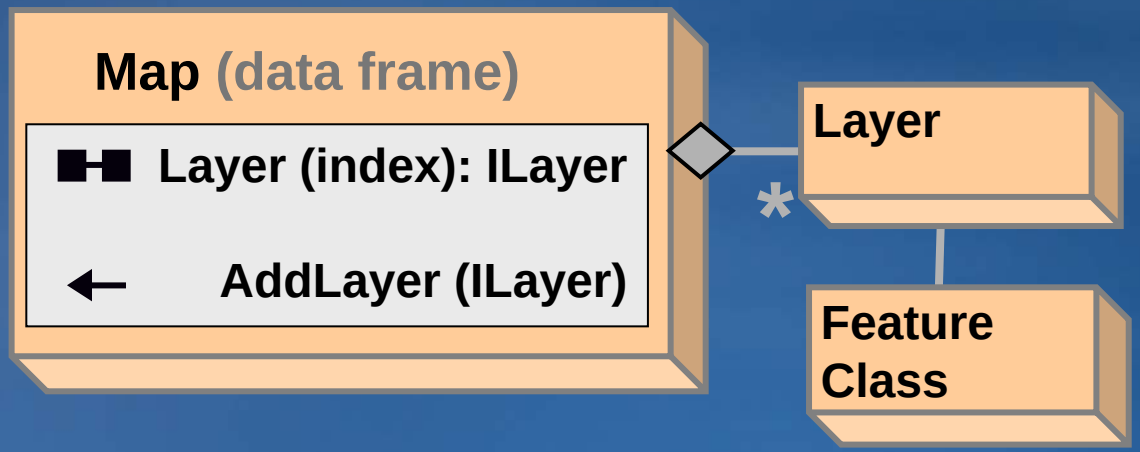
- You write code to run a method
- Some methods return a value, some don't
- The value's type appears to the right of the method
  - Method, colon, and type of its return value
  - LayEgg: Egg - the LayEgg method returns an egg object
  - The Fly method returns nothing

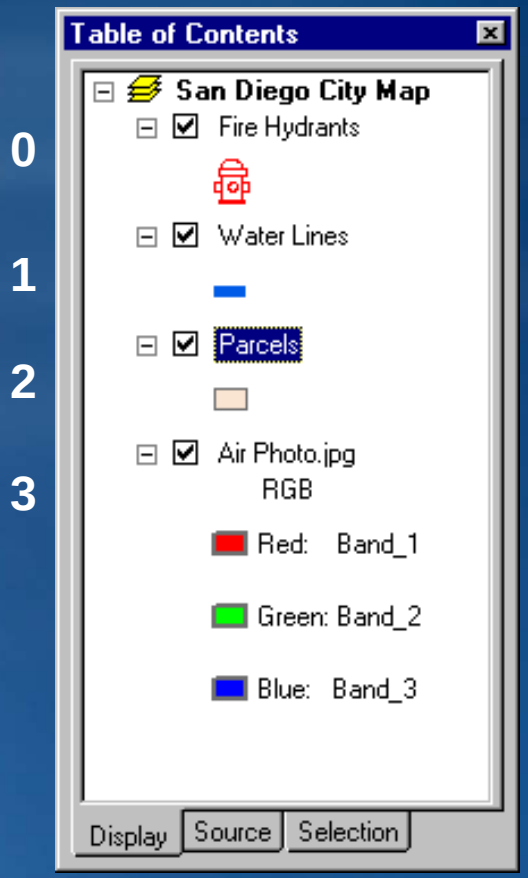


Reading diagrams

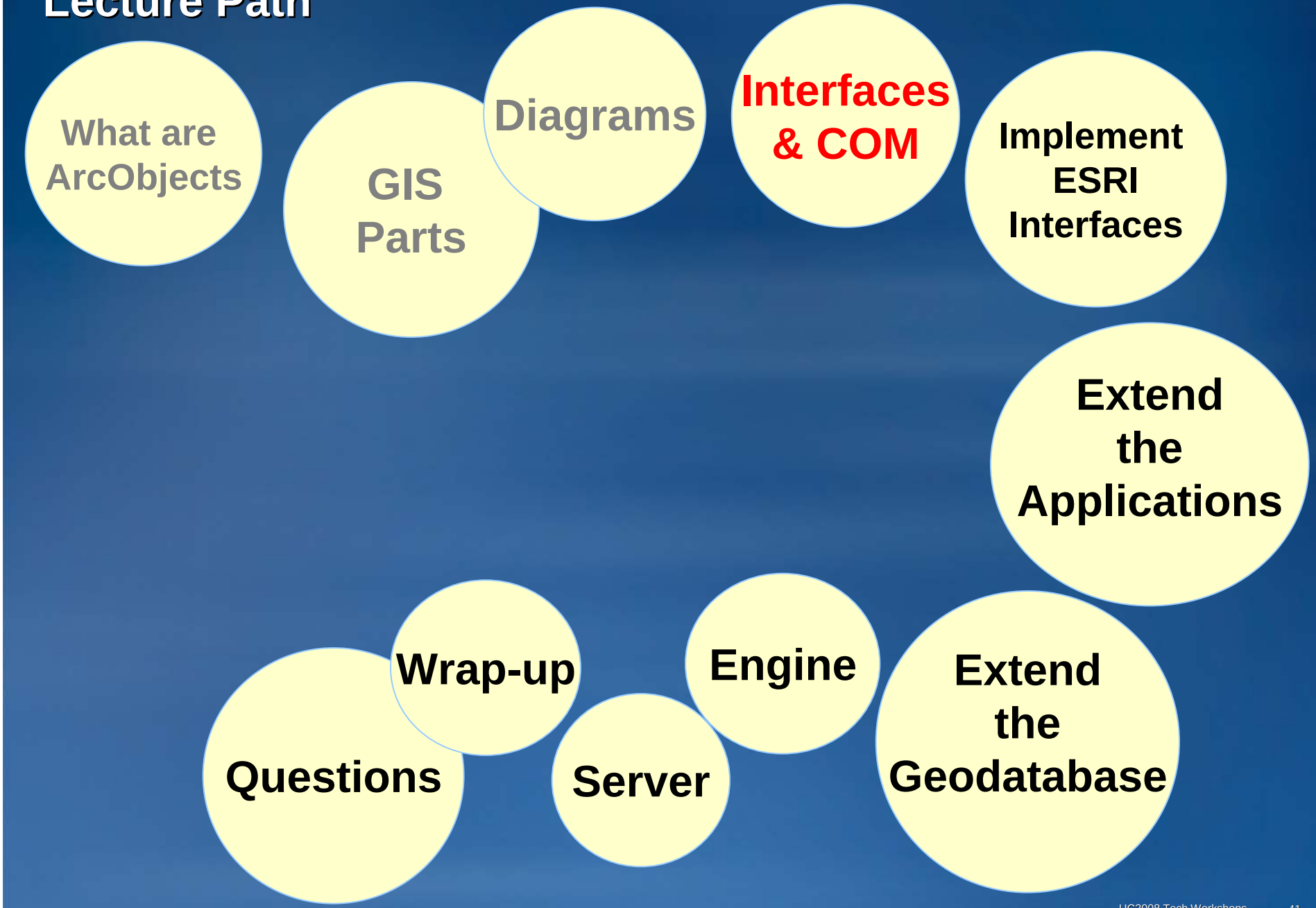
- Classes are rectangles
- Classes have properties and methods
- Get neighboring or connected objects

```
pLayer = pMap.Layer(0)
```





Lecture Path



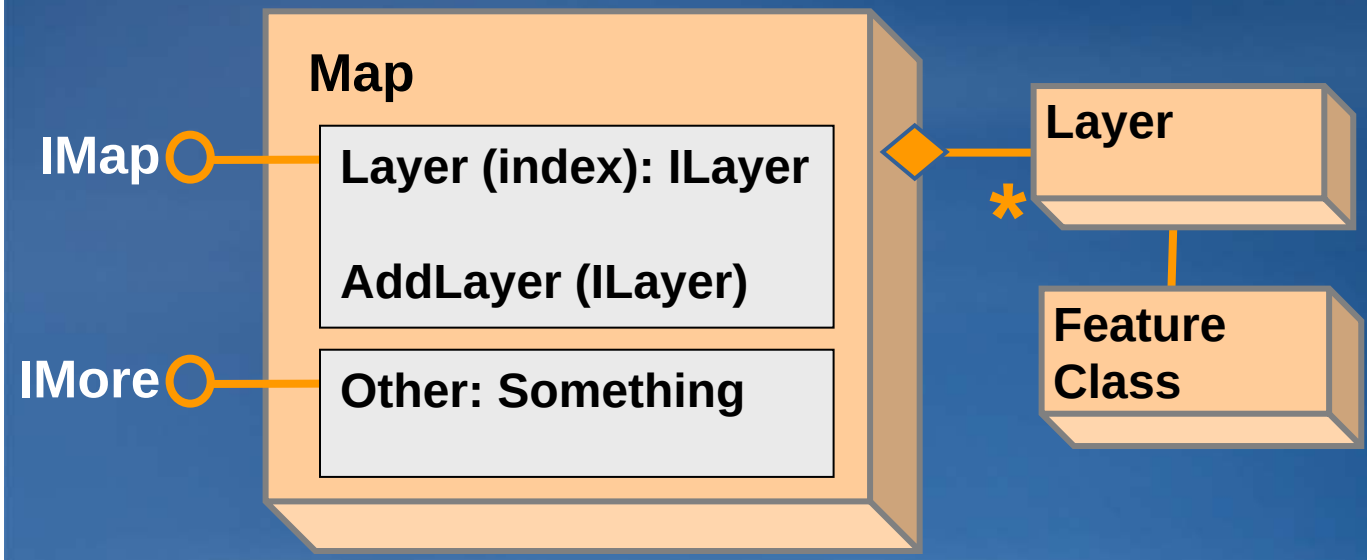
ArcObjects classes are COM

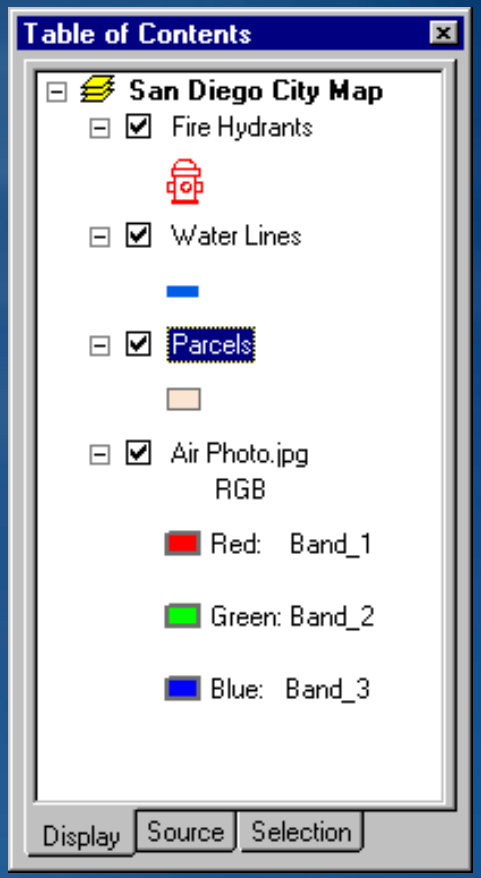
- COM compliant
  - Component object model
  - Industry standard for creating classes
  - Programming language independent
  - COM classes can be reused between applications
- COM classes have programmer interfaces
  - Classes created in one language can communicate with other languages
  - ArcObjects classes are created in C++
  - Use them in C++, VB.NET, C#.NET, VBA, Python, etc.

COM classes have interfaces

- Objects are rectangles
- Interfaces (lollypops) group properties and methods
- Get neighboring or connected objects

  pLayer = pMap.Layer(0)
- Multiple interfaces



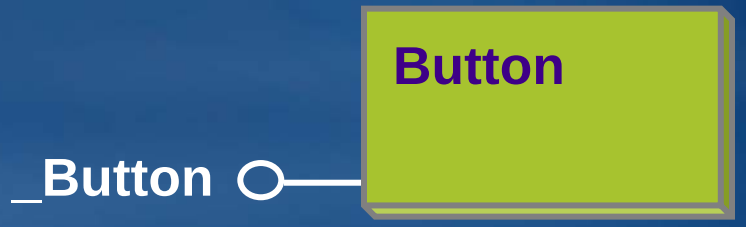


Understanding data types

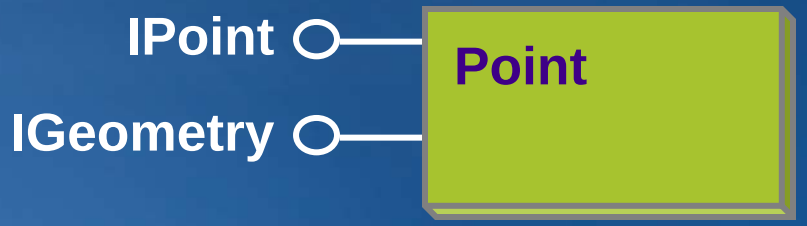
- Intrinsic data types
  - Numbers, strings, dates



- Simple objects (VB, Excel, MapObjects)
  - One default interface
  - Hidden



- ArcObjects
  - Multiple interfaces



Intrinsic data types and simple objects

```
'Intrinsic data
Dim x As Integer        'Declare
x = 4                   'Set
MsgBox x * 10           'Use
```

Integer

+ (add)
- (subtract)
* (multiply)
/ (divide)

```
'Simple Object
Dim b as CommandButton    'Declare
b = frmClock.cmdTime      'Set
b.Caption = “Time”        'Use
```

_CommandButton

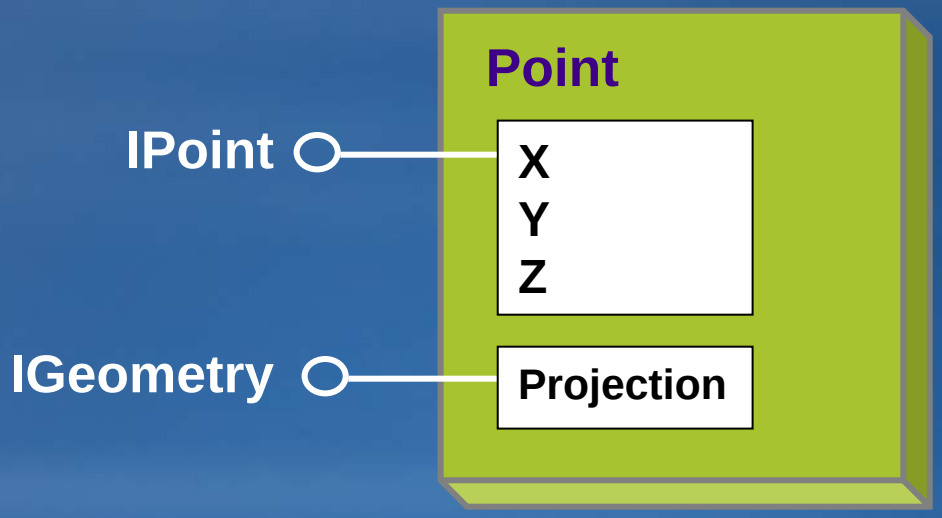
Commandbutton

Color
Enabled
Font
Caption
ToolTip

On a VBA form

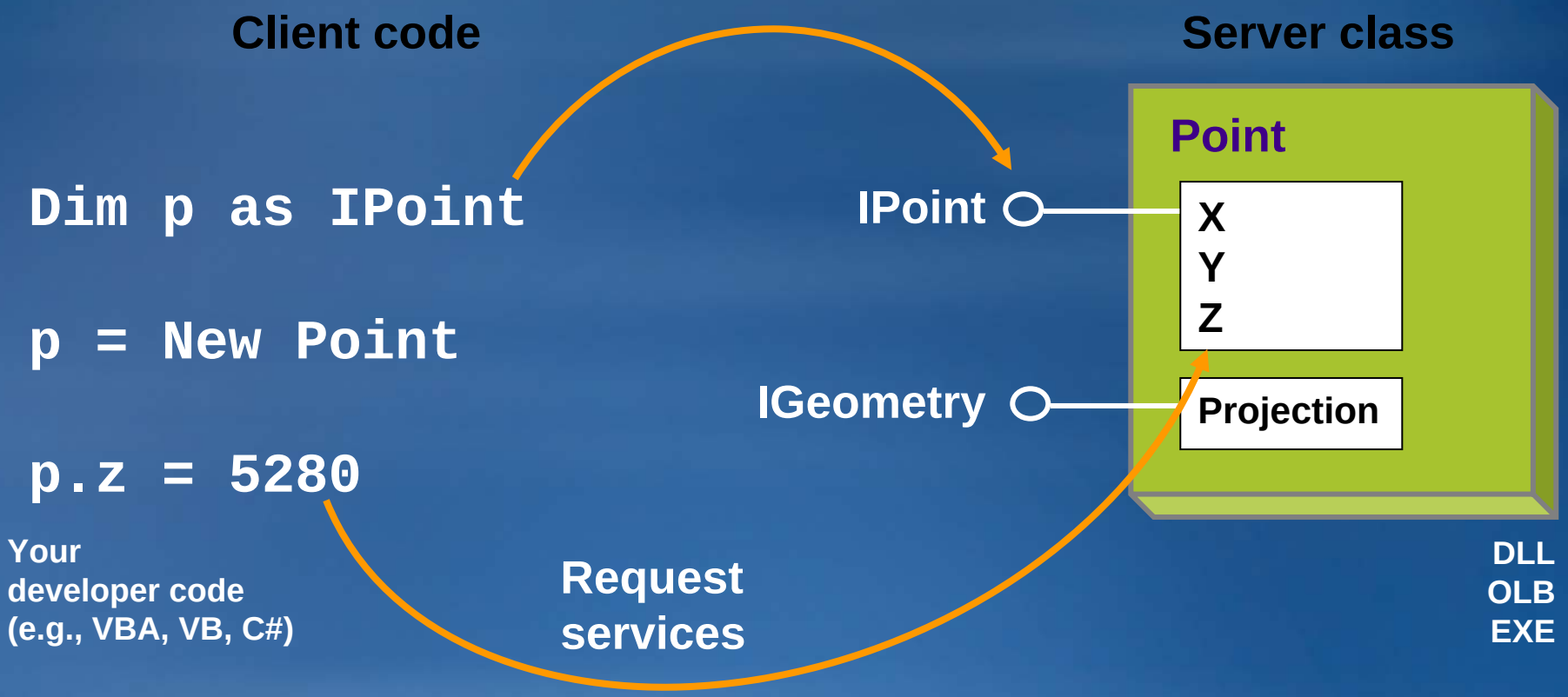
Multiple interfaces

```
'ArcObjects
Dim p as IPoint     'Declare
p = New Point       'Set
p.z = 5280          'Use
'Or
Dim p as IGeometry              'Declare
p = New Point                   'Set
p.Projection = Albers           'Use
```



Client and server environment

- Client code instantiates a class
- Client only knows the methods exist but does not know how they are implemented on the server



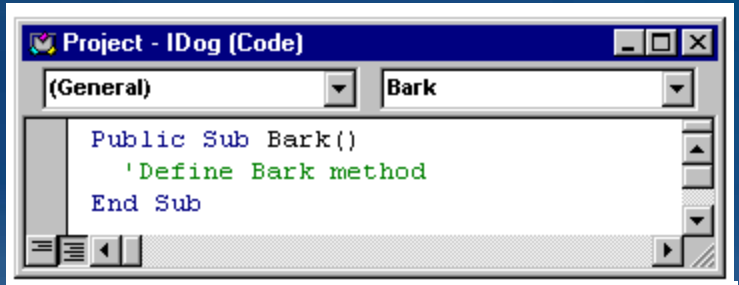
Dog COM class code

- Interface module
  - Define methods
  - No code
- Class module
  - Implement methods

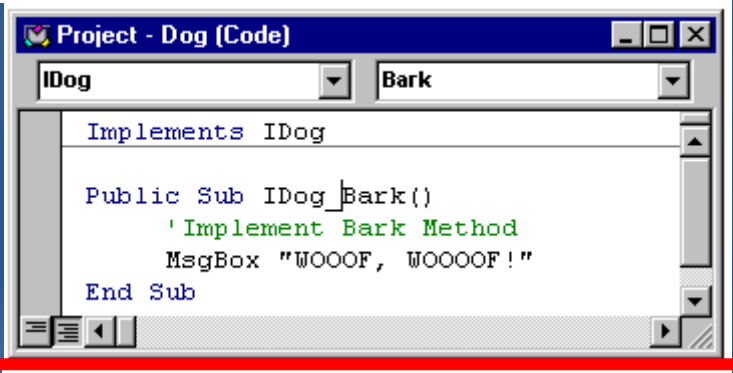
ESRI's ArcObjects code

Your code

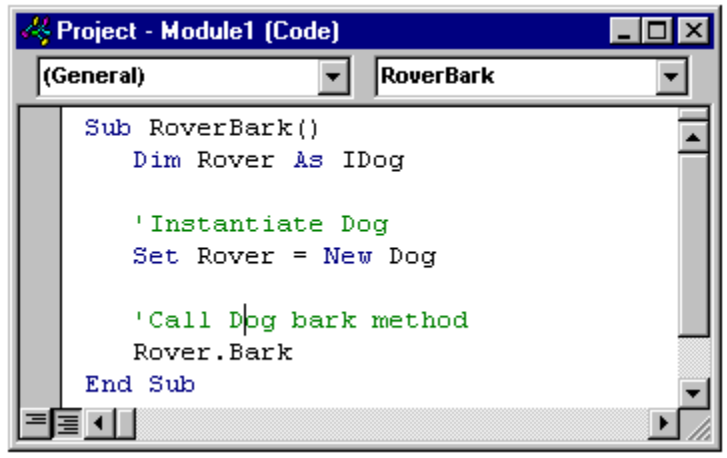
- Developer module
  - Instantiate class
  - Call its methods



Interface



Server



Client

What COM interfaces provide

- Programming language independence
- The ability for functionality in applications to evolve over time
  - Add new interfaces without affecting client code
  - IDog, IDog2, IDog3, IDog4, IDog5
  - Dim d as IDog works forever
- Interface reuse
- **Your classes can implement ESRI interfaces!**

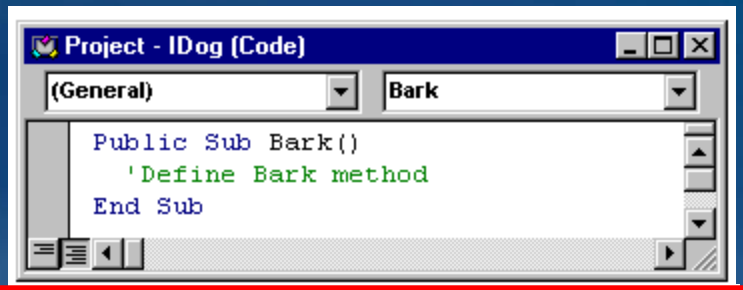
Implement ESRI interfaces

- Interface module

ESRI's ArcObjects code

Your code

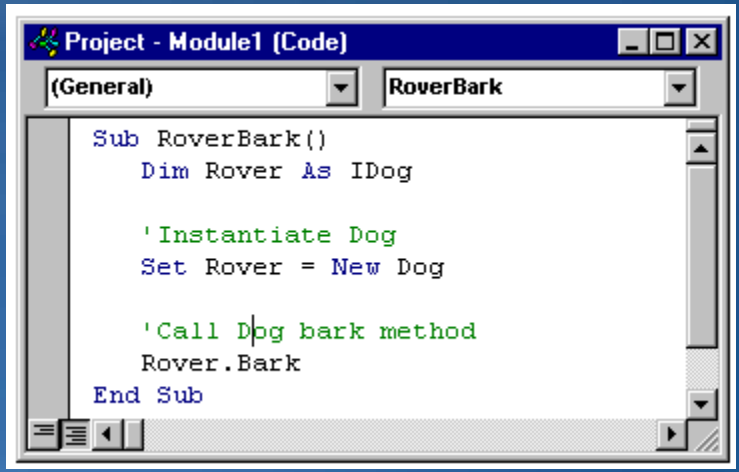
- Class module
  - Implement methods
  - Write your own code
- Developer module
  - Instantiate class
  - Call its methods



Interface

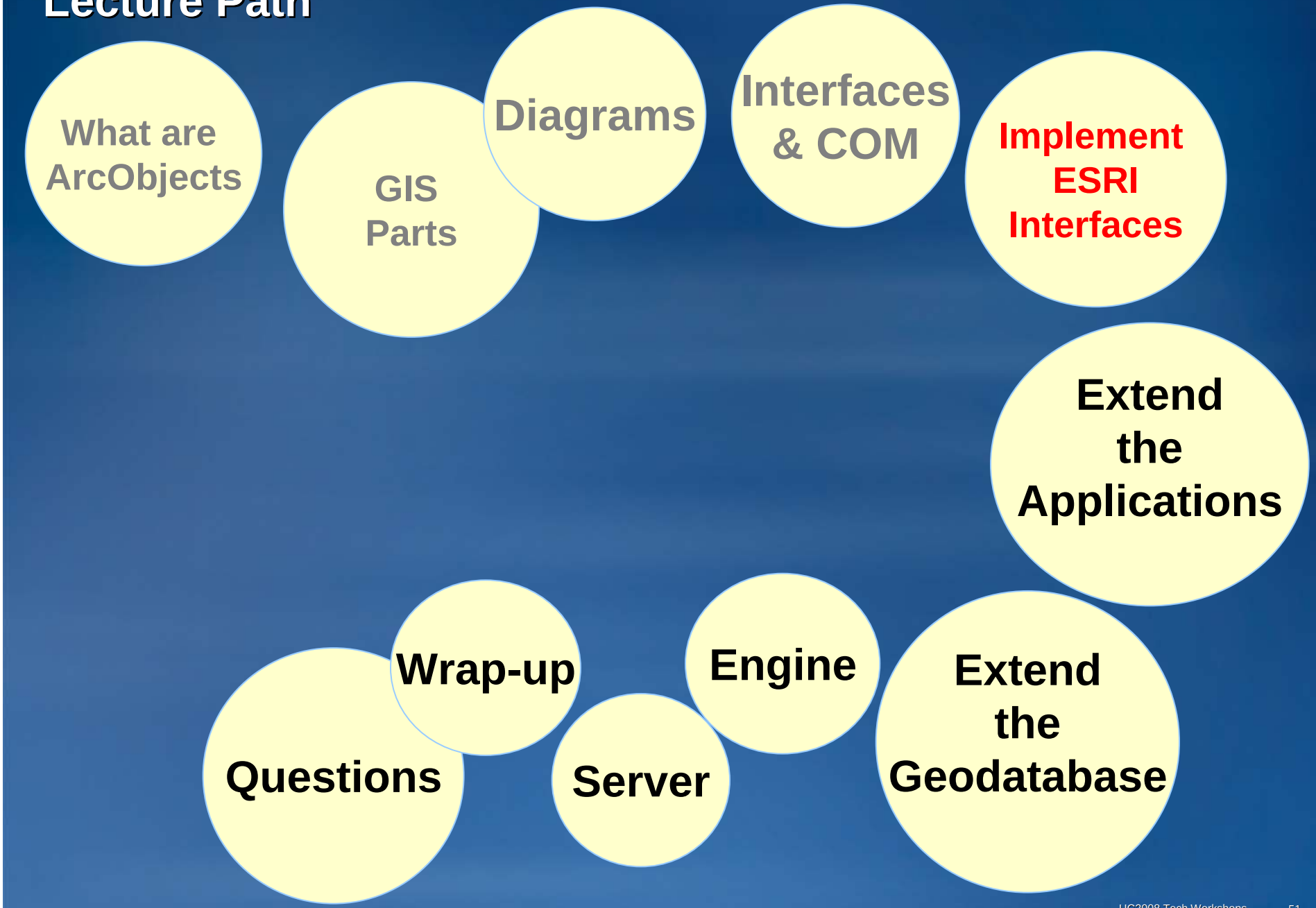


Your Server



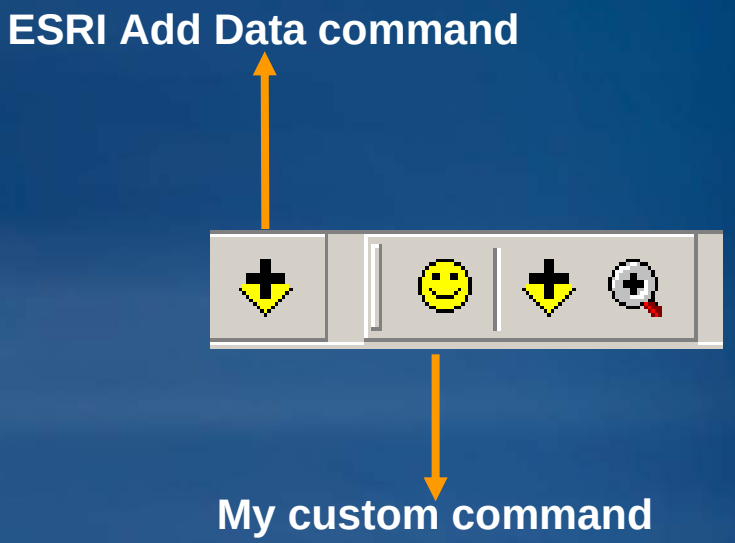
Client

Lecture Path



Creating a custom command

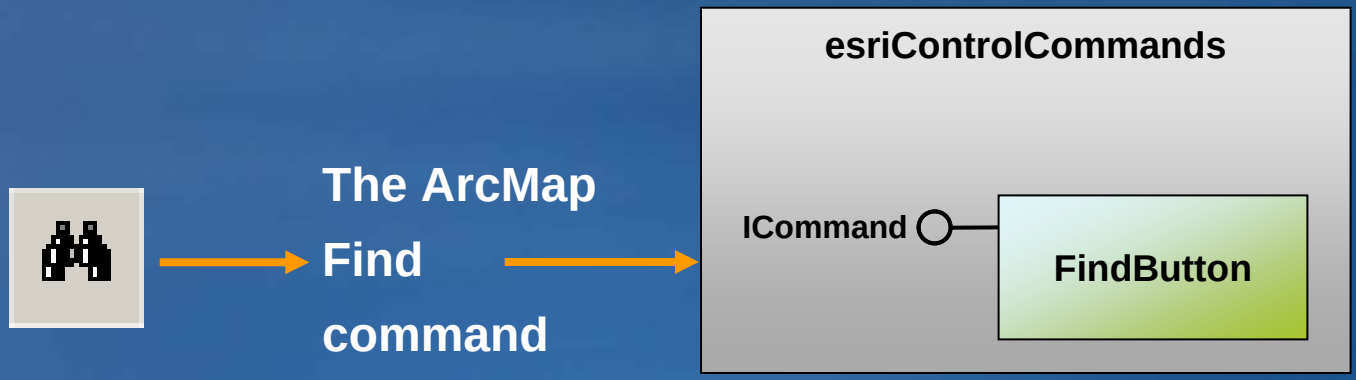
- Single-click buttons
- Execute any ArcObjects code
- Examples
  - Custom add data
  - Custom mapping operations
  - Show a custom dialog or window



Developing custom COM components

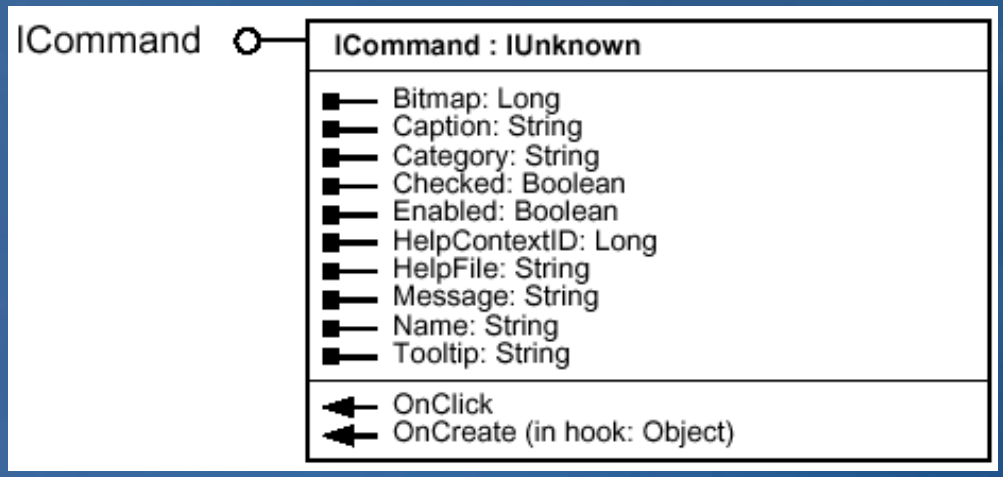
- Find a similar ArcGIS component
- Find out what interfaces it implements
  - Look in the Help
- Buttons implement ICommand

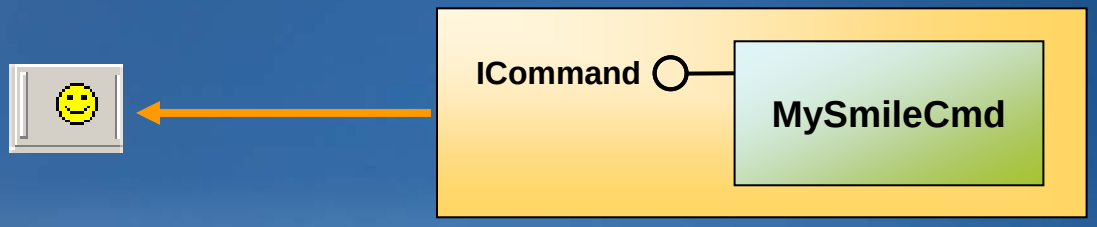




Commands: Implement the ICommand interface

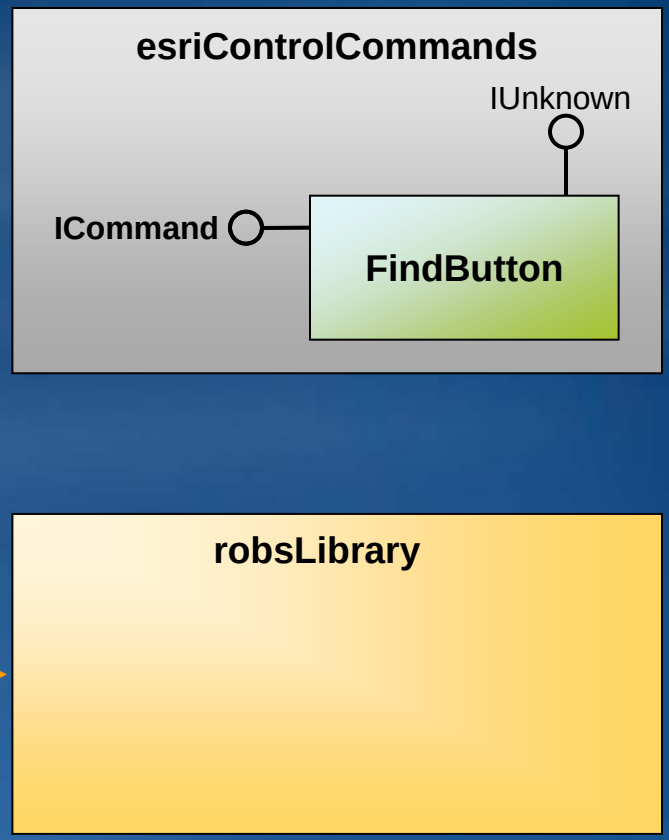
- ESRI developers follow this pattern
- Model your custom button after ESRI's Button class
- You get to code each property and method





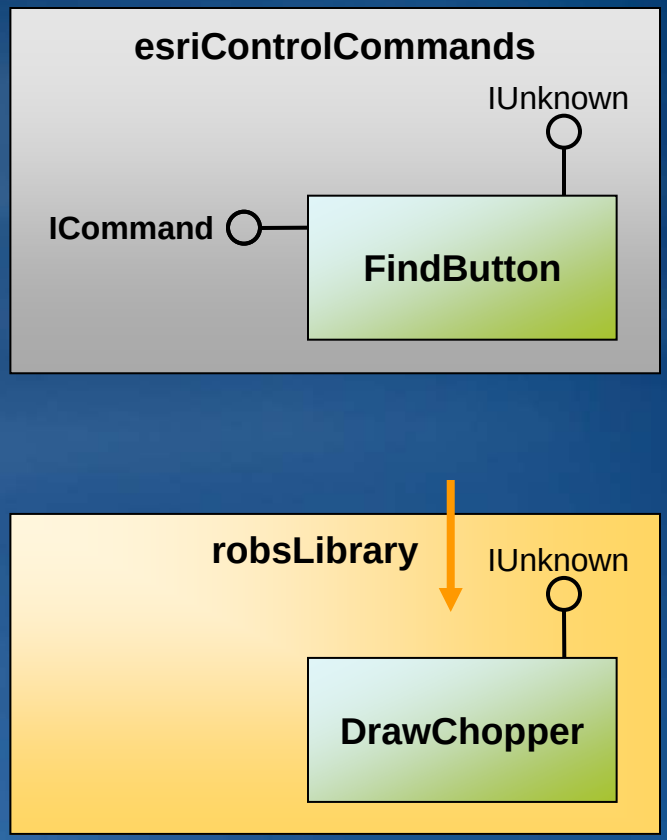
Developing custom COM components

- Create a COM object and plug it into an application
- Steps:
  1. Create a COM/.NET project



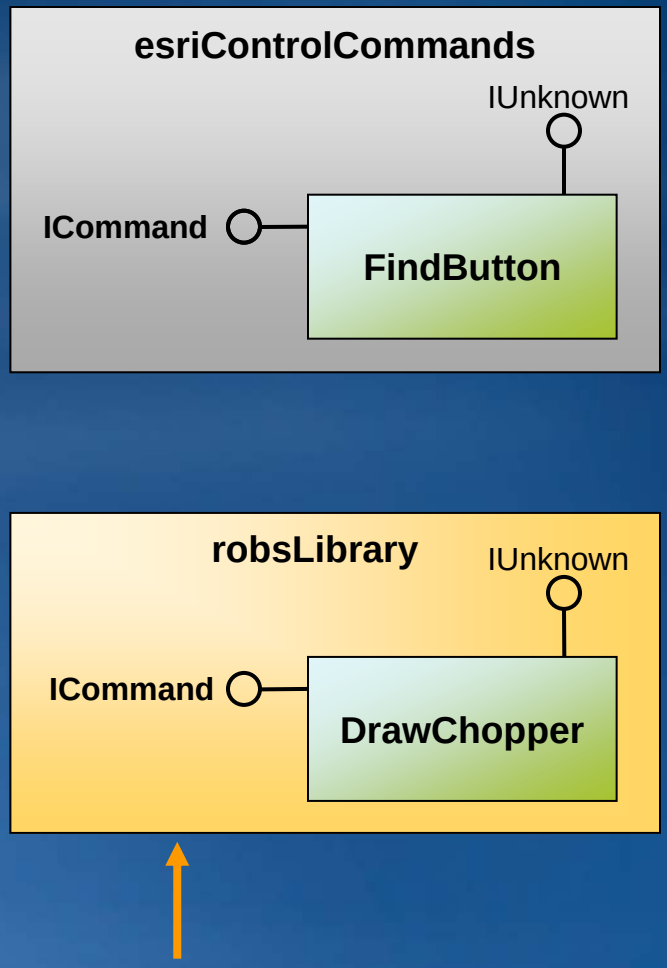
Developing custom COM components

- Create a COM object and plug it into an application
- Steps:
  1. Create a COM/.NET project
  2. Create a COM class



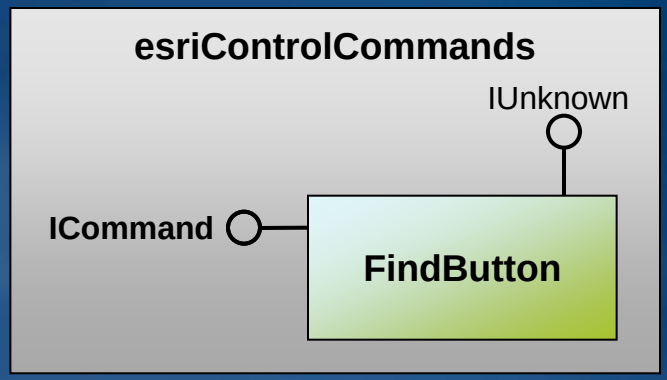
Developing custom COM components

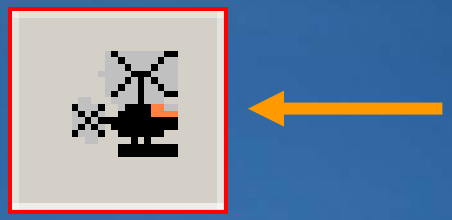
- Create a COM object and plug it into an application
- Steps:
  1. Create a COM/.NET project
  2. Create a COM class
  3. Reference the ArcGIS libraries
  4. Implement ArcObjects interfaces

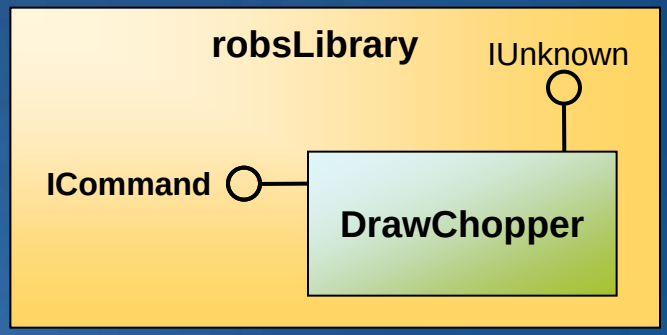


Developing custom COM components

- Create a COM object and plug it into an application
- Steps:
  1. Create a COM/.NET project
  2. Create a COM class
  3. Reference the ArcGIS libraries
  4. Implement ArcObjects interfaces
  5. Compile
  6. Register in an ArcGIS component category (Next Slide)







How does ArcMap know to use my class?

- ArcMap application start up cycle starts
  1. Accesses the appropriate component category

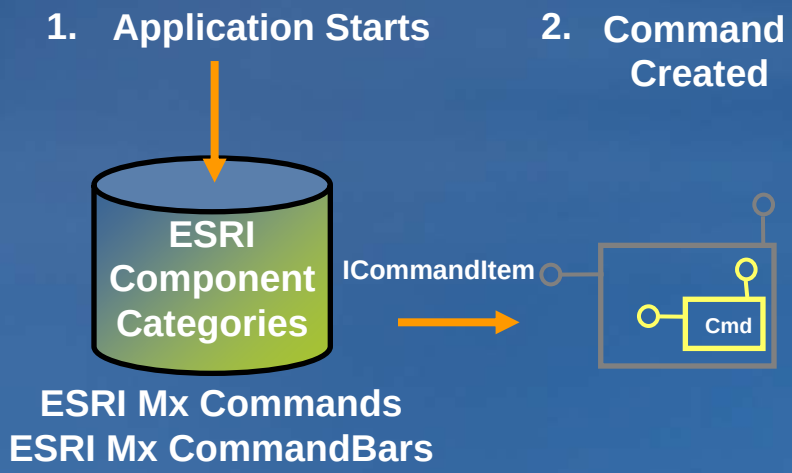
1. Application Starts



ESRI Mx Commands
ESRI Mx CommandBars

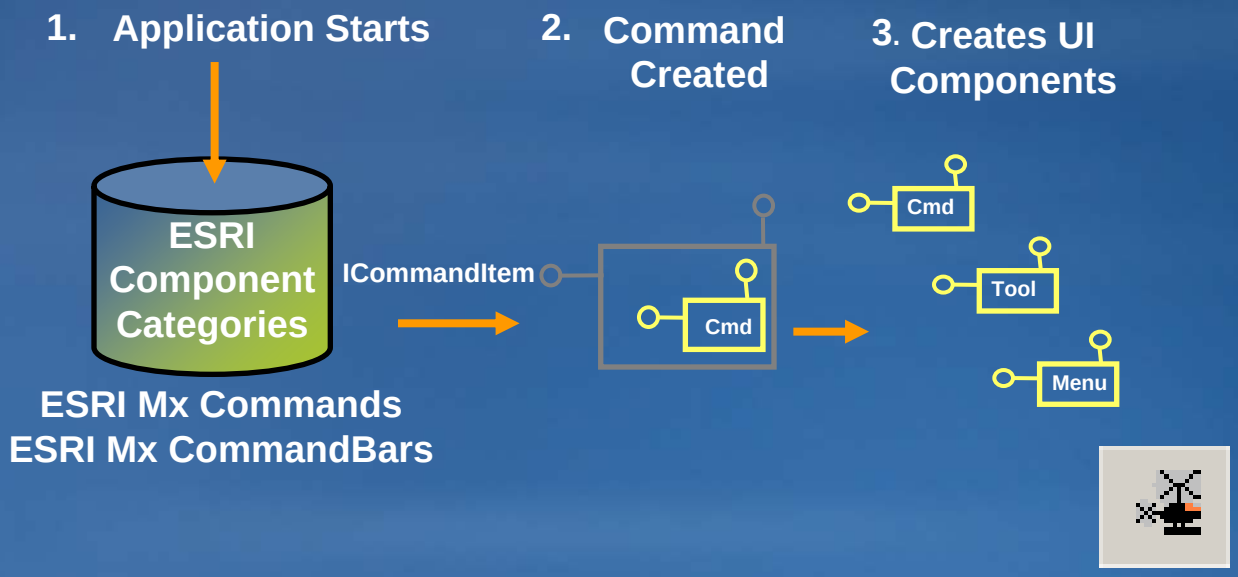
Application start up cycle

- ArcMap starts
  1. Accesses the appropriate component category
  2. Creates an internal objects that implements ICommand



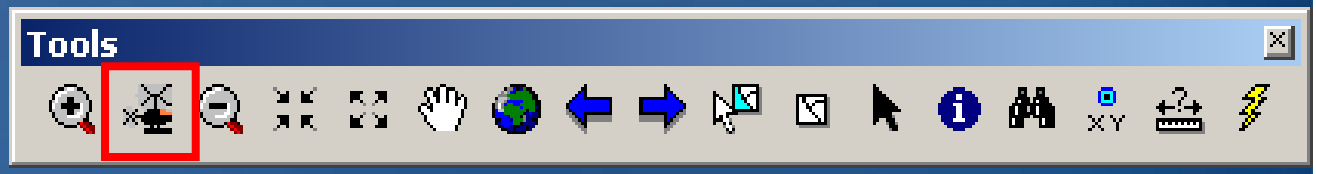
Application start up cycle

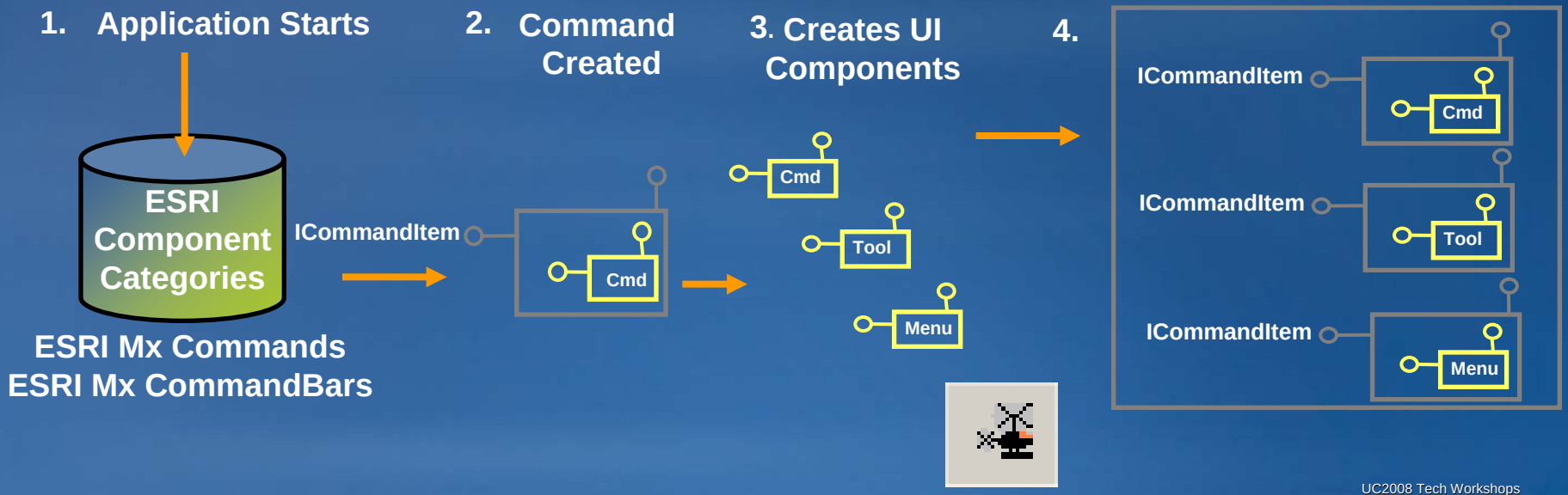
- ArcMap starts
  1. Accesses the appropriate component category
  2. Creates an internal objects that implements ICommand
  3. Creates your UI component (command, tool, toolbar, or menu)



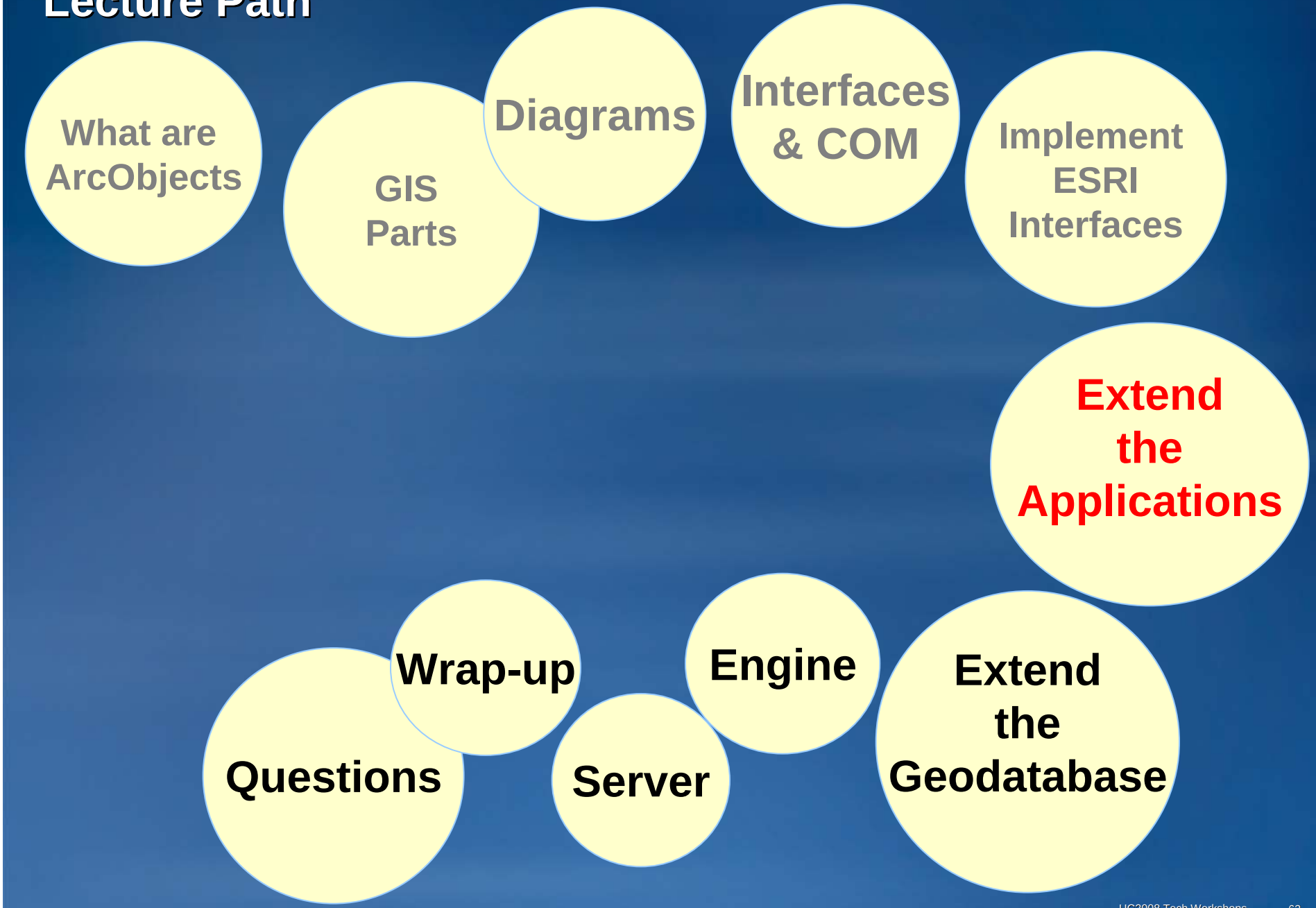
Application start up cycle

- ArcMap starts
  1. Accesses the appropriate component category
  2. Creates an internal objects that implements ICommand
  3. Creates your UI component (command, tool, toolbar, or menu)
  4. Adds the CommandItem to the CommandItem list



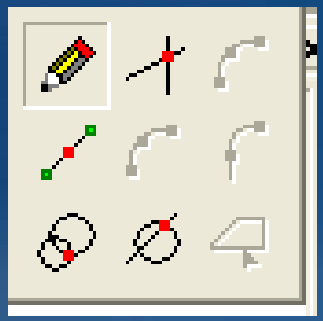


Lecture Path

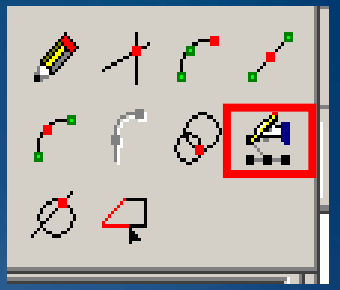


Custom tools also implement ITool

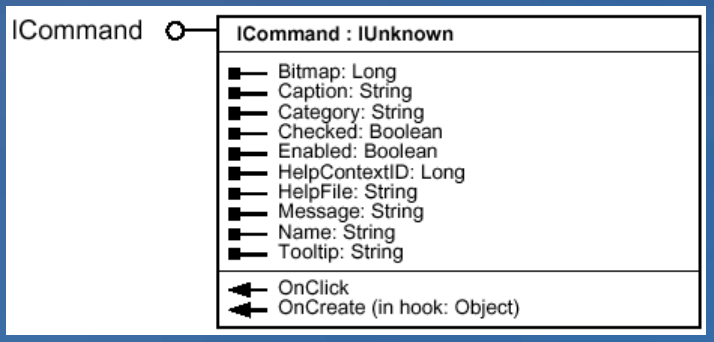
- Commands that listen for mouse and key events
- Allows users to interact with maps
- Examples
  - Map interaction: Tracking
  - Analysis: Select features
  - Editing tools

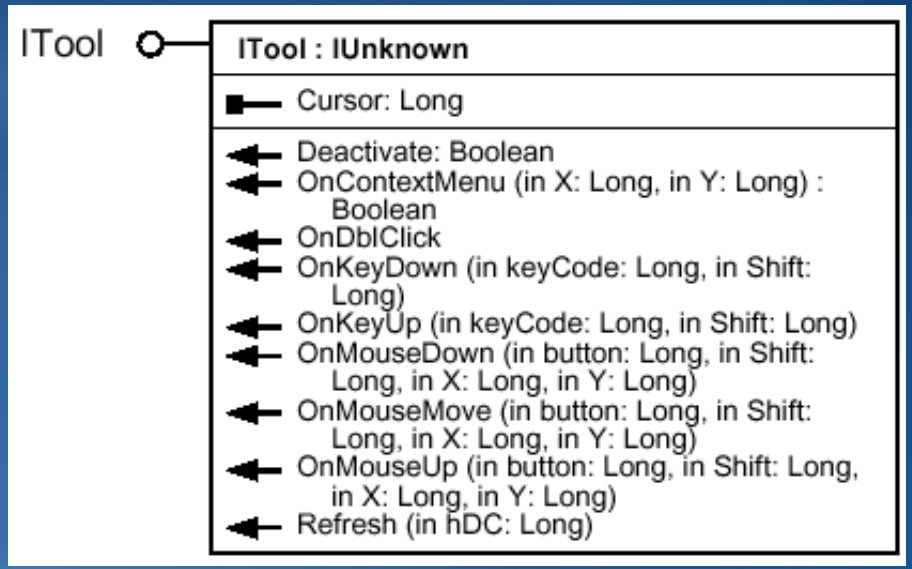


Default Sketch palette



Custom sketch tool in palette

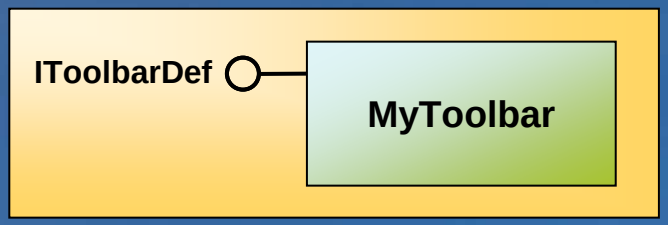


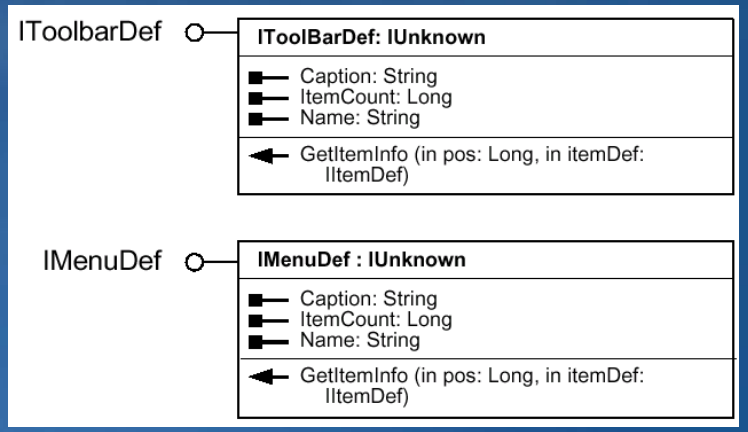


Custom toolbars

- Container to show commands, tools, and menus
- Implement IToolbarDef
- You code the interface's properties and methods
- Examples
  - Show or hide a custom set of tools
  - Associate tools with extensions



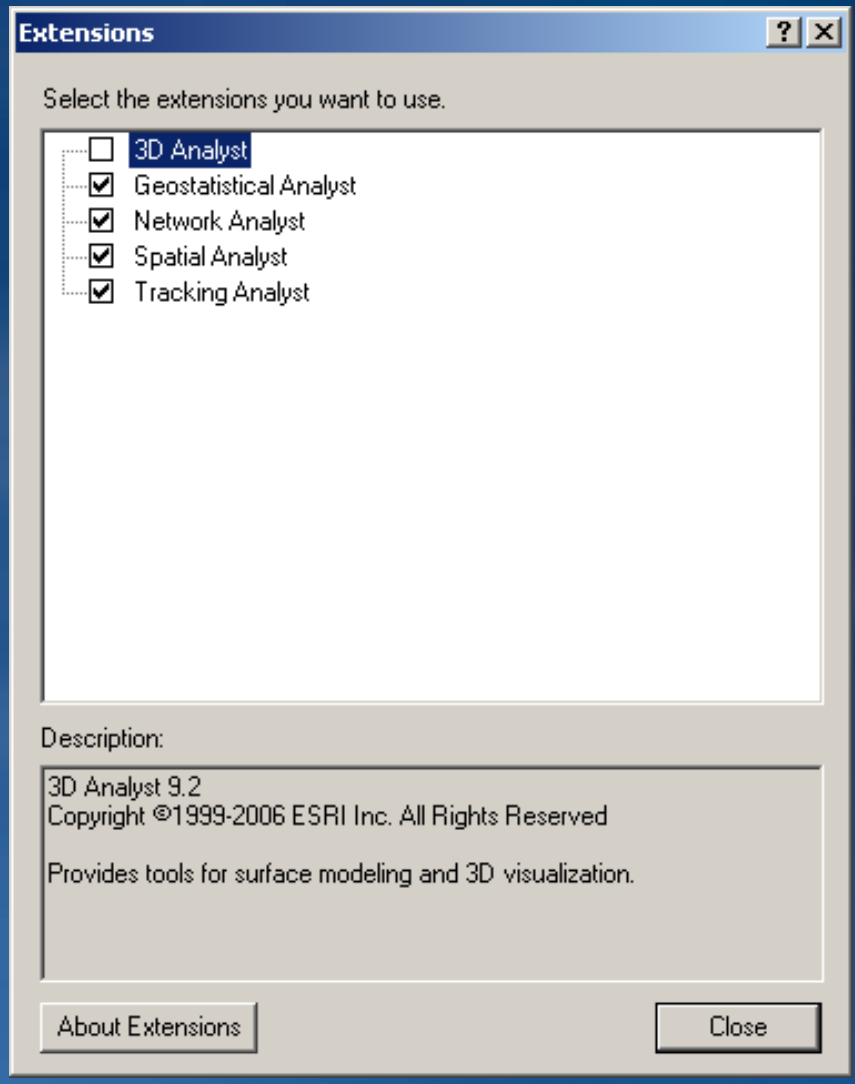




ArcGIS extensions

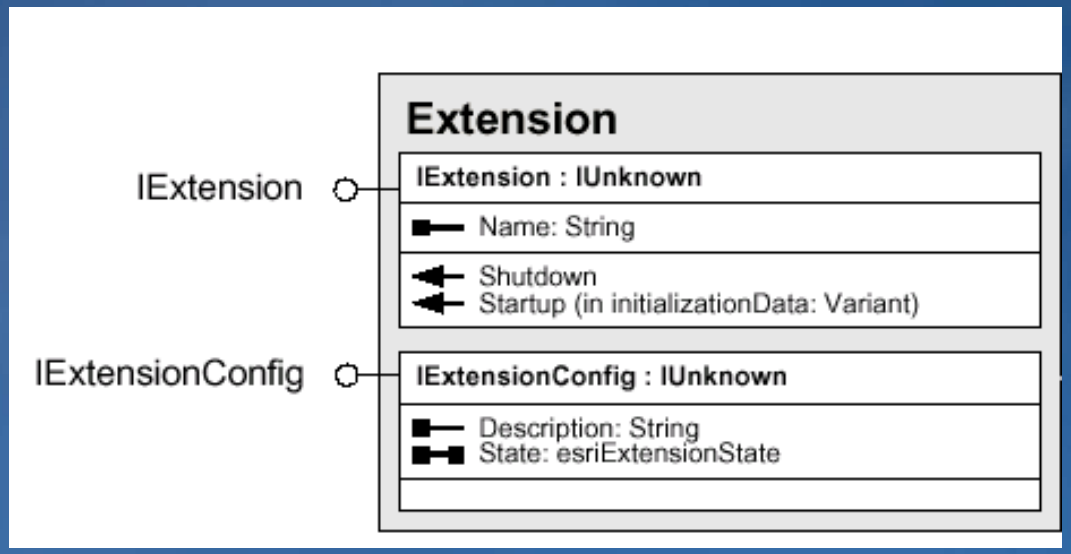
- Mechanism to plug objects into the application
- Visible in the Extension Manager
- Supported by all applications
- Name appears in the list
- Looks like ESRI's extensions

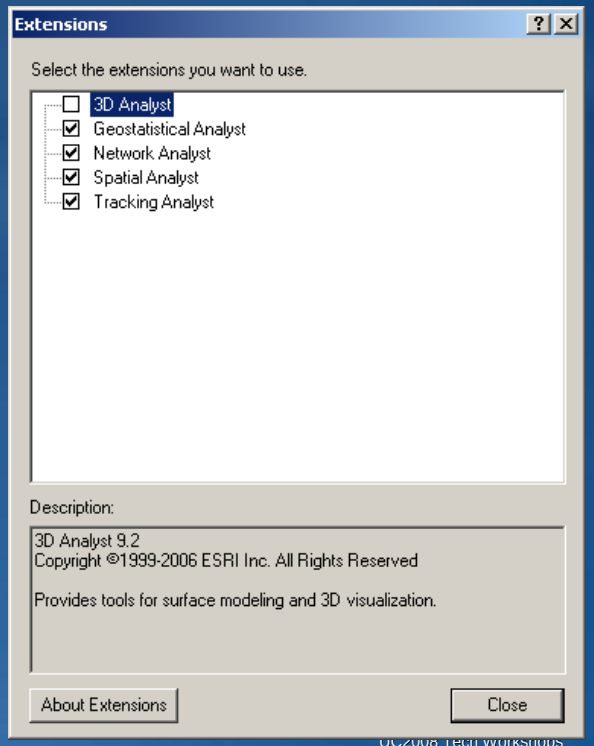
- Description appears at the bottom
  - Extension name and version
  - Copyright and company name
  - Extension purpose



Extension interfaces

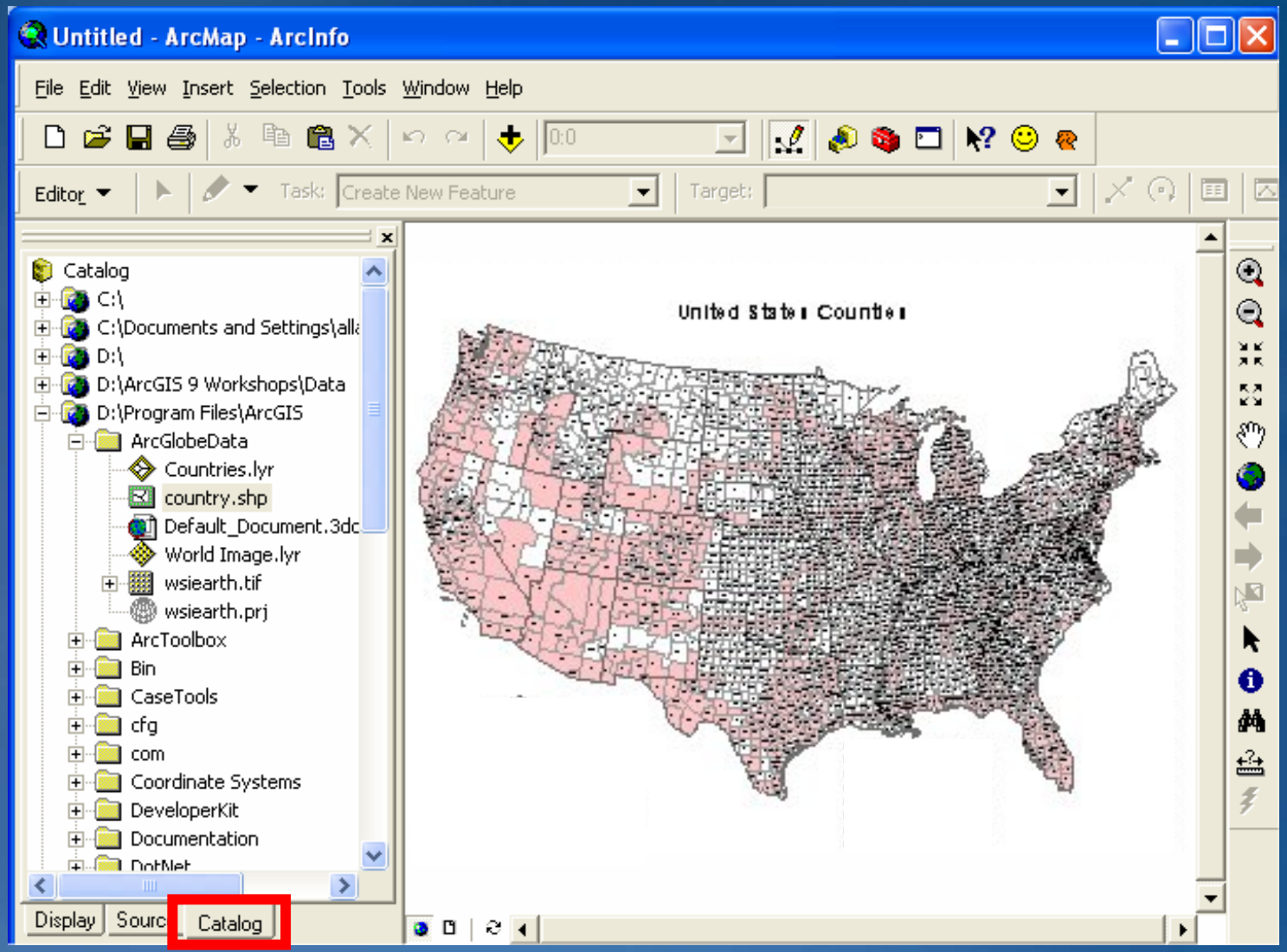
- IExtensionConfig is required
  - Can be like a silent, invisible extension, and won't appear in list
  - Load data
  - Check status
- IExtension is optional
  - Adds extension to the Extension Manager window's list
  - Could load your toolbar and commands





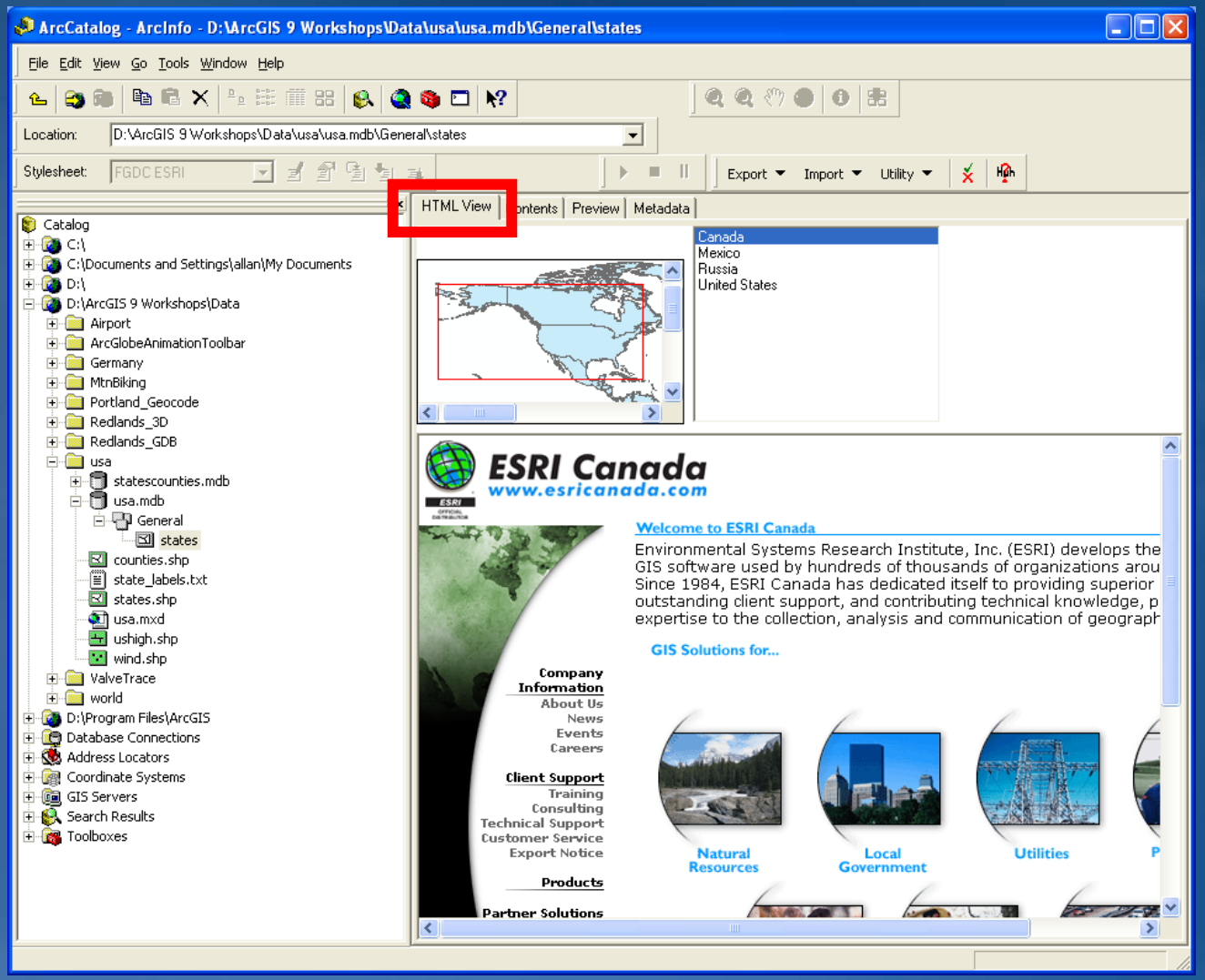
Add table of contents tabs

- Implement IContentsView

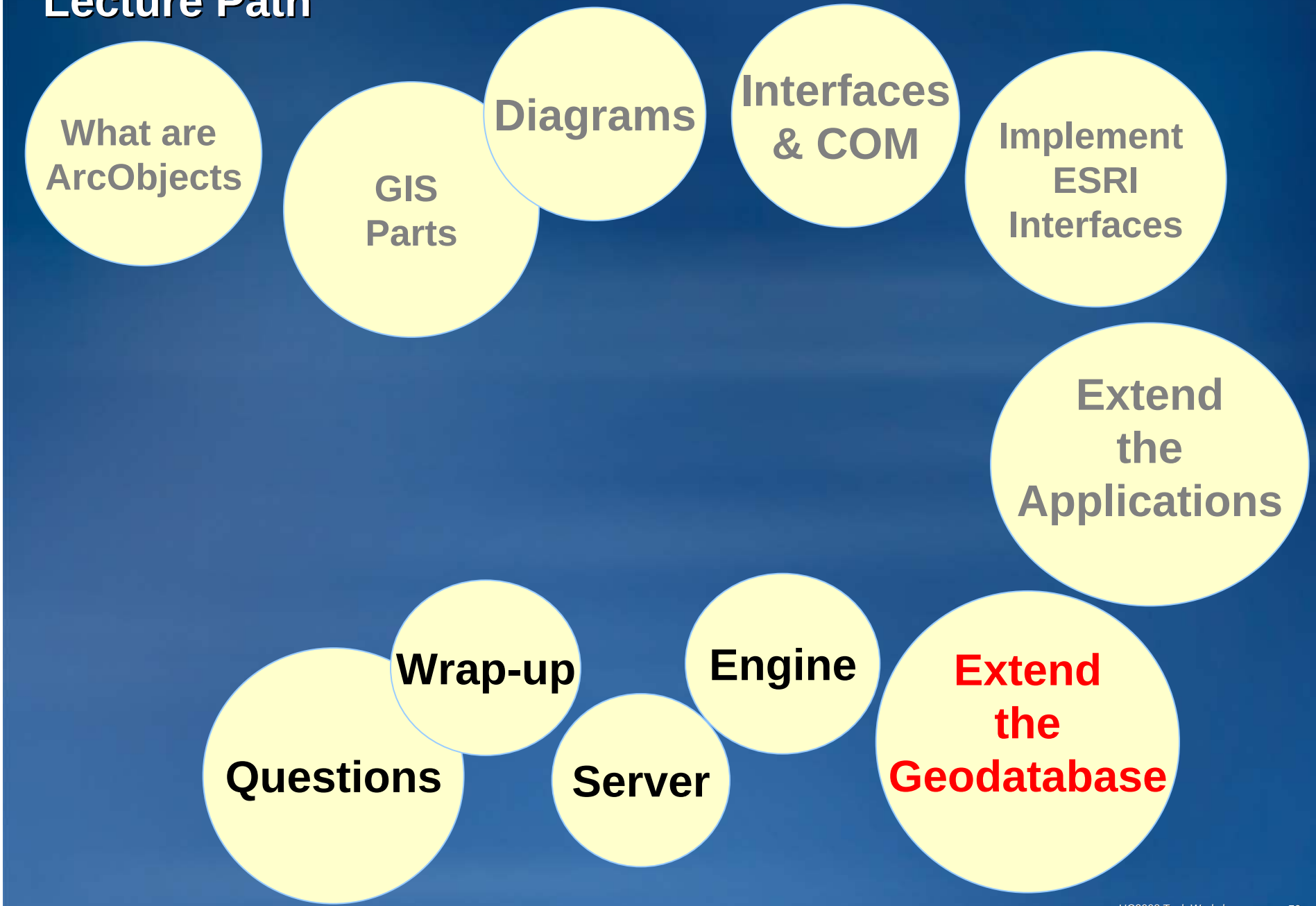


Add ArcCatalog tabs

- Implement IGxView

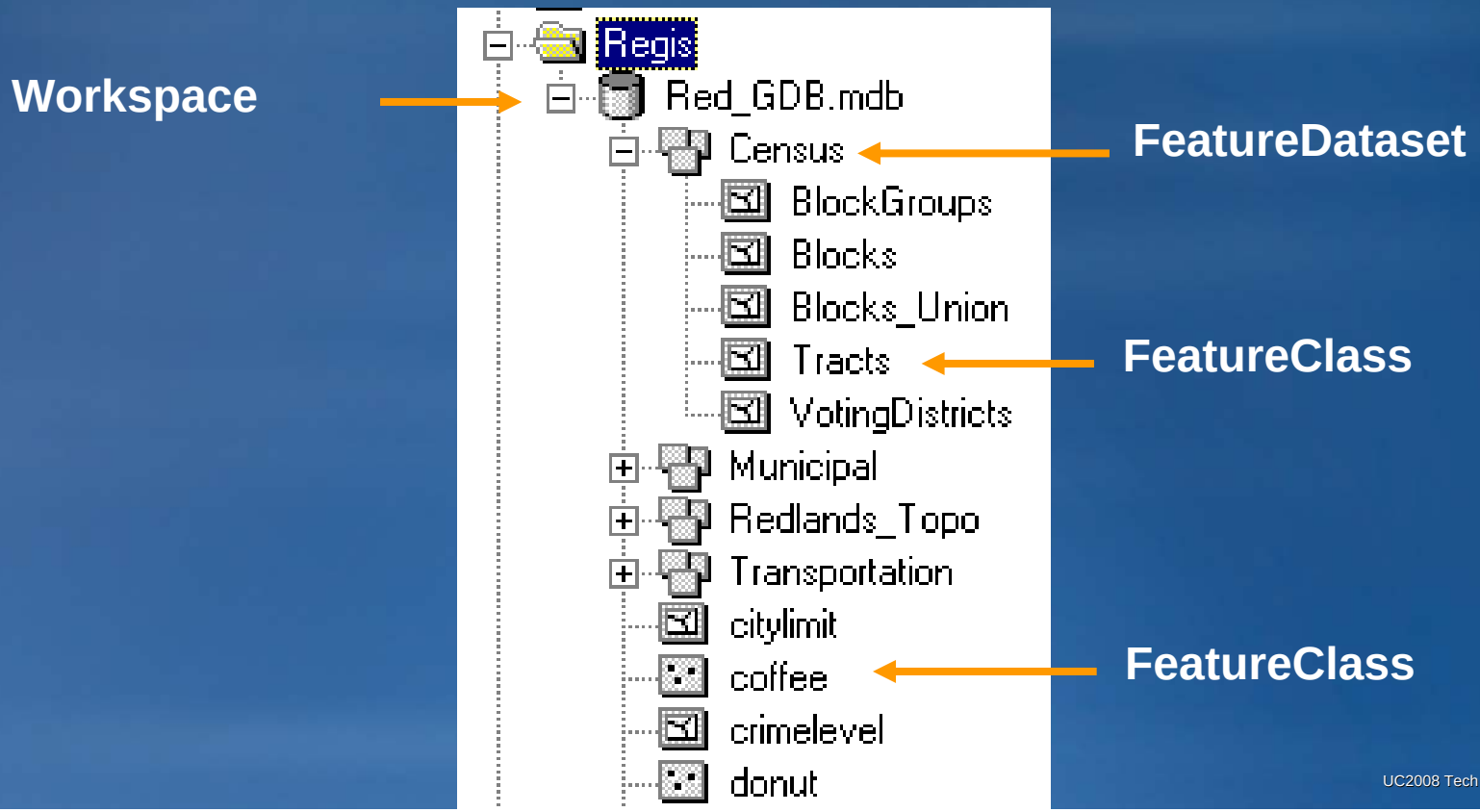


Lecture Path



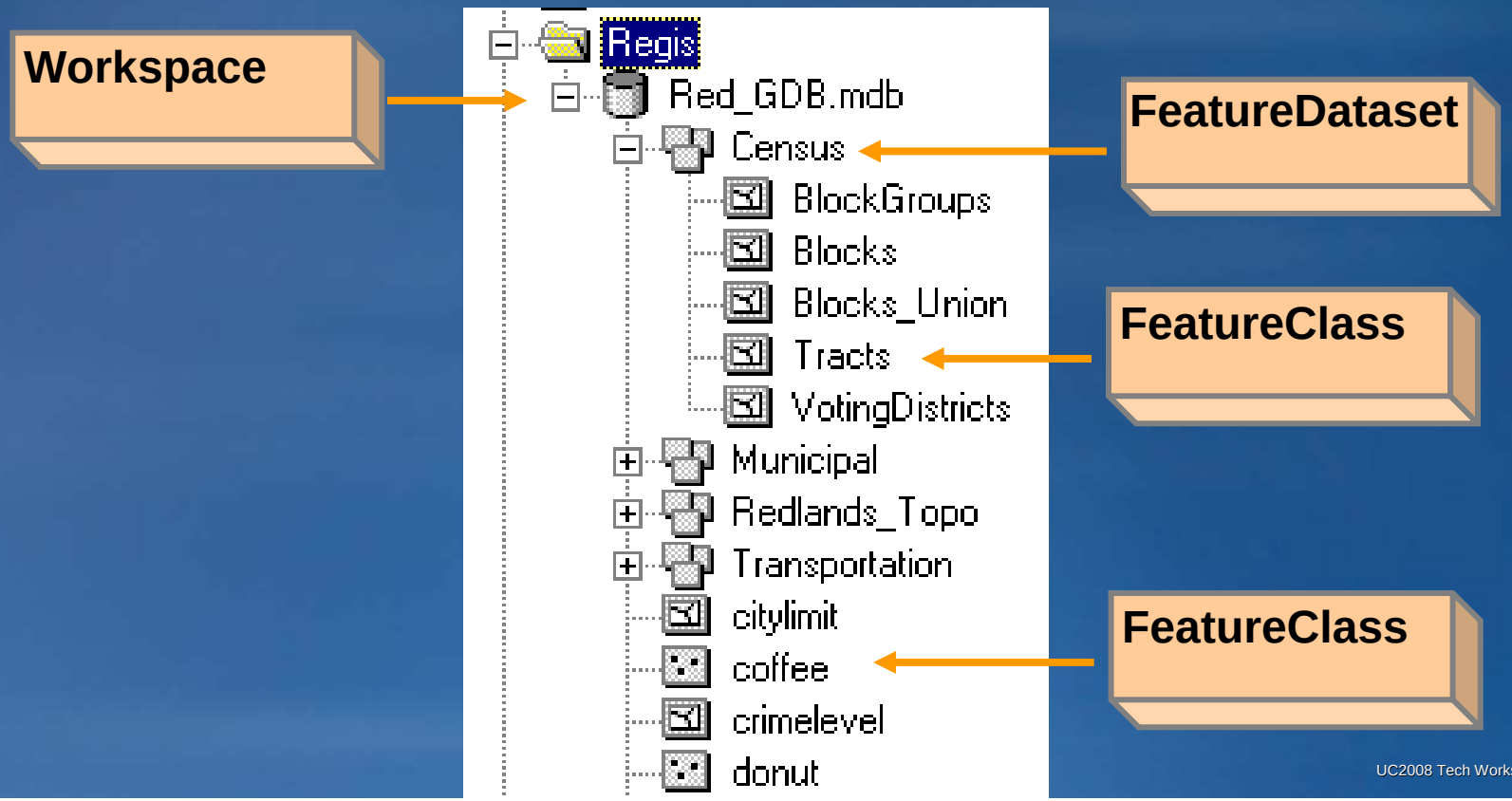
Geodatabse objects match your experience as a user

- Programmers start the same place users start

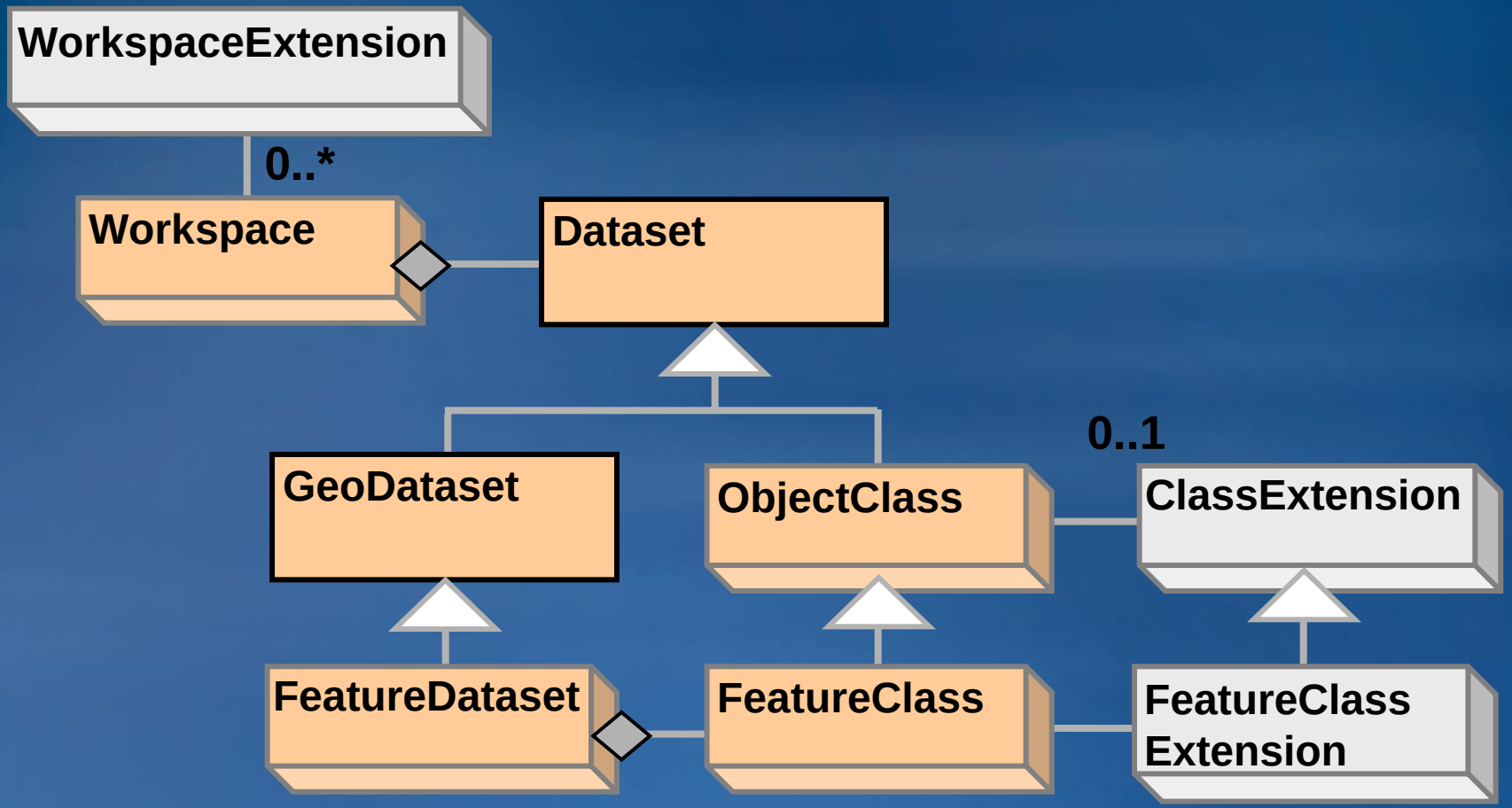


API objects match your experience as a user

- Programmers start the same place users start
- Geodatabase API uses the same (user) terms
- Includes any data, not just geodatabase formats



Extendable objects



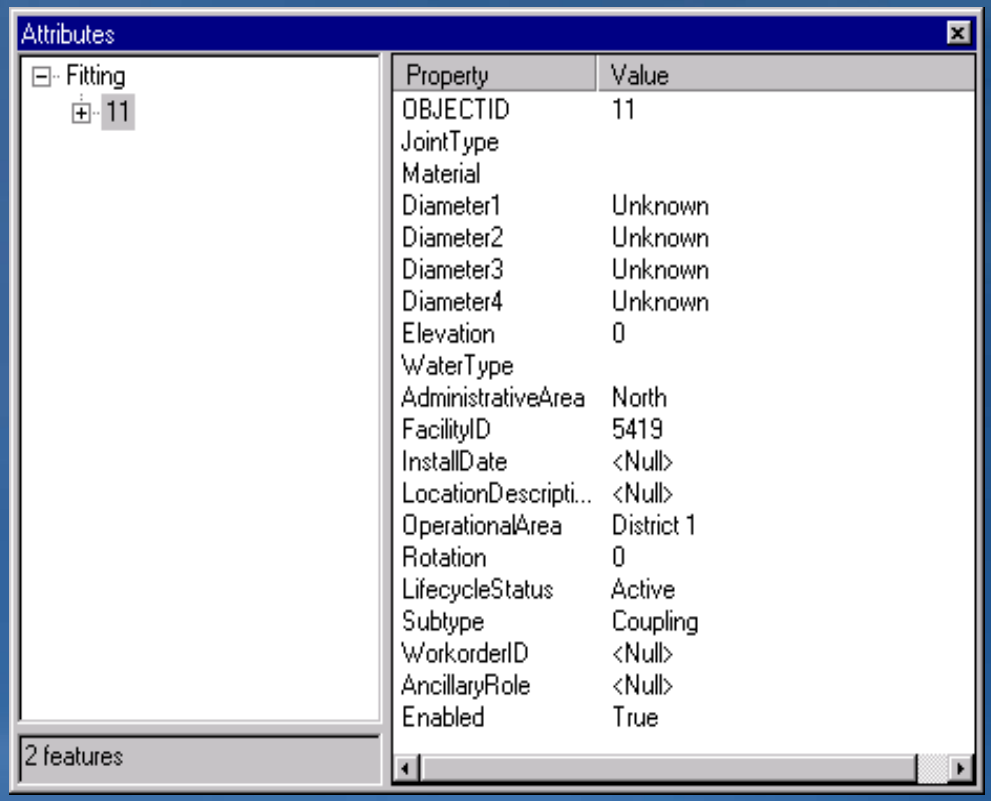
Class extension uses

- Schema generation
- Custom drawing
- Custom property inspection (next)
- Validation
- Custom split policies
- Related object creation notification

Class extension example

- ESRI's Attribute inspector
- From the Editor toolbar

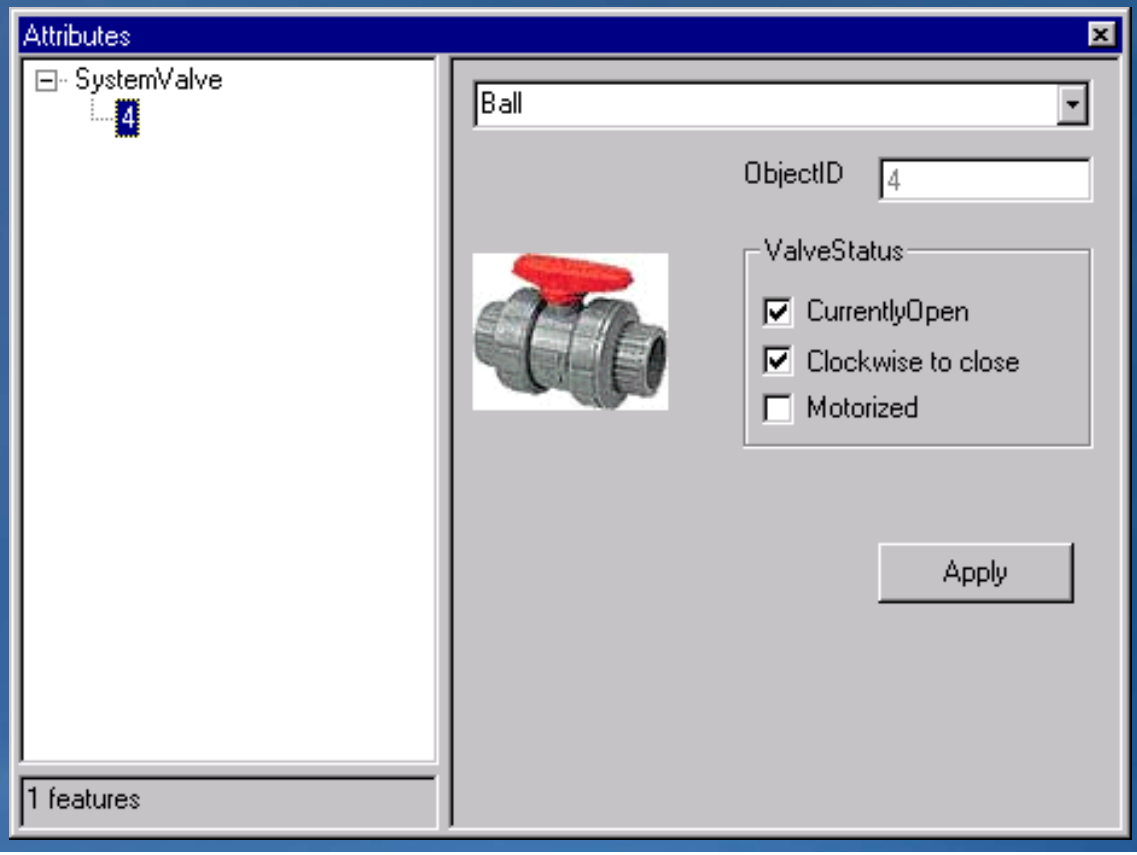




Custom inspector

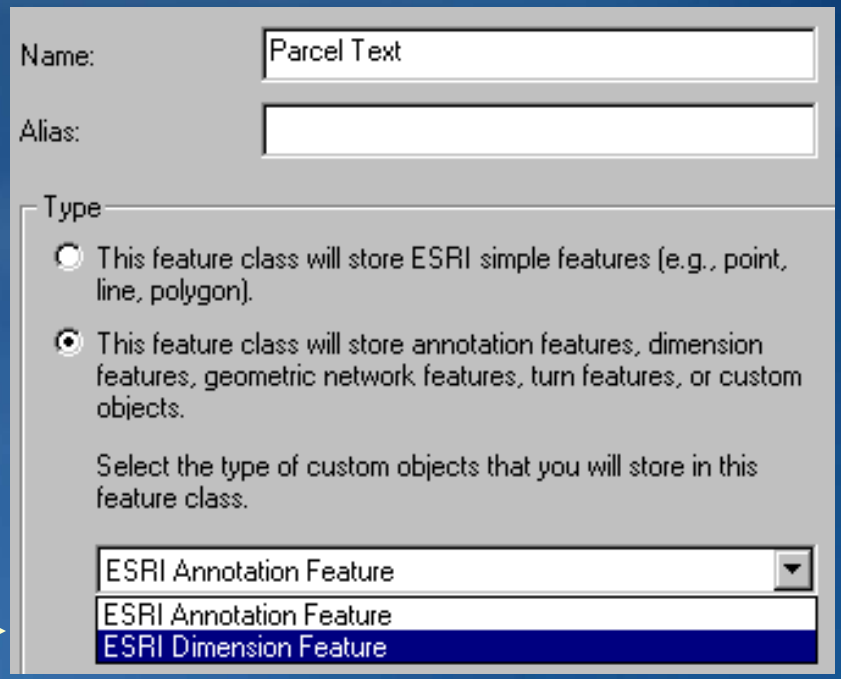
- Extend behavior of an entire feature class
- Your window appears instead of ESRI's
- Behavior stored with the data, not in an mxd file





Make your own custom feature classes

- IFeatureClassDraw
  - Override a feature class's drawing for any client
  - You can make a custom renderer and property page for it
- Control new class creation
  - User can create your feature class

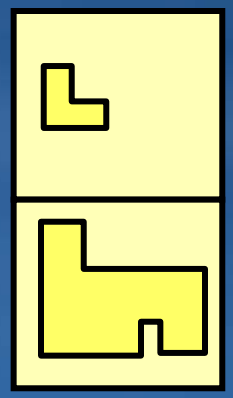


Your description here
(In ArcCatalog)

Behavior interfaces: Listen and React

- IObjectClassValidation

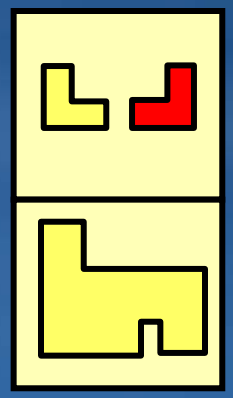
Validation event:
Building height
must be 10 times
the number of stories



Behavior interfaces: Listen and React

- IObjectClassValidation
- IRelatedObjectClassEvents
- IConfimSendRelatedObjectEvents

Validation event:
Building height
must be 10 times
the number of stories

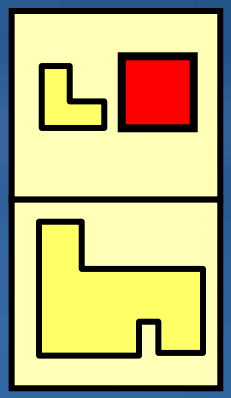


As a building is added, notify related parcel to update its structure count

Behavior interfaces: Listen and React

- IObjectClassValidation
- IRelatedObjectClassEvents
- IConfimSendRelatedObjectEvents
- IObjectClassEvents *(OnCreate, OnDelete, OnChange)*

Validation event:
Building height must be 10 times the number of stories



OnModify:
Record current time and user name in the table

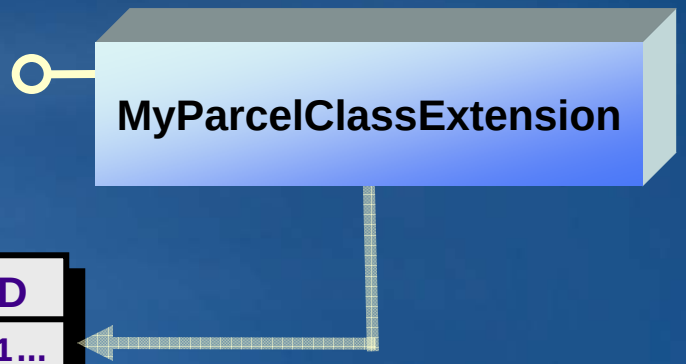
As a building is added, notify related parcel to update its structure count

Registering class extensions with the geodatabase

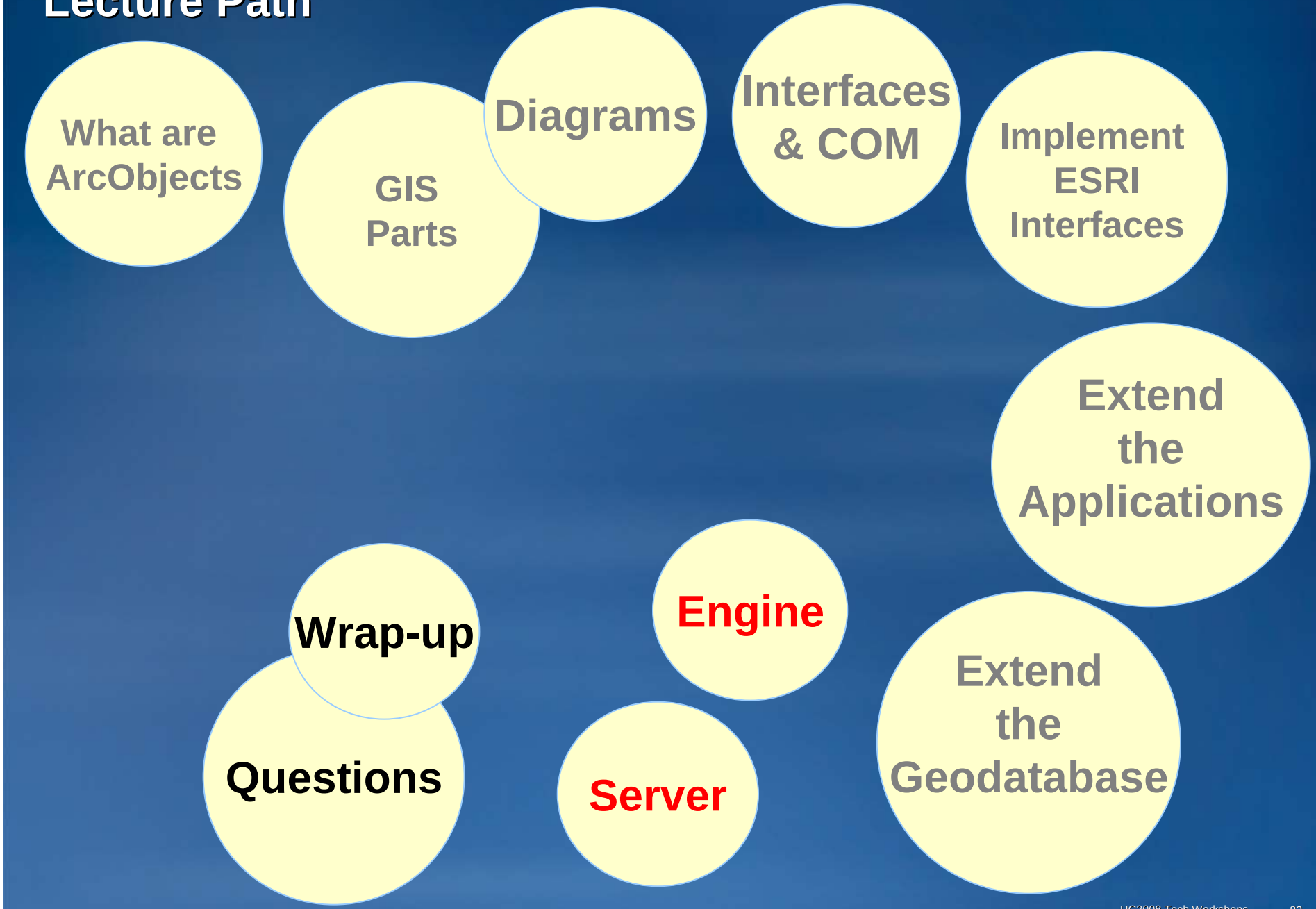
- Limit of one class extension per class
- Class extension's GUID must be registered on all client machines accessing data
- Methods to register extension:
  - During creation: IFeatureWorkspace CreateFeatureClass
  - After creation: IClassSchemaEdit AlterClassExtensionCLSID
  - When modeling and designing in Visio

The object classes table within a geodatabase

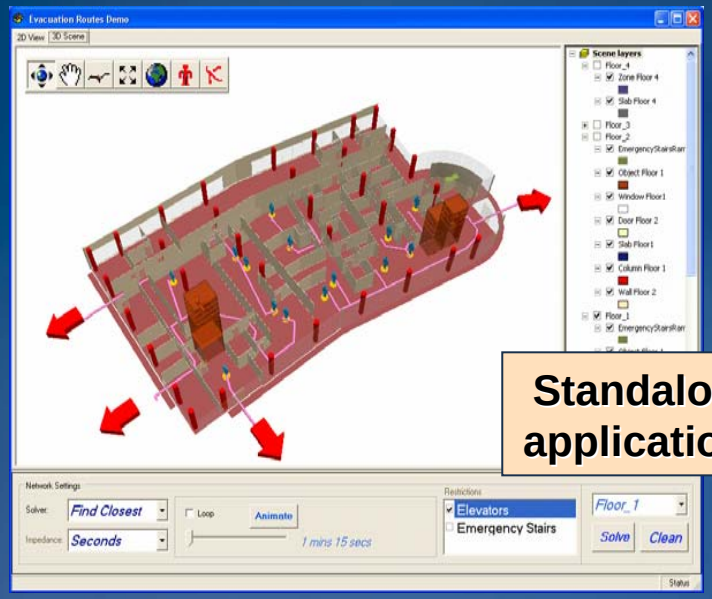
ID	Name	CLSID	EXTCLSID
1	Parcels	{3070721…	{0368CF51…
2	Buildings	{3070721…	
3	Owners	{3070721…	{044782D…



Lecture Path



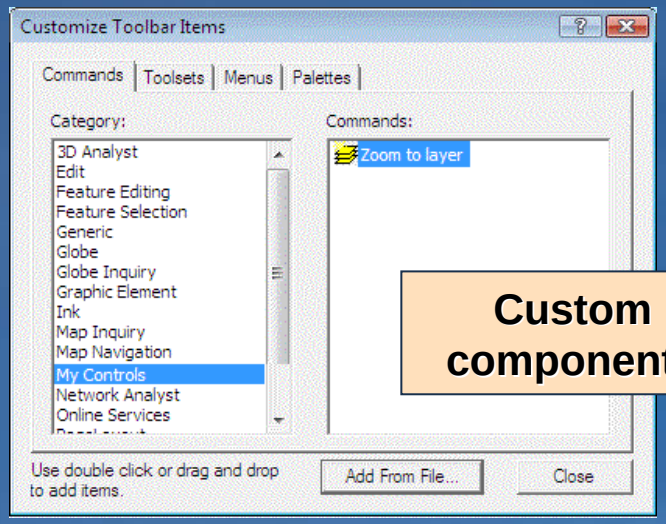
What you can do with ArcGIS Engine?



Standalone applications

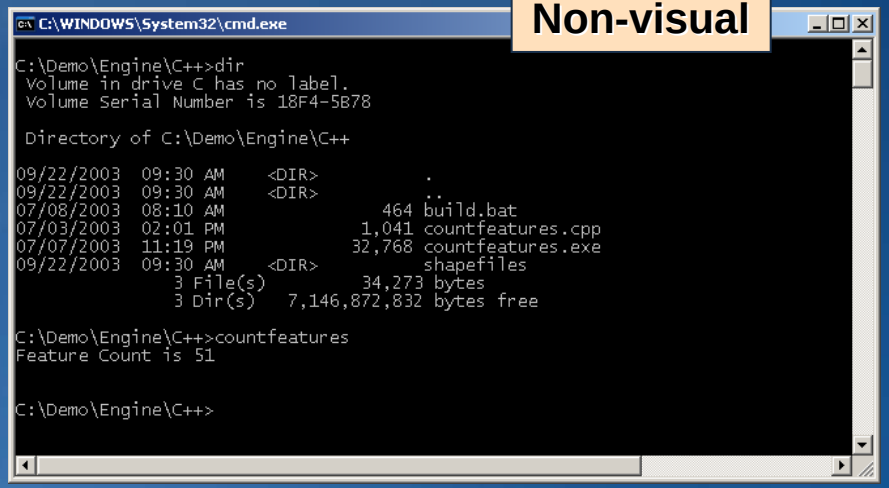


Embedded applications



Custom components

Non-visual

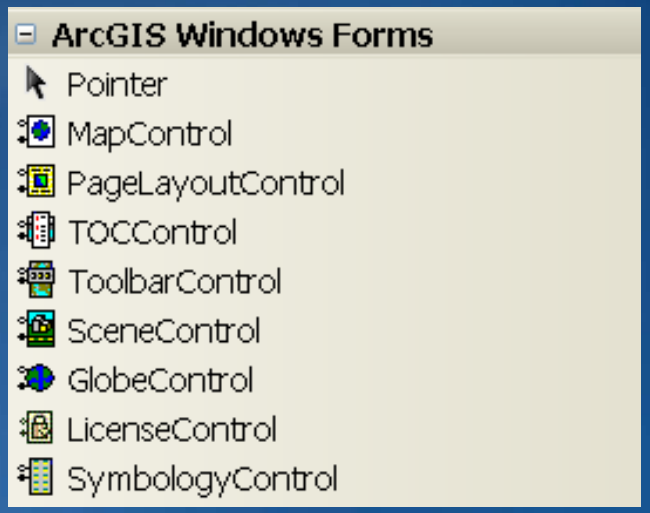


ArcGIS Engine: Product overview

- **Deployment:** ArcGIS Engine Runtime
  - Two levels of product licensing
    - Standard Runtime = ArcView
    - Geodatabase Update = ArcEditor
  - Extensions

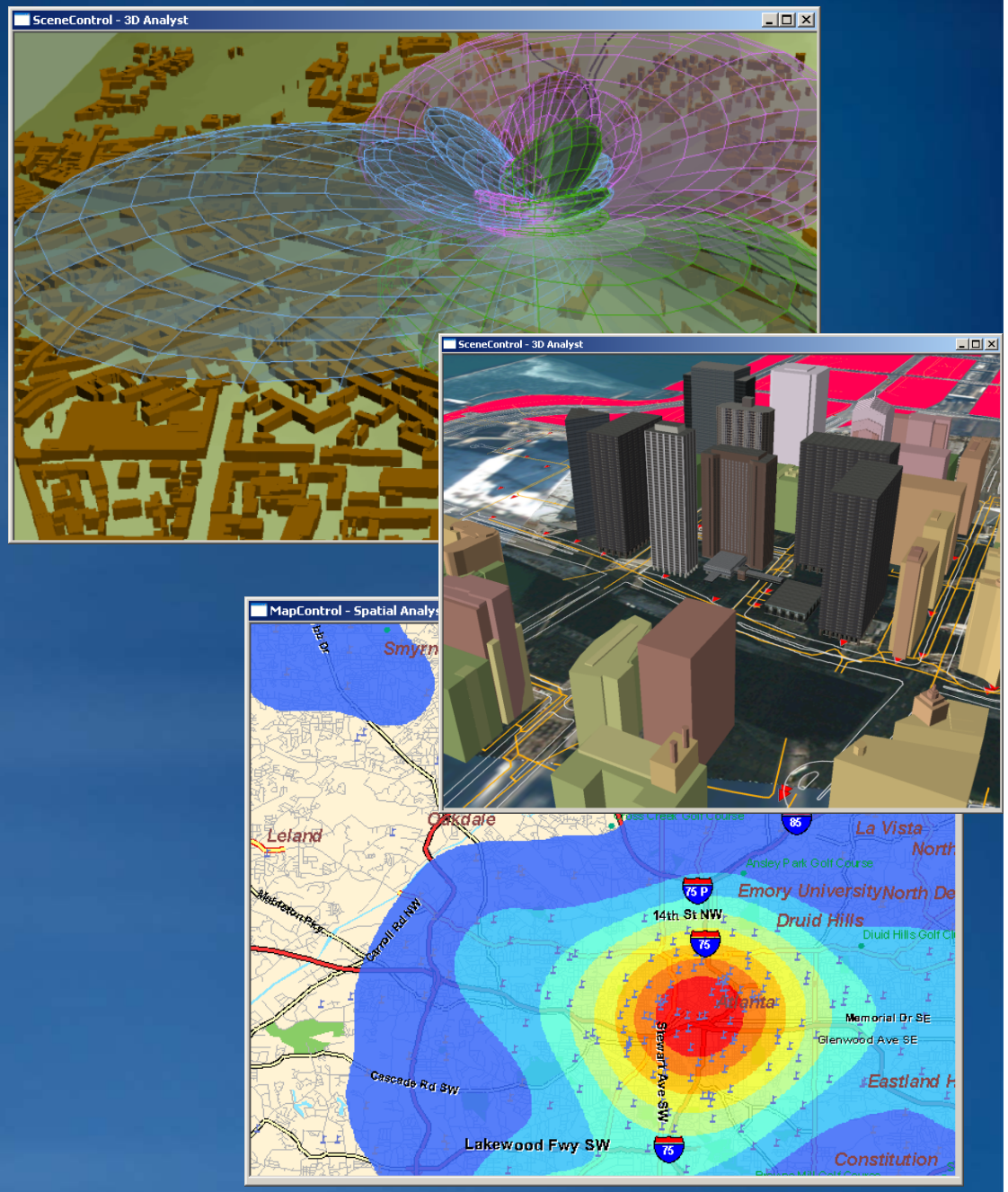
Windows, Solaris, Linux (Intel)

- **Development:** ArcGIS Engine Developer kit
  - ArcGIS Engine Runtime
  - Developer help system and samples
  - APIs for COM, .NET, Java, C++
  - Visual development components
  - IDE integration for Visual Studio .NET



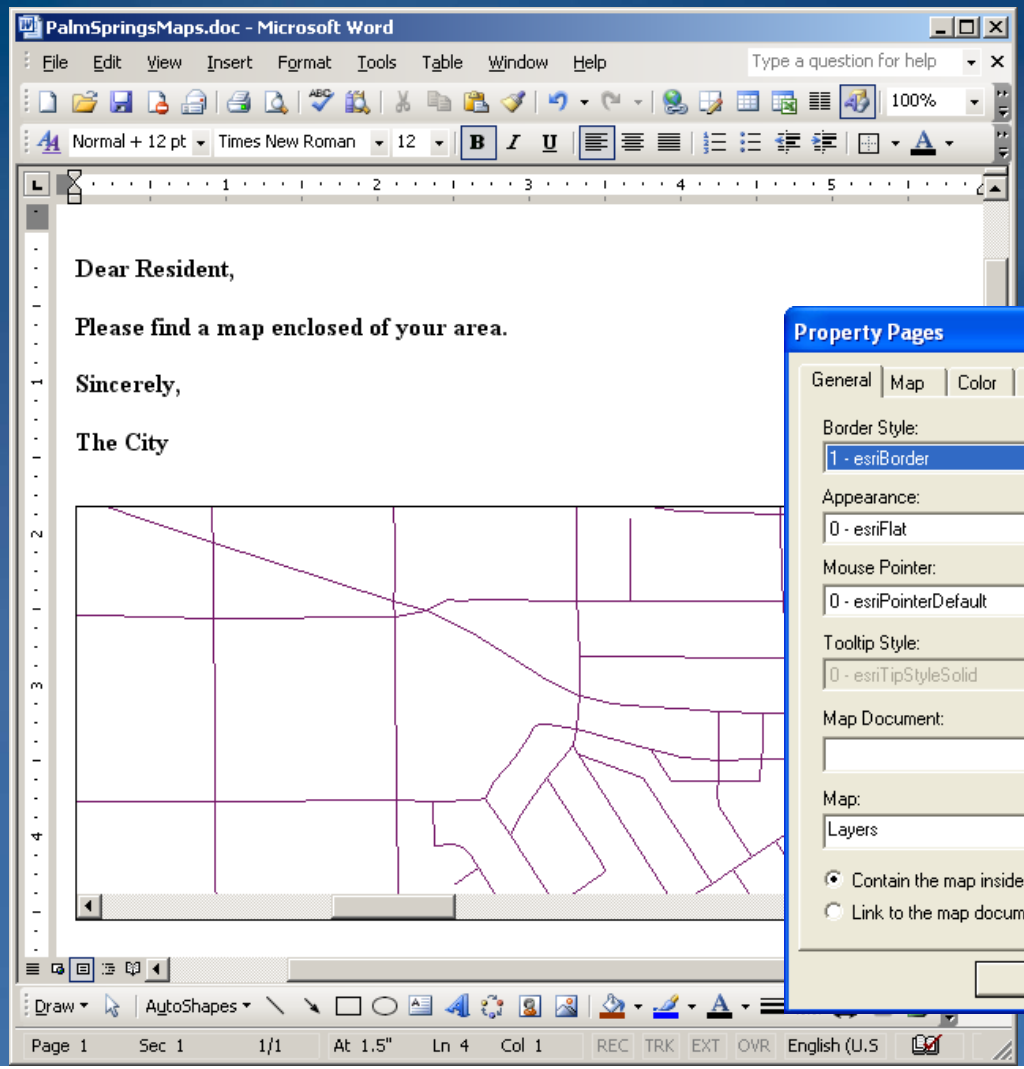
Developer Controls

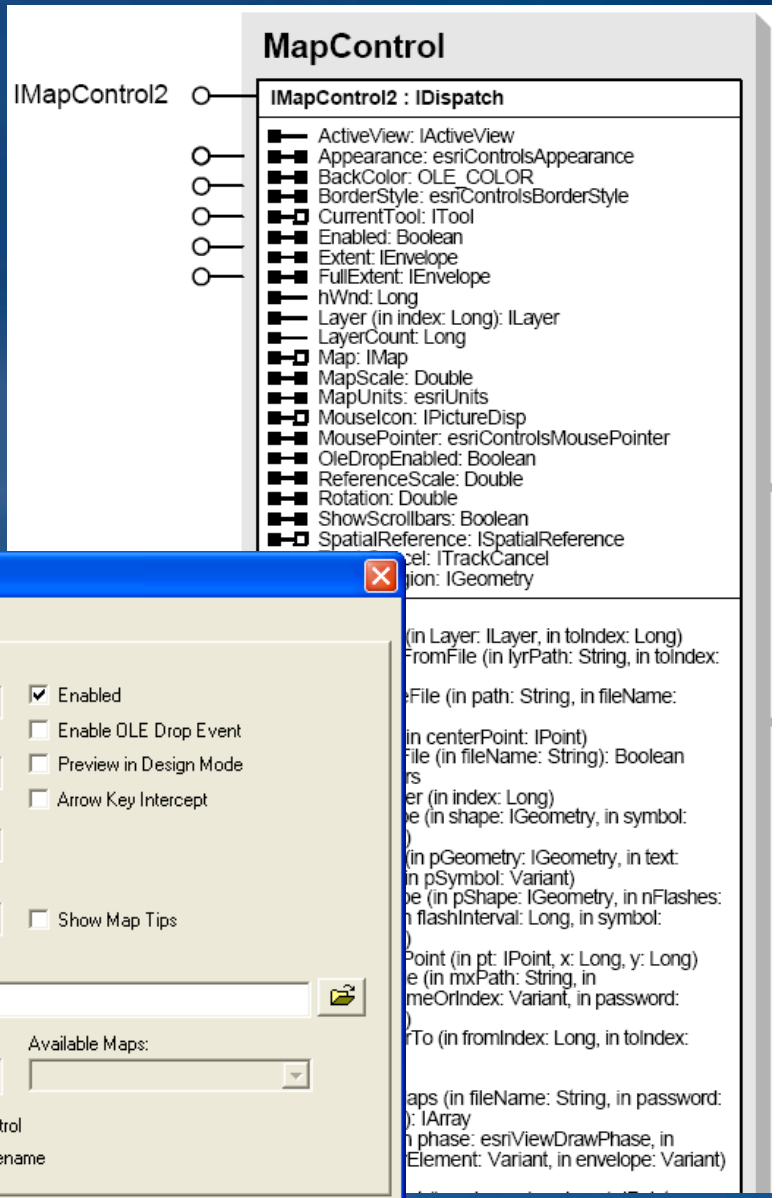
- MapControl
- PageLayoutControl
- ToolbarControl
- TOCControl
- ReaderControl
- SceneControl
- GlobeControl



The Map Control

- Put a map control in a Word document other applications

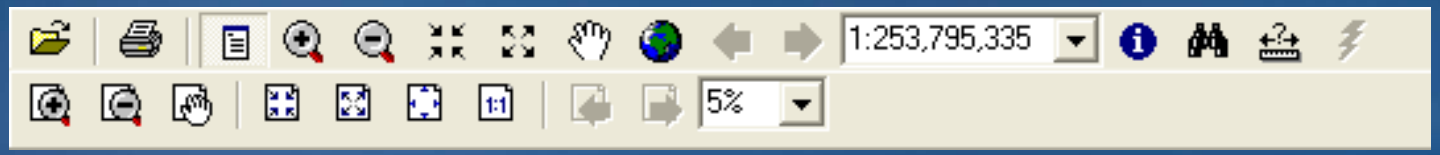




Engine Tools and Commands

More than 100 tools and commands included





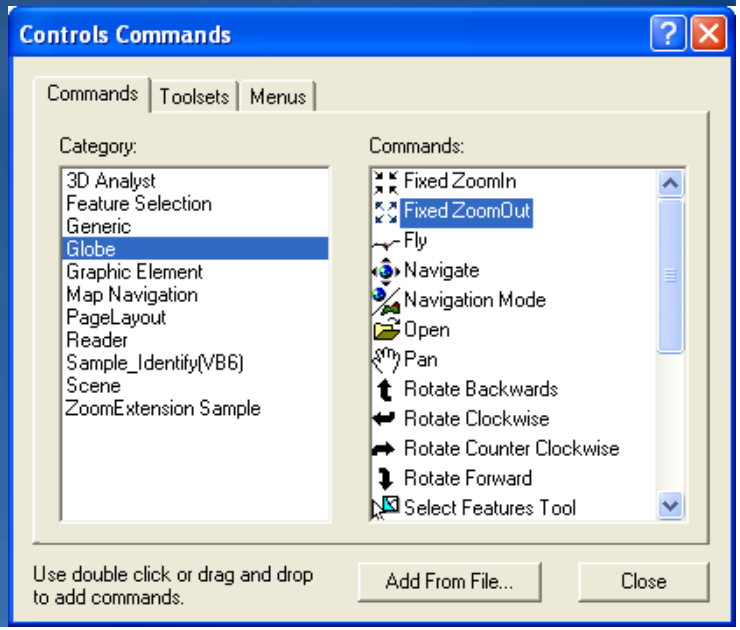






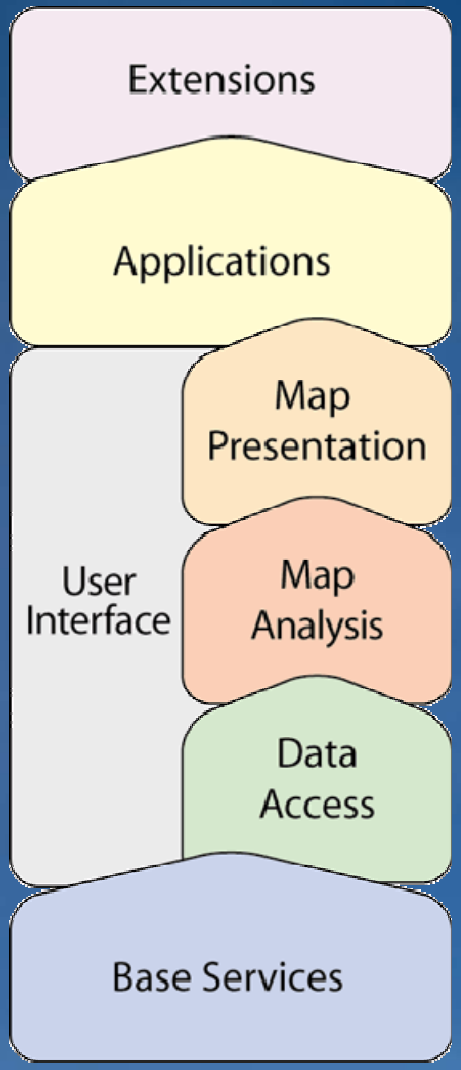




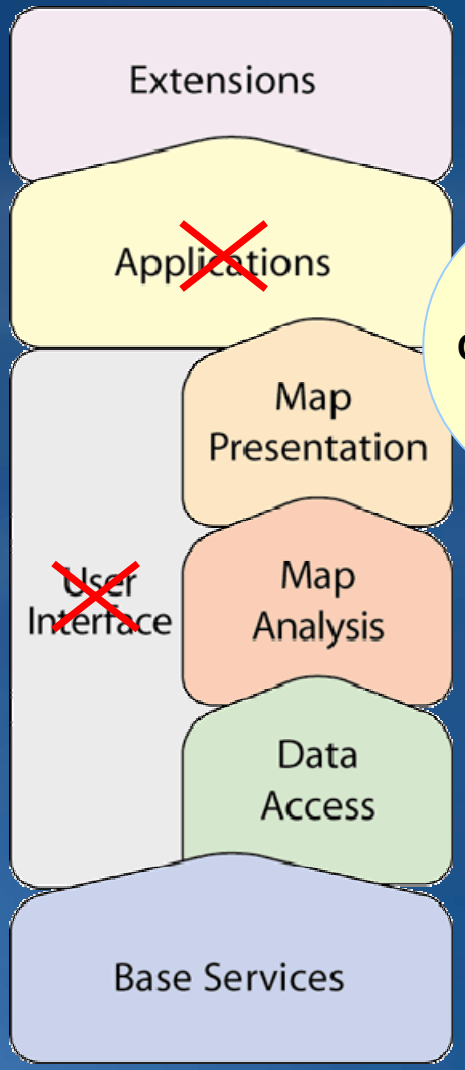


Engine – Desktop comparison

- Desktop ArcObjects libraries



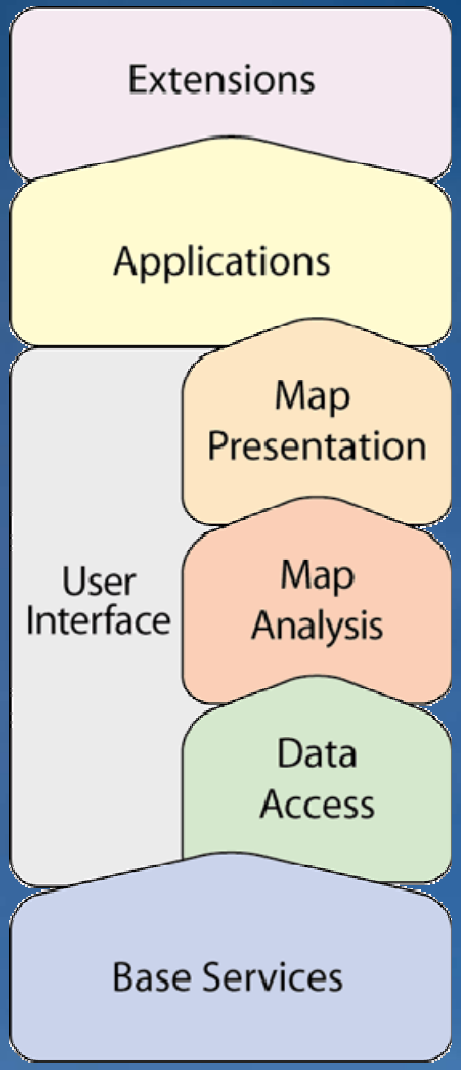
- Engine ArcObjects libraries



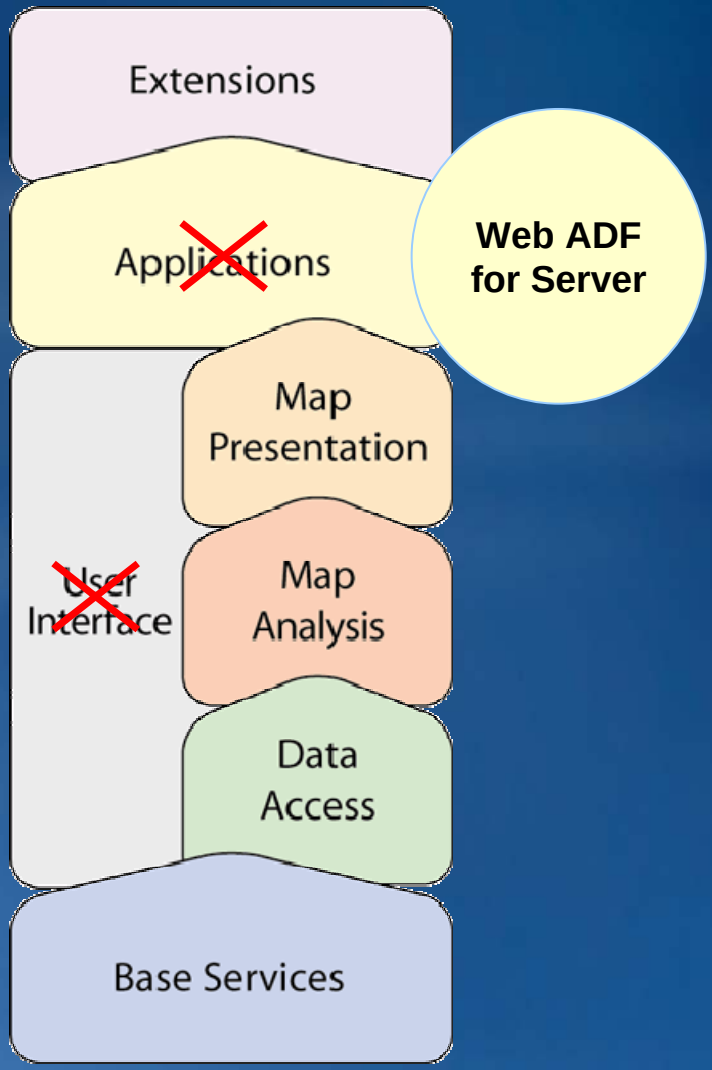
Developer Components for Engine

Server – Desktop comparison

- Desktop ArcObjects libraries



- Server ArcObjects libraries

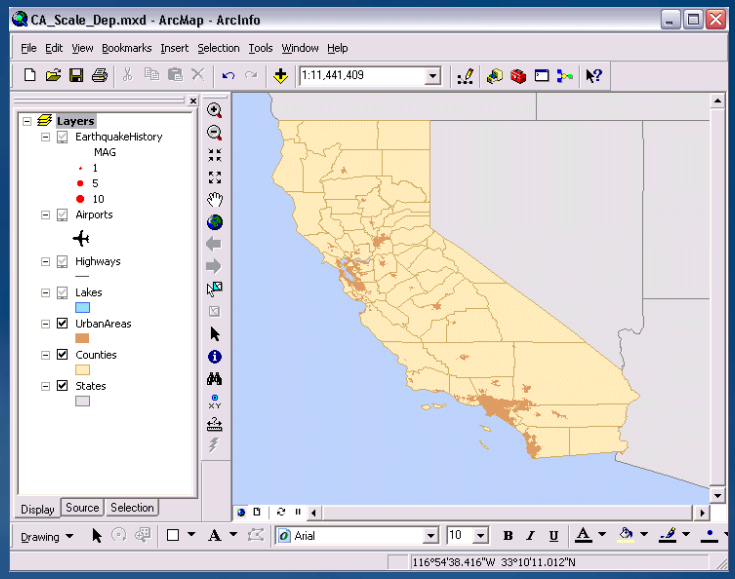


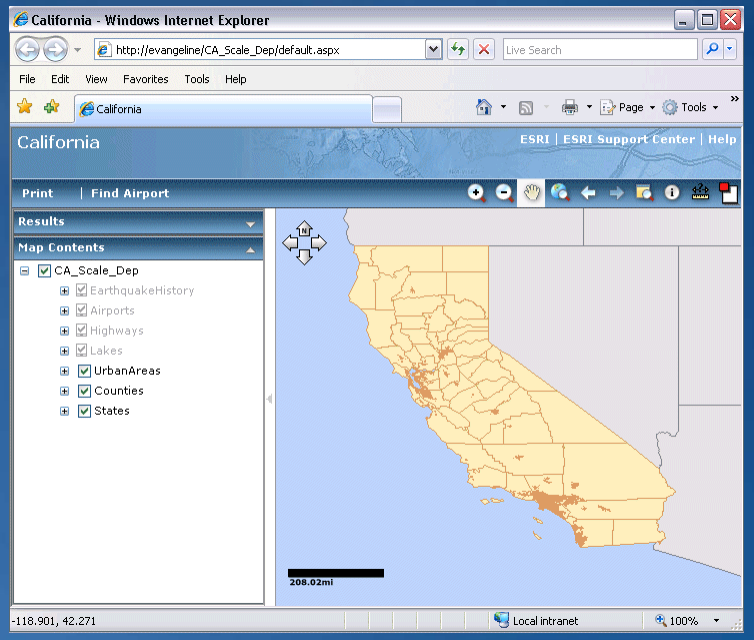
GIS on the Web in 3 steps

Author GIS content
- Create GIS resources
- ArcGIS Desktop applications

Serve content
- Publish GIS resources as services
- ArcCatalog and ArcGIS Server Manager

Use GIS services
- Web mapping applications
- ArcGIS Explorer
- ArcGIS Desktop
- Many others

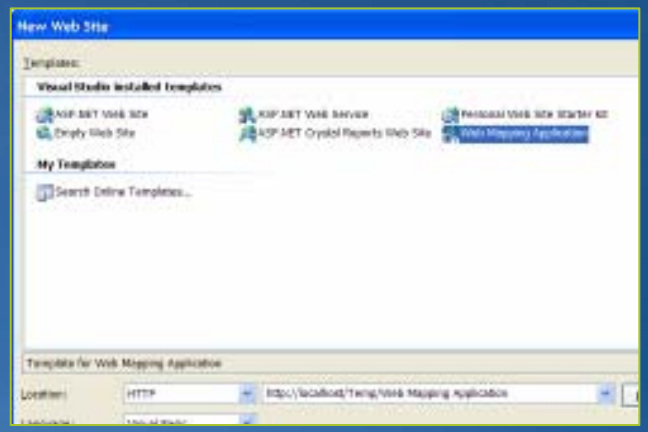




Creating Web ADF applications

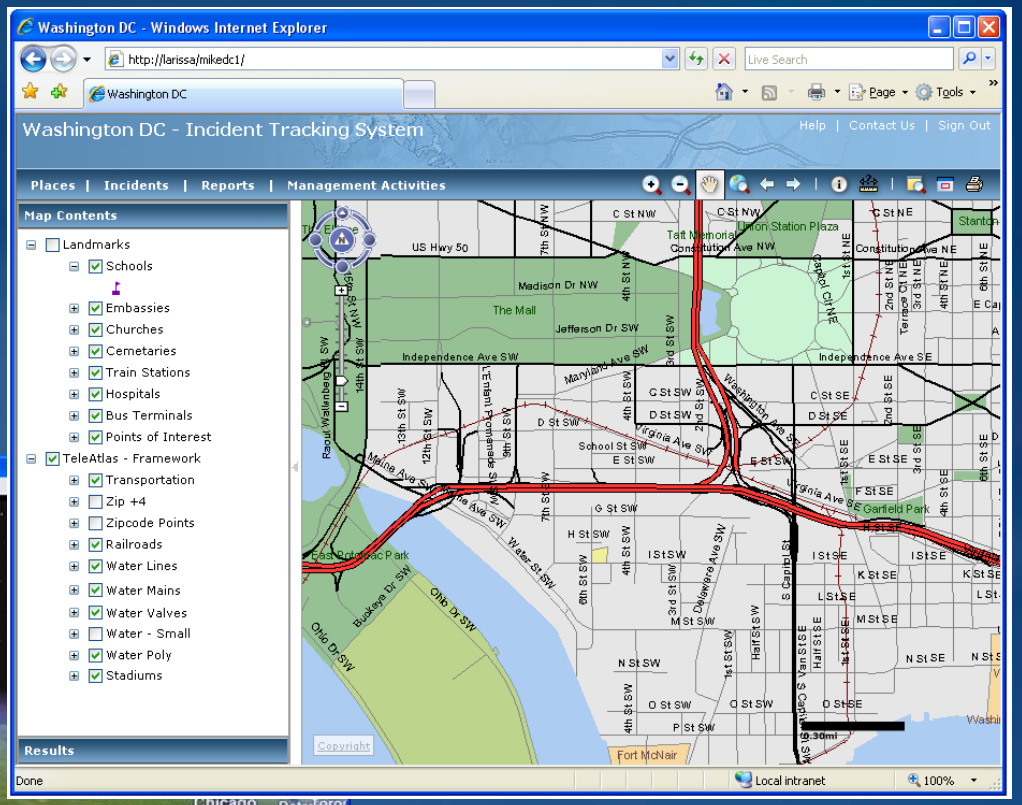
1. Use ArcGIS Server Manager
   - Web site builder
   - Modify in Development IDE
2. Use a template
   - Same template used by ArcGIS Server Manager
   - Visual Studio, Eclipse
3. Create using Web controls

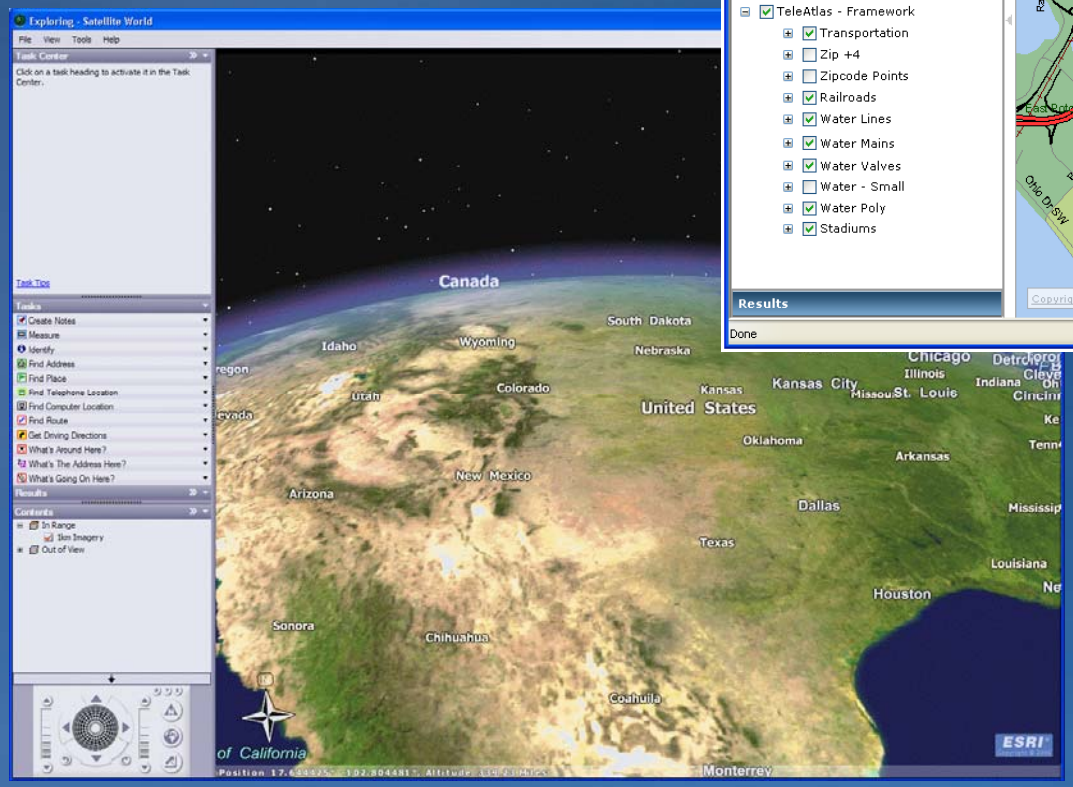




Types of GIS services

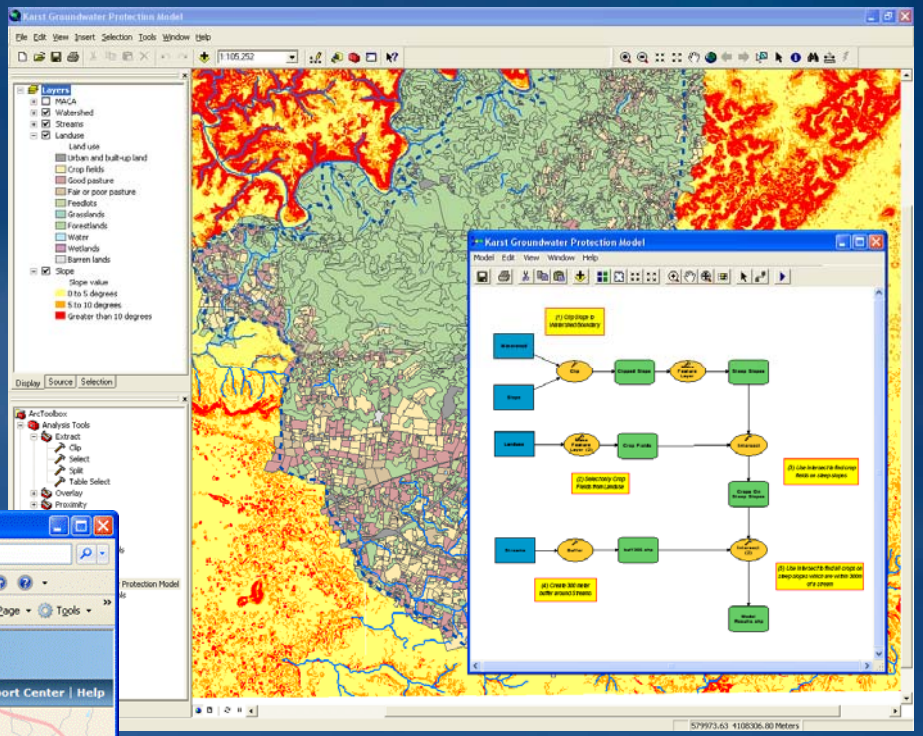
- Visualization
  - –Map services
  - –Image services
  - –Globe services

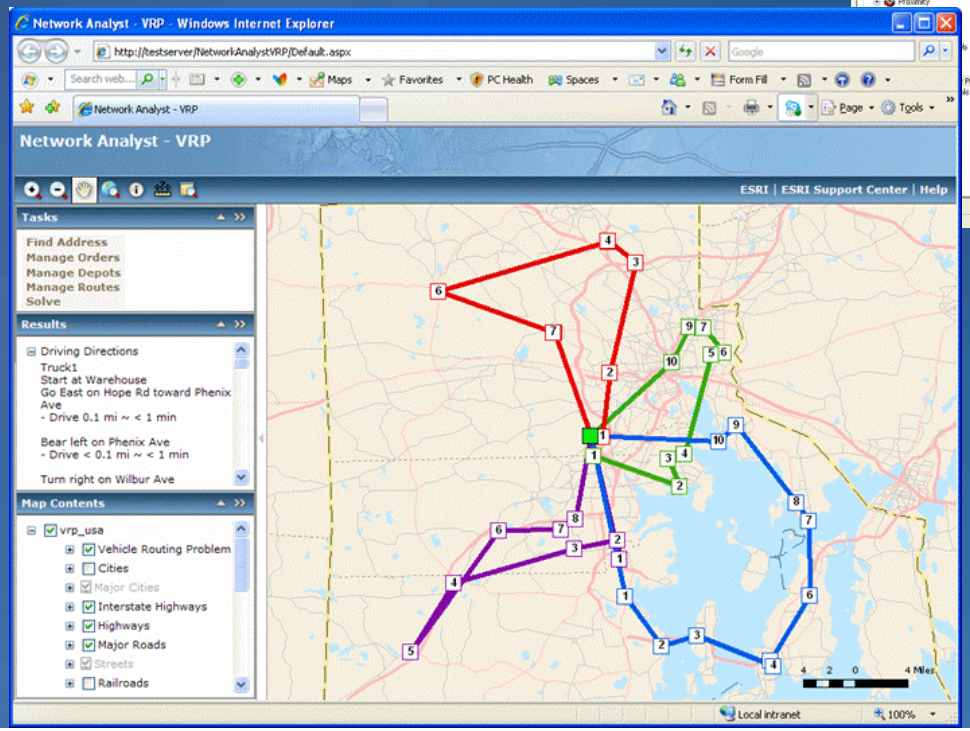




Types of GIS services (cont')

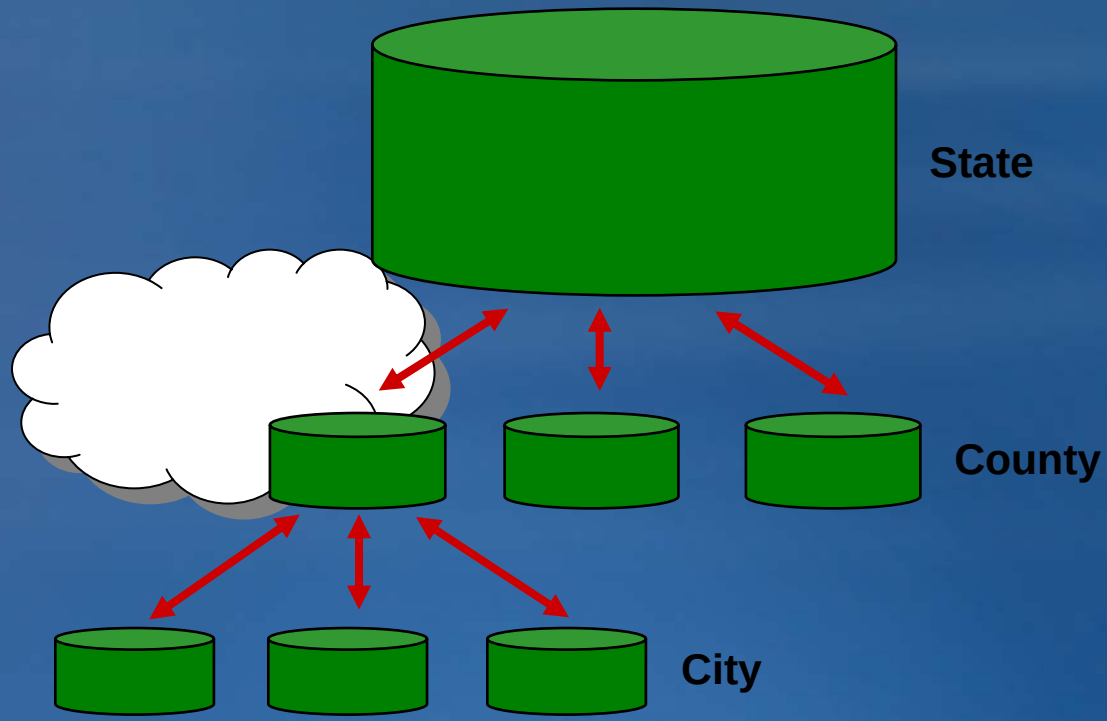
- Geographic analysis
  - –Geoprocessing services
  - –Geocoding services
  - –Network analysis services
    - Requires Network Analyst extension



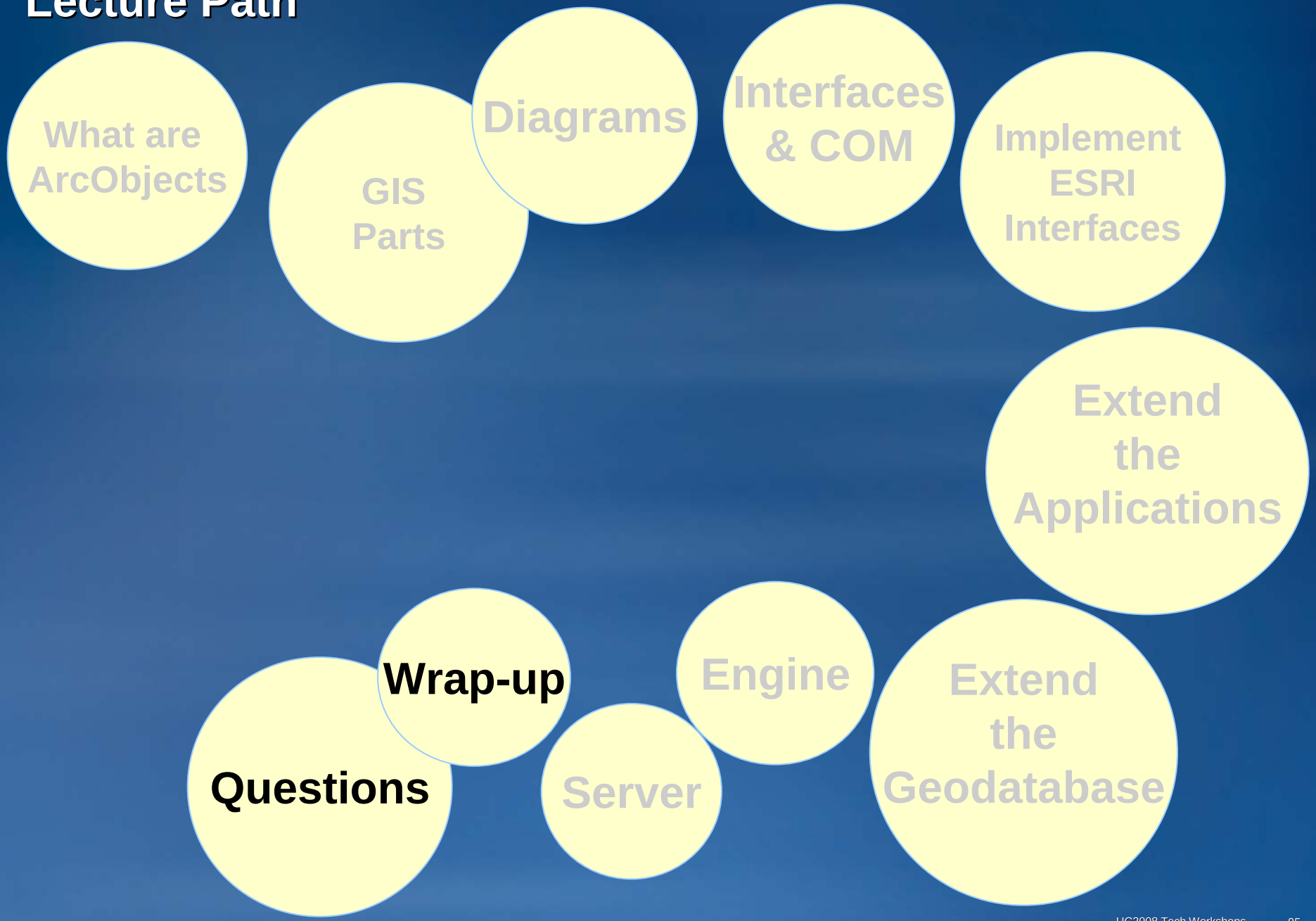


Types of GIS services (cont')

- Data access
  - Geodata services
    - Access, copy, edit, and synchronize data over the Web

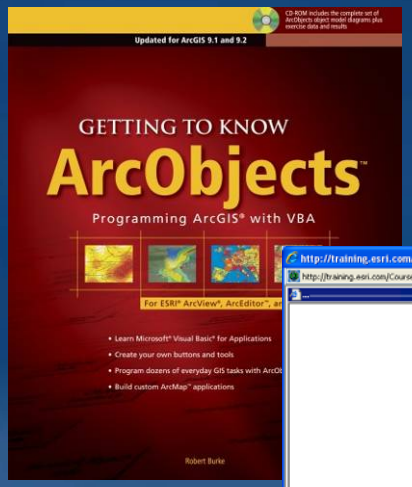


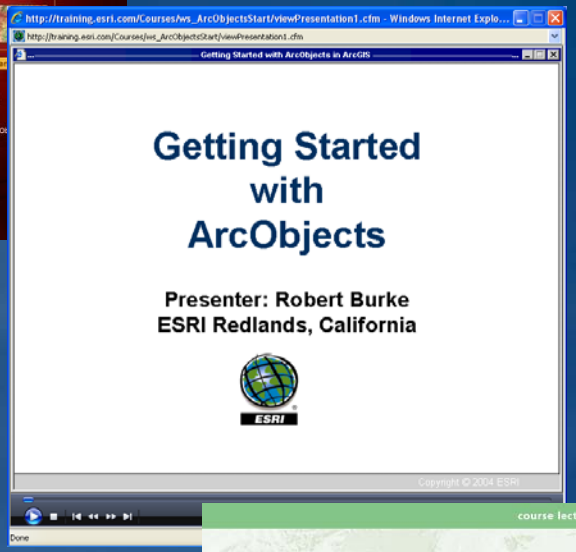
Lecture Path

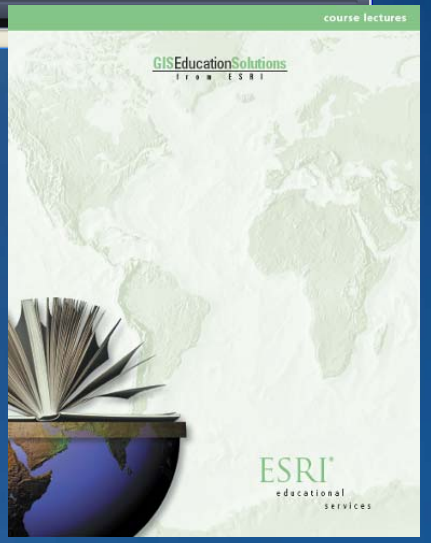


Books and training classes

- Book
  - Getting to Know ArcObjects
  - Book sales at the Spatial Outlet
- Live training seminar recordings
  - http://training.esri.com/Courses/ws_ArcObjectsStart/index.cfm?
- Instructor-led Training
  - Introduction to ArcObjects
    - VBA, .NET, or Java
  - Extending ArcGIS Desktop Applications
  - Developing applications with ArcGIS Engine
  - Introduction to ArcGIS Server
  - ArcGIS Server Administration
  - Developing ArcGIS Server Applications
    - JAVA or .NET

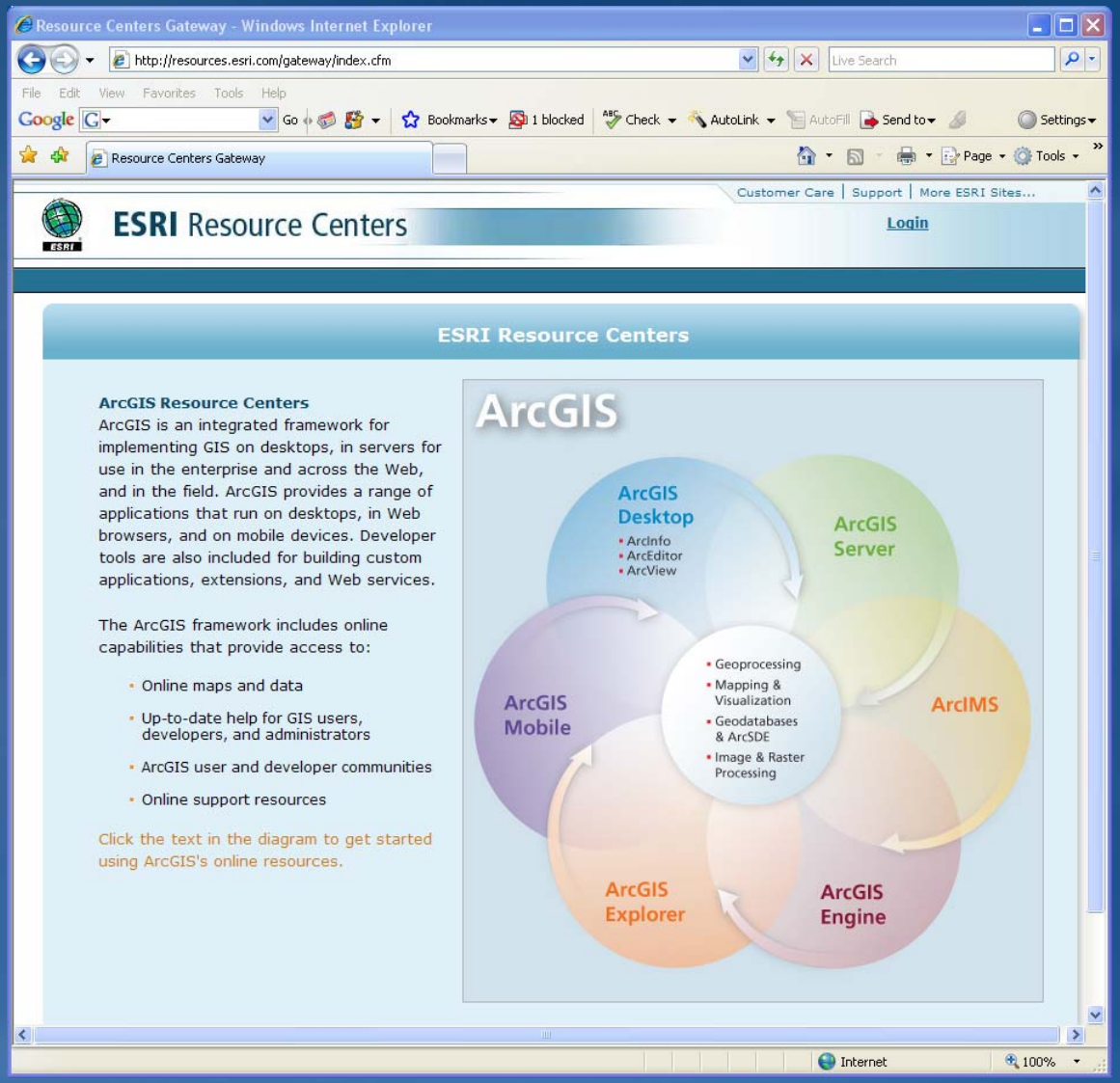






Online Help

- http://resources.esri.com



Online Help

- **http://blogs.esri.com/Dev/blogs/arcobjectsdevelopment/default.aspx**



Surveys

Please fill out a survey

Questions

Thank you for attending

- **I hope to see you in a class again soon**

