Case Study: Microsoft COM

Outline

- Goals of COM
- The Common Object Model
- Microsoft Interface Definition Language
- COM Architecture
  - Presentation Layer Implementation
  - Session Layer Implementation

Goals of COM

- ***Provide a component object model that facilitates binary encapsulation and binary compatibility***
  - ***Binary encapsulation: Clients do not have to be re-compiled if server objects change***
  - ***Binary compatibility: Client and server objects can be developed with different development environments and in different languages***
- ***COM is proprietary de-facto standard***

COM Object Model

- ***Interfaces***
- ***Implementations and Objects***
- ***Classes***
- ***Attributes***
- ***Operations***
- ***Requests***
- ***HRESULTS***
- ***Inheritance***

Microsoft IDL (MIDL)

- *Language for expressing all COM concepts*
- *MIDL is*
  - *programming-language independent*
  - *evolved from OSF/RPC IDL*
  - *not computationally complete*
- *Different programming language bindings are available*
- *Explanation of Model and Language by same example*

COM Interfaces

UUID

```
[object,
 uuid(1CF2B120-547D-101B-8E65-08002B2BD118)]
interface IOrganization : IUnknown {
  ...
};
[object,
 uuid(1CF2B120-547D-101B-8E65-08002B2BD116)]
interface IClub : IOrganization {
  ...
};
```

Root Interface

Interfaces

Interface Inheritance

COM Implementations

- *Implement Interface in Prog. Lang., e.g. C++*

```
#include "Soccer.h"
class Player : public IPlayer {
 private:
  char* name;
  short Number;
 protected:
  virtual ~TrainerPlayer(void);
 public:
  TrainerPlayer(void);
  IMPLEMENT_UNKNOWN(TrainerPlayer)
  BEGIN_INTERFACE_TABLE(TrainerPlayer)
  IMPLEMENTS_INTERFACE(ITrainer)
  IMPLEMENTS_INTERFACE(IPlayer)
  END_INTERFACE_TABLE(TrainerPlayer)
  void book(); // IPlayer methods
};
```

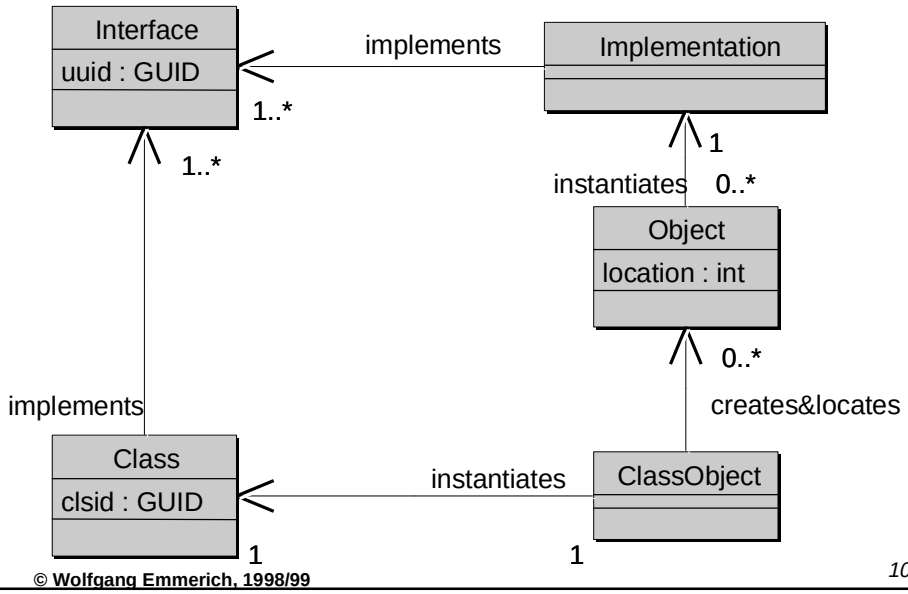

COM: Objects

- *Instances of COM Implementations*
- *References to COM objects are called interface pointers*
- *Interface pointers refer to main memory locations*
- *References support location transparency*
- *Object references are persistent*

COM Classes

- *Named implementations*
- *Have one or several interfaces*
- *Are the principal mechanism to create COM objects*
- *Can return interface pointers to specific COM objects*

COM Objects, Interfaces and Classes



COM: Attributes

- *COM does support attributes*
- *Attributes must be represented as set and get operations by the designer*
- *COM has a keyword to designate this*
- *Example:*

```
interface IOrganization : IUnknown {
 [propget] HRESULT Name([out] BSTR val);
};
```


COM: Operations

Parameter kind

Parameter list

```
interface IClub : IOrganization {
 [propget] HRESULT NoOfMembers([out] short *val);
 [propget] HRESULT Address([out] ADDRESS *val);
 [propget] HRESULT Teams([in] long cMax, [out] long *pcAct,
          [out,size_is(cMax),length_is(*pcAct)] ITeam *val);
 [propput] HRESULT Teams([in] long cElems,
          [in,size_is(cElems)] ITeam *val);
 [propget] HRESULT Trainers([out] ITrainer *val[3]);
 [propput] HRESULT Trainers([in] ITrainer *val[3]);
           HRESULT transfer([in] IPlayer *p);
};
```

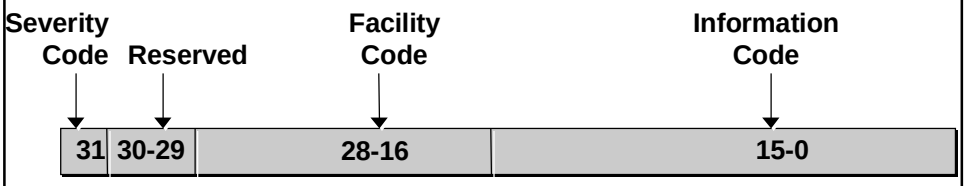
Return value indicating success/failure

Operation name

Parameter, e.g. Interface pointer

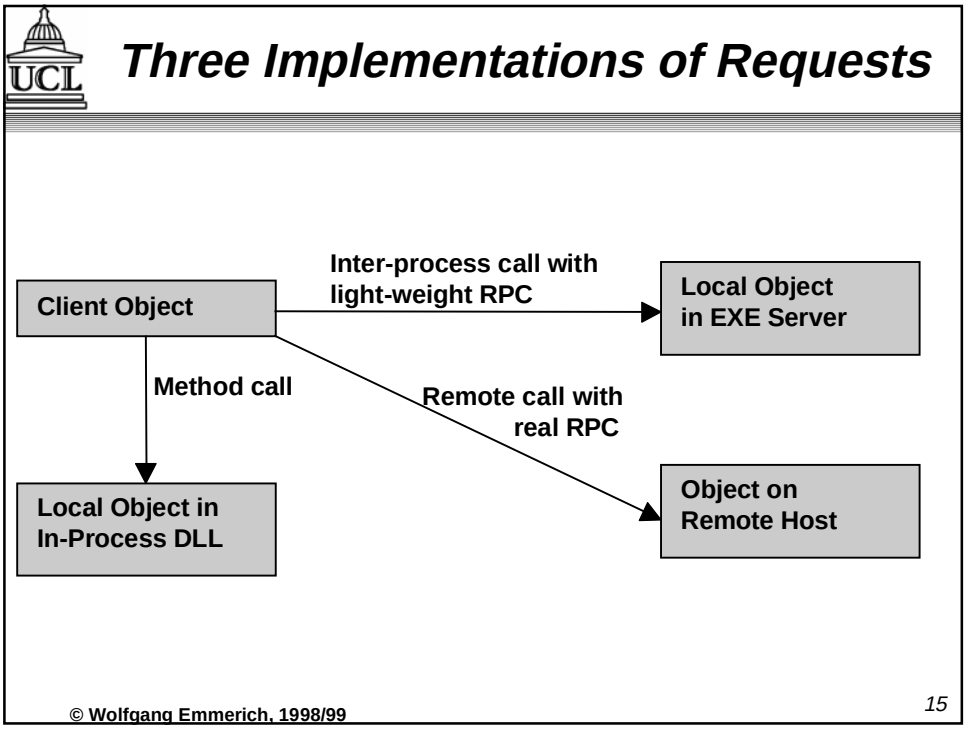
COM: HRESULTS

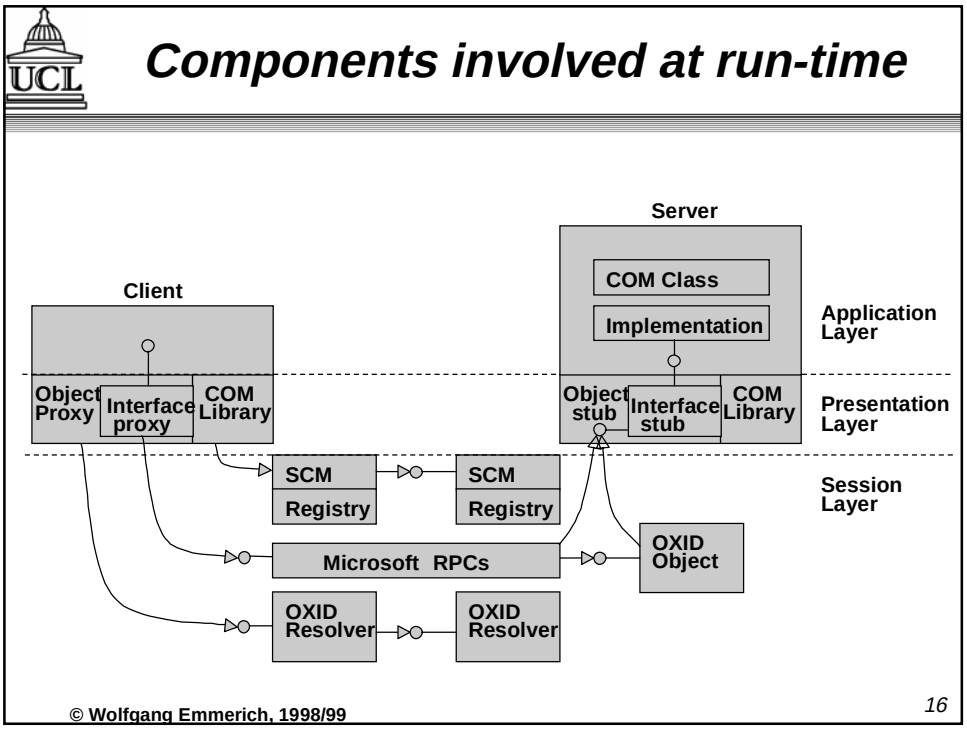
- *HRESULTS are 32-bit integers*
- *Structured into four fields*



COM Operation Invocations

- *Invocation is defined by client objects*
- *Invocation determines*
  - *Interface pointer of server object*
  - *Name of invoked operation*
  - *Actual parameters*
- *Invocation is executed synchronously*
- *Invocation can be defined*
  - *statically*
  - *dynamically*
- *Clients have to interpret HRESULTS!*


Three Implementations of Requests
Client Object
Inter-process call with
light-weight RPC
Local Object
in EXE Server
Method call
Remote call with
real RPC
Local Object in
In-Process DLL
Object on
Remote Host
© Wolfgang Emmerich, 1998/99
15


Components involved at run-time
Server
COM Class
Implementation
Client
Application
Layer
Object
Proxy
Interface
proxy
COM
Library
Object
stub
Interface
stub
COM
Library
Presentation
Layer
SCM
Registry
SCM
Registry
Session
Layer
OXID
Object
Microsoft RPCs
OXID
Resolver
OXID
Resolver
© Wolfgang Emmerich, 1998/99
16