DCOM
(Distributed Component Object Model)

Overview

• What is DCOM?
• Where does it come from?
• What makes it interesting?

Objectives

• This introduction is about:
  – DCOM
  – Changing a local scheme (COM) into a distributed one (DCOM)
  – Pitfalls in doing that
• This introduction is not about:
  – Programming using DCOM
  – COM/OLE/ActiveX/…
  – Comparison to other schemes (CORBA, Java RMI,…)

What is DCOM?

Distributed Component Object Model

“Application-level protocol for object-oriented remote procedure calls.”

IETF Internet Draft 5/96

“Object RPC”

“Microsoft RPC Version 2.0”

“COM with a longer wire”

Microsoft Press Release 9/96
The Clipboard

DDE
OLE
OLE2

DDE (Dynamic Data Interchange)

DDE
OLE
OLE2

The quick brown fox jumped over the...

The quick brown fox jumped over the...
OLE (Object Linking and Embedding)

• visible
  – inter-application drag-and-drop
  – in-place editing
  – …

• invisible
  – DDE gone
  – message-based model gone
  – COM

The Clipboard
DDE
OLE
OLE2

class information
Clipboard
Class info
The quick brown fox jumped over the ...
fox jumped over the ...

presentation data
presentation data

presentation data
presentation data

App1
The quick brown fox jumped over the ...

OLE2
COM
(Component Object Model)

- Specification describing
  - what an object is
  - interface advertising
  - object lifetime

QueryInterface()
- inquire about interfaces used by particular object

AddRef()
Release()
- Manage reference counters for object.

COM/DCOM Architecture

- COM Clients
  - Proxies map object method invocations into calls to COM/DCOM objects

- COM Servers
  - In-process
  - local
  - remote
Inproc (in-process) Servers

- Object executes in same process space as client.
- Server code implemented as DLL (dynamic linked library).
- Object has complete access to client’s memory (and vice versa)

Local Servers

- COM object is implemented in another process.
- Memories are protected.
Remote Servers

- Generally, remote servers are not different from local servers.
- DCOM as “COM with a longer wire”.

DCOM Support for Distribution

- Security
- Object References (binding)
  - representation & communication
    marshaled interface references
  - maintenance
    reference counting, pinging, delta pinging
- Optimized interfaces
  - e.g. batched QueryInterfaces() calls
- Management of causally related chains of calls
- Non-coordinated versioning of interfaces
Issues in Supporting Distribution
Example: Reference Counting

- Used for resource management:
  manage objects’ lifetimes

- Local case: reference counting
  AddRef()
  Release()

- Reference counting is fragile:
  – Abnormal termination of programs
  – Distributed case: effects of communication over network

---

Issues in Supporting Distribution
Example: Reference Counting (cont)

- Backup mechanism: Pinging
  – Periodically exchange “keepalive” messages

- Scalability?

- Large number of objects: Delta Pinging
  – Pinging of sets of objects
    ping(set_id, additions, deletions)

- Large number of clients?
Want to know more?

- Specification

- Programming

- A dissenting voice