



DCOM 기술분석 및 전망

전 제 칠¹

◆ 목 차 ◆

1 서 론	4. COM+
2 COM	5 애플리케이션의 예
3 DCOM	6 결 론

1. 서 론

1980년대 후반부터 소프트웨어에도 생산성 개념이 도입되기 시작했다. 어떤 도구와 생산 방법을 사용하면 비용을 최소화하면서 생산성, 즉 시기 적절한 소프트웨어 업그레이드를 이룰 수 있는가에 대한 연구가 많이 이루어졌었다. 수많은 연구들 중에 소프트웨어 설계 및 구현 방법론 중 가장 대두된 방법은 객체지향 프로그래밍(OOP : Object Orient Programming)이다. 하지만 소프트웨어 시장의 성숙으로 인해 객체지향 프로그래밍만으로 해결하지 못하는 문제들이 나타났고, 이런 문제들을 컴포넌트라는 개념으로 해결한 것이 바로 COM(Component Object Model)이다. 기존의 객체지향 프로그래밍 기법은 소스 코드 재사용에 의한 문제, 개발 언어에 대한 의존 문제, 버전 관리의 불편함 등 많은 문제점을 내재하고 있다.

이와 같은 문제에 직면해 객체간의 표준 통신 방법을 정하고자 하는 노력의 첫 결과로 문서 안에 문서를 넣기 위한 방법으로 OLE(Object Linking and Embedding) 버전 1이라는 기법을 도입하였고, 이 기법에서는 객체와 객체간에 DDE(Dynamic Data Exchange)라는 통신 방법을 적용했다. OLE

기술은 단순히 워드 문서 안에 엑셀 문서를 넣으려는 데서 출발해 나온 방법이었고, 그 기능도 제한적이었을 뿐만 아니라 범용성도 떨어져 다른 곳에서는 사용할 수 없다는 문제점을 노출했다.

범용성있는 객체간 통신 기법의 필요성에 의해 DDE 대신 COM 개념을 도입해 OLE 버전 2를 만들었고, OLE 2.0은 COM을 기초로 하여 그 위에 계층적으로 만든 객체 지향적 서비스들의 집합체이다. DCOM은 COM 객체가 근거리 통신망, 원거리 통신망 또는 인터넷에서 동작할 수 있도록 COM을 확장한 것이다. 네트워크 상에 있는 COM 객체를 활성화시키고 그 메소드를 호출할 수 있는 기능을 제공한다. DCOM은 COM과 함께 또는 COM 위에 만들어진 규격이고 서비스이다.

2. COM

2.1 COM의 특성

COM이란 한마디로 어떤 프로그램이나 시스템을 이루는 컴포넌트들이 상호 통신할 수 있도록 하는 메커니즘이라고 할 수 있다. 여기서 컴포넌트란 .ocx, .dll, .exe를 확장자로 갖는 실행 가능한 바이너리 파일이다. 물론 COM은 이외에도 다음과 같은 특성 및 기능을 갖는다.

† 정희원 : (주)디지털타임 대표이사

2.1.1 언어 독립성(Language Independence)

다른 개발 도구(VB, VJ++, VC++, Delphi 등)를 사용하여 작성한 컴포넌트들이 상호 통신할 수 있다는 것을 의미한다. COM을 지원하는 툴로 만든 프로그램은 컴포넌트를 마치 동일한 개발 도구로 생성한 프로그램 모듈처럼 사용할 수 있다는 것을 의미한다.

2.1.2 Binary Standard

VC++로 작성된 COM 컴포넌트를 VB 등에서 사용하기 위해 VC++의 소스가 전혀 필요없다는 것을 의미한다. VB 개발자는 VC++로 작성된 .exe, .ocx, .dll 등과 같은 binary만 있으면 이 컴포넌트를 사용할 수 있다.

2.1.3 Version Control

COM에 기반하여 작성한 컴포넌트는 하위 호환성 및 상위 호환성을 제공한다. 현재 사용되고 있는 모듈에 새로운 기능이 요구된다면 COM에서는 쉽게 새로운 기능을 기존의 모듈에 추가할 수 있다.

2.1.4 Location Transparency

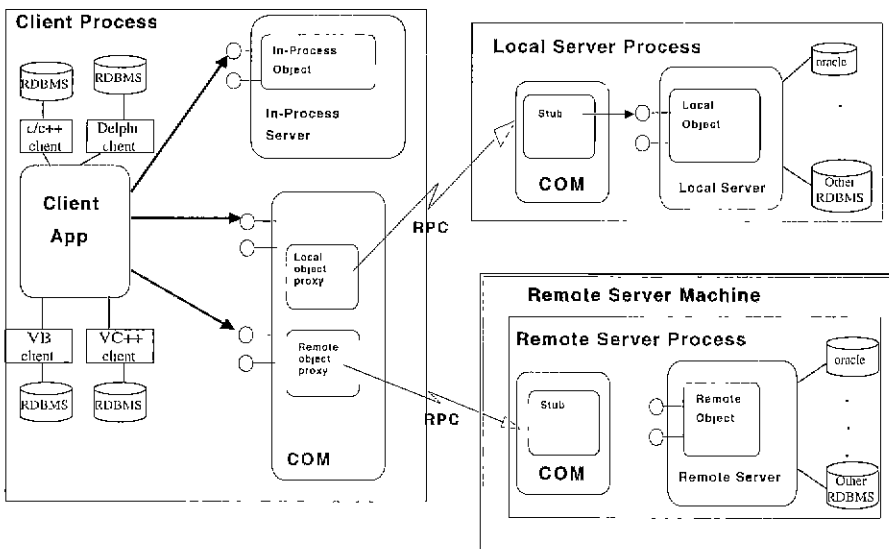
컴포넌트가 물리적 위치에 관계없이 다른 컴포넌트 혹은 컴포넌트 클라이언트에 의해 사용될 수 있다는 것을 의미한다. COM 컴포넌트는 In-process, out-of-process, out-of-machine(네트워크를 통한 리포트 머신)에서 실행될 수 있다.

In-process 형태로 수행되는 컴포넌트는 .dll, .ocx 형태로 생성되어 컴포넌트 클라이언트의 메모리 영역으로 로드되어 사용된다. Out-of-process 형태로 수행되는 컴포넌트는 .exe 형태로 생성되어 클라이언트와 동일한 머신에서 수행된다. Out-of-machine 형태로 수행되는 컴포넌트도 .exe 형태로 생성되어 클라이언트가 수행되는 머신과 네트워크를 통해 연결된 다른 머신에서 수행된다.

클라이언트가 서버에 의해 제공되는 COM objects에 접근할 수 있는 세 가지 방법이 있다.

(1) In-process server

클라이언트는 서버에 포함된 라이브러리에 직접 접근될 수 있다. 클라이언트와 서버는 같은 프로세스 안에서 실행된다.



(그림 1) COM objects에 접근하기 위한 세가지 방법

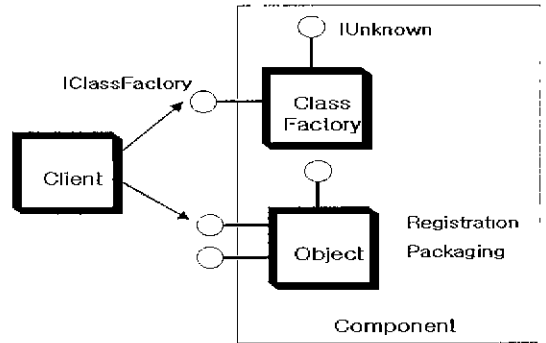
(2) Local Object Proxy

클라이언트는 다른 프로세스 안에서 움직이고 있는 서버에 연결될 수 있지만, RPC(Remote Procedure Call) 통신 메카니즘을 통해 같은 machine에 있게 된다.

(3) Remote Object Proxy

클라이언트는 다른 machine에서 작동하고 있는 원격 서버에 접근할 수 있다. 클라이언트와 서버 간의 네트워크 통신은 RPC를 통해 구성한다. 이렇게 원격 서버에의 접근을 지원하는 메카니즘을 DCOM이라 한다.

나 갯수를 바꿔서는 안된다. 경우에 따라서는 구현하고자 하는 서비스를 표준 인터페이스가 수용하지 못할 수도 있다. 이때는 새로운 인터페이스, 즉, 커스텀(Custom) 인터페이스를 디자인해야 한다.



(그림 2) COM 프로그래밍 모델

2.2 인터페이스

인터페이스는 객체가 제공하는 서비스로 메소드와 속성으로 구성된다. 각 인터페이스는 다른 인터페이스와 구별하기 위해 Interface Identifier (IID)라고 불리는 Universally Unique ID(UUID)를 할당받는다.

모든 COM 객체는 자신의 인터페이스를 구현하며 또한 IUnknown 인터페이스를 반드시 상속 받아야 한다. IUnknown 인터페이스는 COM 객체의 모든 다른 인터페이스를 접근하게 하는 Query-Interface 메소드와 AddRef와 Release 메소드를 가지고 있다.

오토메이션 객체는 IUnknown 인터페이스 외에 IDispatch 인터페이스를 포함하는 COM 객체로서 구조는 클라이언트가 인터페이스를 사용하기 위해 인터페이스의 구조를 컴파일할 때 전혀 모를도록 설계된 것이다.

클라이언트는 단지 인터페이스만을 사용한다. 따라서 인터페이스를 변경하는 것은 클라이언트의 코딩을 변경해야함을 의미한다. 그래서, 인터페이스가 어떤 식으로든 발표되면 인터페이스 함수의 순서를 바꾼다든지, 인터페이스 함수의 의미를 바꾼다든지, 인터페이스 함수의 인자의 타입이

3. DCOM

DCOM은 네트워크를 통해 안정적이고 효과적인 방법으로 소프트웨어 컴포넌트들이 통신할 수 있는, 이전에는 네트워크 OLE라 불리기도 했던 프로토콜이다. 즉, DCOM은 COM 객체가 근거리 통신망, 원거리 통신망 또는 인터넷에서 동작할 수 있도록 COM을 확장한 것으로, 네트워크 상에 있는 COM 객체를 활성화시키고 그 메소드를 호출할 수 있는 기능을 제공한다.

3.1 DCOM의 구조

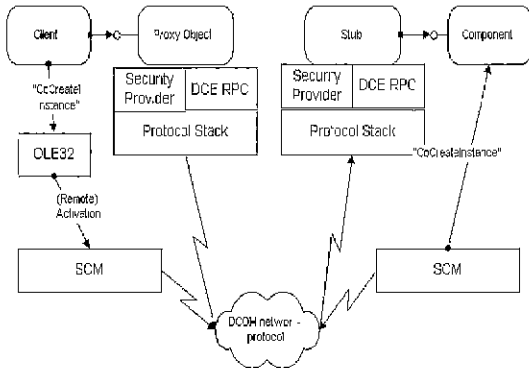
(그림 3)에 있는 것처럼 DCOM은 COM 애플리케이션이 원격 컴포넌트를 프록시와 스텝을 이용하여 마치 같은 기계 내에 있는 것처럼 사용하

1) 컴포넌트가 프로세스밖 서버나 프로세스밖 원격 서버안에서 제공하는 함수 호출시 함수 호출을 가로채 LPC나 RPC로 전달하는 기능을 하는 DCOM 실행부 객체를 말한다. 이 프록시 객체는 디플트로 DCOM에서 제공하는 것을 사용할 수도 있으나 프로그래머가 자신의 컴포넌트에 맞도록 만들 수도 있다.

게 한다.

3.2 DCOM 특성

DCOM을 사용하면 COM의 도움을 얻어 원격 컴퓨터 안에 있는 COM 컴포넌트를 사용하는 것처럼 마음대로 사용할 수 있다. 이런 특징 때문에 DCOM을 이용해 여러 개의 원격 컴포넌트로 구성된 분산 응용 프로그램을 만들 때 많은 장점이 생긴다.



(그림 3) DCOM의 구조

3.2.1 컴포넌트 재사용

DCOM은 복잡한 네트워크 프로그래밍에 대한 부담을 줄여 개발 시간을 단축시켜줄 뿐만 아니라 기본적으로 COM의 특성을 지니므로 COM의 뛰어난 컴포넌트 재사용 특성을 이용해 필요한 컴포넌트를 개발하지 않고 구매해 사용함으로써 개발 시간 및 비용을 획기적으로 줄여줄 수 있다.

3.2.2 위치 투명성²⁾ 제공

DCOM의 도움없이 분산 응용 프로그램을 개발한다고 하면, 응용 프로그램 디자인사 여러 가지 지역적인 특성을 고려해야만 적절한 성능을 얻을

수 있다. DCOM은 지역적으로 떨어져 있는 컴포넌트를 이용할 때 프로그램 안에서는 이 컴포넌트의 위치를 구별하지 않아도 되게 위치 투명성을 제공하고 있다. 따라서, 프로그래머들은 분산 응용 프로그램의 목적에 부합하도록 프로그래밍하면 되고, 소스 코드의 수정없이 컴포넌트를 적절하게 배치해 컴포넌트간 통신 부담을 최적화할 수 있다.

3.2.3 복수 스레드의 서버

윈도우 NT 4.0에 자유 스레딩(free threading)이라고 하는 병행 모델이 도입되어 한 객체에 대한 복수개의 호출이 동시에 별개의 스레드에 할당되어 처리될 수 있다.

3.2.4 보안

DCOM은 관리자나 개발자가 선택할 수 있는 다단계의 보안을 제공한다.

4. COM+

4.1 COM+의 등장 배경

4.1.1 COM에서 MTS로

MTS는 Windows NT Server를 위해 만들어진 소프트웨어의 하나이다. 이름이 의미하듯이 MTS는 Windows NT Server에서 실행되는 미들 티어 개체가 미들 티어에서 분산 트랜잭션을 실행하고 제어할 수 있게 한다. 그러나 MTS는 트랜잭션 모니터 이상의 것으로, 미들 티어에서 실행되는 COM 개체에 새로운 런타임 환경을 제공한다. MTS에는 COM에 포함되어 있지 않은 많은 중요한 인프라 지원이 추가되었다.

4.1.1.1 보안기능의 확장

MTS는 분산 트랜잭션 지원 기능과 함께 COM의 보안 모델을 확장했다. MTS 보안은 롤(role)이

2) Location Independence. 프로그래머에게 컴포넌트가 어디에 배치됐는지 모르도록 하는 DCOM의 핵심 기능. 이 기법으로 다양한 성능 최적화가 가능하다.

라는 개념에 기반을 둔다. 롤은 MTS 응용 프로그램에서 하나 이상의 사용자에 대한 보안 프로파일을 나타내는 추상적인 개념이다. 개발자는 설계 과정에서 롤을 선언하거나 프로그래밍하여 보안 검사를 설정할 수 있다. 배포 과정에서 관리자는 일련의 롤을 Windows NT 도메인에 있는 사용자 계정과 그룹 계정에 매핑한다.

4.1.1.2 스레드 풀링 스키마

MTS가 플랫폼에 추가한 또 다른 중요한 기능은 백그라운드에서 스레드 풀링을 수행하여 병행성을 유지하는 스키마이다. MTS 프로그래머는 하나의 클라이언트 응용 프로그램마다 스레드(작업)가 하나씩은 있도록 응용 프로그램을 작성해야 한다. MTS 런타임은 논리적 작업을 물리적 작업에 자동으로 바인드한다. 클라이언트의 갯수가 사전에 지정한 임계값에 도달하면 MTS 런타임은 여러 클라이언트에서 스레드를 공유하기 시작한다.

4.1.1.3 구성 및 관리기능의 개선

MTS의 또 다른 중요한 특징은 컴퓨터 구성과 네트워크 관리 기능이 개선되었다는 것이다. COM에서는 COM 기반 응용 프로그램을 네트워크에서 배포, 관리하는 비용이 많이 들고 복잡했다. MTS 관리 도구를 사용하면 미들웨어 개체를 실행하는 서버 컴퓨터를 훨씬 쉽게 구성하고 관리할 수 있다.

4.1.1.4 속성 기반 프로그래밍

MTS는 속성 기반 프로그래밍이라는 중요한 개념을 플랫폼의 프로그래밍 모델에 도입했다. 플랫폼이 제공하는 서비스는 속성의 선언을 통해 드러난다.

속성 기반 프로그래밍 모델의 장점은 속성 설정으로 설계 과정에서 기본 설정을 지정할 수 있

으며 이 기능은 런타임 환경에서 기본적으로 지원되고 응용 프로그램을 제품에 적용한 후 이 응용 프로그램이 시스템 서비스를 사용하는 방법을 관리자가 쉽게 다시 구성할 수 있다는 것이다. 따라서, 코드를 수정하거나 응용 프로그램을 다시 컴파일할 필요가 전혀 없게 된다.

4.1.2 COM과 MTS 문제의 비교

Windows NT Server를 대상으로 하여 멀티 티어 응용 프로그램을 구축하는 프로그래머의 가장 큰 고민은 언제 COM을 사용하고 언제 MTS를 사용할 것인가를 결정하는 것이다.

COM은 Windows NT Server와 함께 출시되지만 MTS는 그렇지 않다. MTS를 사용하려면 Windows NT Option Pack이라는 별도의 소프트웨어를 설치해야 한다. 이것은 MTS가 COM의 일부가 아님을 의미한다. MTS는 단순히 COM의 맨 위에 제공되는 레이어임을 파악해야 한다. COM은 MTS를 적용하기 위해 변경되지 않았다. 이것은 두 개의 런타임 레이어가 완전히 통합되지 않았음을 의미한다. 이런 까닭에 플랫폼은 비효율적이고 모호해지며 혼동되게 된다. MTS 프로그래머는 두 프로그래밍 모델의 작동 방법을 모두 알아야 하는 어려움이 있다.

4.2 COM+의 등장

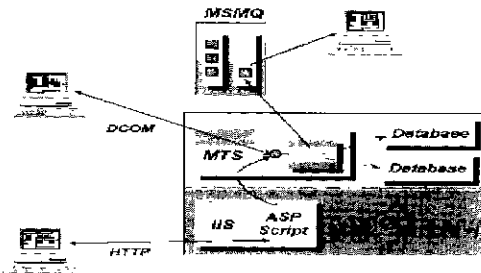
Windows 2000의 출시로 COM과 MTS의 장점이 COM+라는 하나의 새로운 런타임으로 통합되었다. COM+는 이진 구성 요소 및 인터페이스 기반 프로그래밍에 기반을 둔다. 메서드 호출은 알기 쉬운 RPC 레이어를 사용하여 프로세스 및 컴퓨터 경계간에 원격으로 처리된다. 또한, COM+ 구성 요소가 클라이언트 제품에 적용되었을 때 업그레이드 및 확장시 응용 프로그램에 전혀 영향을 주지 않는다.

MTS와 마찬가지로, COM+는 분산 트랜잭션과

를 기반 보안을 지원한다. COM+는 기본 스프레드 폴링 스키마를 제공하며 이 스키마는 MTS에 사용된 스키마만큼 알기 쉽게 되어 있다. 또한 COM+ 프로그래밍 모델은 인터셉션을 사용한 속성 선언을 통해 개발자에게 플랫폼 서비스를 드러낸다. 그러나 COM+는 MTS보다 속성 기반 프로그래밍을 더 많이 사용한다. 트랜잭션 서비스와 통합 보안 이외에도 COM+는 사용자 정의 개체 구성, 동기화, 개체 폴링과 같은 서비스를 지원한다. 대기 중인 구성 요소 및 COM+ 이벤트도 COM+의 새로운 기능이다.

5. 애플리케이션의 예

Microsoft 컴포넌트 서비스가 어떻게 동작하는지를 다음의 예를 통해 보자. 카탈로그 마케팅 회사가 Microsoft 컴포넌트 서비스 COM/DCOM, MTS, ASP, IIS, MSMQ를 사용하여 NT에 기초한 주문 처리 시스템을 선택했다고 가정하자. 간단한 다이어그램은 아래 (그림 4)와 같다.



(그림 4) 주문 처리 시스템 다이어그램

시스템의 핵심부는 주문을 처리하는 데 필수적인 비즈니스 로직을 포함하는 하나 이상의 COM

컴포넌트이다.

주문이 어떻게 작성되는지에 대해 보자. 이런 시스템에서, 주문처리 컴포넌트에 대한 두 가지 다른 사용자 인터페이스가 있다. 대부분의 카탈로그 회사와 같이, 주문을 하는 고객으로부터 전화를 받는 오퍼레이터가 있다. 이 오퍼레이터들은 Win98, Windows NT가 가동 중인 워크스테이션에서 고객 주문을 생성하기 위해 커스텀 클라이언트를 사용한다. 이 클라이언트는 DCOM을 사용해서 서버에 있는 주문처리 컴포넌트와 통신하게 된다.

또 다른 하나는, 고객이 인터넷을 통해 직접 주문을 하도록 하는 것이다. 이 인터페이스를 가능하게 하기 위해 IIS를 통한 ASP가 작성된다. 웹 브라우저를 사용하는 고객은 회사의 웹사이트를 통해 이 ASP 스크립트를 액세스하고, 주문을 보낸다. ASP 스크립트는 간단하게 COM 객체를 사용할 수 있기 때문에 웹을 통한 고객 주문은 DCOM 클라이언트와 마찬가지로 주문 처리를 위해 같은 MTS 컴포넌트에 의존하게 된다.

6. 결 론

모든 소프트웨어 개발자는 적은 시간에 가장 효율적인 애플리케이션을 만드는 공통적인 목표를 가지고 있다. 그리고, 그 모든 개발자들은 그 환경에서 가능한 서비스를 개발하는 것이 그 목표를 달성하는 주요한 부분인 것을 알고 있다. 목표 달성을 위해 컴포넌트 개념을 도입한 COM은 네트워크를 통해 안정적이고 효과적인 방법으로 소프트웨어 컴포넌트들이 통신할 수 있는 DCOM으로 확장됐다. 또한, 미들 티어에서 실행되는 COM 개체에 새로운 런타임 환경을 제공하고 분산 트랜잭션, 통합 보안, 스프레드 폴링에 대한 지원과 구성 및 관리 기능이 개선된 MTS가 등장했다. 최근 Windows 2000의 출시로 COM과 MTS의 장점들이 COM+라는 하나의 새로운 런타임으로 통합

3) Business Logic Component. 실제 회사 업무 흐름을 구현하는 컴포넌트를 말한다. DCOM은 프로그래머에게 네트워크 프로그래밍 및 기타 관리에 필요한 프로그래밍에 대한 부담을 덜어주도록 디자인된 업무 로직 컴포넌트 개발에 집중하도록 한다.

되었다.

이런 일련의 개발은 컴포넌트 재사용으로 인한 비용/개발 시간의 절감 효과, 위치 투명성 때문에 얻을 수 있는 장점들, 개발 언어를 마음대로 선택할 수 있는 이점, 클라이언트와 서버의 효율적인 연결관리, 적절한 컴포넌트 배치를 통한 성능 향상, 편리한 컴포넌트 버전 관리, 기본적으로 구비된 보안 기능, 다양한 전송 프로토콜 지원, 플랫폼 독립성, 인터넷 프로토콜과의 완벽한 결합 등 많은 기능들을 구현하여 인터넷 환경에서 분산 애플리케이션 개발에 강력함을 더하기 위해 수행되었다.

앞으로 컴포넌트 서비스들은 차세대 엔터프라이즈 애플리케이션의 기초가 될 것이다.

참고문헌

- [1] Box, Don. Essential COM. Menlo Park, CA: Addison-Wesley Publishing Company, 1988.
- [2] Grimes, Richard H. Professional DCOM Programming. Chicago, IL: WROX Press, Inc., 1997.
- [3] Redmond III, Frank E. DCOM: Microsoft Distributed Component Object Model. Foster City, CA: IDG Books Worldwide, 1997.
- [4] MSJ(Microsoft System Journal) - ActiveX/COM Q&A
- [5] 마이크로소프트 홈페이지, <http://www.microsoft.com/com/wpaper/>
- [6] MSDN Library
- [7] 정보과학회지, 1999년 7월, 정보과학회

전 제 철



1991년 서강대 전산과 졸업
 1991년-1996년 현대전자 S/W연구소
 1996년-1997년 현대정보기술 S/W
 연구소
 1997년 디지털드림 설립 및 대표이사
 취임

1999년 Carnegie Melon Univ. GSIA 최고 경영자 과정
 현재 (주)디지털드림 대표이사