

COM을 이용한 데이터베이스 인터페이스 설계

박상훈 강성기 박진섭
대전대학교 컴퓨터공학과
{shpark,napole2}@zeus.taejon.ac.kr
Jspark@dragon.taejon.ac.kr

Development of Database Interface using COM

Sang-Hoon Park Sung-Gi Kang Jin-Sub Park
Dept. of Computer Engineering, Taejon University

요 약

본 논문에서는 데이터베이스 인터페이스 설계가 COM(Component Object Model)으로 구현하는 것이 가능한지를 실험적으로 설계하고 구현하였다. 본 객체지향 코드 설계는 캡슐화, 상속성, 다형성 측면에서 설계자 및 사용자에게 많은 장점을 제공한다는 측면에서 널리 적용되고 있다.

본 논문에서는 COM을 이용하여 기존의 데이터베이스를 접근, 제어 할 수 있음을 보인다. 본 논문은 기존 데이터베이스로서 MS Access(DB)를 대상으로 하되 MS Access가 제공하는 최종 사용자 기능을 무시하고 새로운 프로그램(COM)을 설계하여 기존사용방법과 같은 효과와 기능을 발휘할 수 있도록 한다. 프로그램의 설계는 크게 서버와 클라이언트로 나눈다.[10] 실험적 개발의 결과를 볼 때 DB엔진 설계 시 COM 으로서의 설계에 활용 할 수 있을 것으로 기대된다. 한편 클라이언트는 다양한 언어로 설계가 가능하기 때문에 다양한 사용자, 다양한 접속 요구에 쉽게 적용할 수 있다.

1. 서론

최근 객체지향 기술에서 보다 더 나아가 컴포넌트 기반[1,2,3]의 소프트웨어 개발이 부각되고 있으며, 컴포넌트를 이용한 새로운 정보기술 패러다임이 떠오르고 있다. 기존의 소프트웨어 개발 방법에서 조립 기반의 부품화를 통하여 생산성 및 품질의 획기적인 향상을 기대하고 있다. 특히 웹 기반의 기업 레벨 정보시스템 구축 및 DB의 인터페이스 설계에 있어서 그 중요성이 더욱 부각되어지고 있다. 일반적으로 데이터베이스 인터페이스 설계에 있어서 ODBC[4,5]를 이용하여 데이터베이스를 관리한다. ODBC는 기본적으로 덩치가 큰 데이터베이스 엔진이므로 DB를 쉽고 빠르게 접근하여 데이터를 처리할 수 없는 단점을 가지고 있다.

이러한 해결책으로 본 논문에서는 COM(Component Object Model)을 이용한 DB 인터페이스 설계에 초점을 두었다. COM[1,2,3,10]은 이진 소프트웨어 컴포넌트를 기반으로 애플리케이션을 구현할 수 있도록 하는 소프트웨어 구조이다. 본 논문의 인터페이스 설계[10]는 서버, 클라이언트로 나누어져 있고 클라이언트 입력은 일반 SQL 문을 받아들여 사용자가 원하는 결과를 보여 줄 수 있도록 한다. 서버는 클라이언트에서 해독된 SQL문을 받아들여 DB의 데이터를 직접 핸드링 할 수 있도록 한다

본 논문의 구성은 다음과 같다. 2장에서는 COM과 ODI를 비교 설명하고 COM 인터페이스의 장점에 대해 기술다. 3장에서는 JetEngine에 대해 기술한다. 4장에서는 실험 결과를 토대로 분석한다. 마지막으로 5장에서는 결론 및 후 연구방향을 제시한다.

2.COM 과 ODBC

2.1 COM과 ODBC의 개념

COM은 Microsoft의 기본 객체 기술에 대한 규정의 이름이며 COM 객체의 의미와 호출되는 방법을 정의한다.COM 구성요소[1,2,3,10]를 잇는 접착제와 같은 역할을 하는데로 관련이 없는 소프트웨어 객체를 의미 있는 방식으로 결합하고 그것들이 상호 작용하게 만든다.

ODBC(Open DataBase Connectivity)[4,5]는 데이터베이스 엔진을 이용하여 유저 인터페이스 프로그램을 작성할 때 사용하는 애플리케이션 인터페이스이다. 결국 ODBC는 드라이버와 연결되는 명령 통로로 생각하면 된다. 본 논문의 실에서는 ODBC를 사용하지 않고 COM을 사용하여 데이터베이스에 접근하였다 <표 1>은 ODBC와 COM을 비교 분석 것이다.

<표 1>에서 보여 주듯이 ODBC를 이용하여 데이터베이스 접근하는 것보다 COM을 이용하여 접근하는 것이 더 효율적이라는 것을 알 수가 있다.

표 1 ODBC와 COM의 비교 분석

	ODBC	COM
1	ODBC는 데이터베이스 안의 기본 구조를 바꾸는 것이 용이하지 못하다.	ODBC 보다는 데이터베이스 파일을 새로 만들고, 사용 도중에 새로운 필드가 필요하면 생성하고 기존의 필드는 삭제하며, 테이블 또한 새로 만들고 삭제하는 것이 쉽다.
2	ODBC 드라이버를 이용	DLL과 EXE 형태를 통해 제공되기 때문에 속도가 ODBC보다 빠르다.
3	드라이버가 갱신되면 응용프로그램에 영향을 미친다.	COM은 다중 인터페이스를 지원하는 객체와 결합된 불변의 인터페이스를 통해 버전 부여 문제 해결.

2.2 COM 인터페이스

COM에 있어서 인터페이스[1,2,10]는 소프트웨어 구성 요소와 클라이언트 사이에서 강력하게 유형화된 계약이며 의미적으로 연관된 비교적 작지만 유용한 일련의 작업을 제공한다. <표 2>에서는 COM 인터페이스의 장점을 나타내고 있다.

표 2 COM 인터페이스의 장점

1	COM 컴포넌트에 새로운 기능을 부여하여 기존 클라이언트들과는 독립적으로 수행
2	빠르고 간단한 객체 상호작용
3	인터페이스 재사용
4	지역/원격지 투명성
5	프로그래밍 언어 독립성

3. Microsoft Jet Engine

Jet 엔진[6,7]은 MS 제품 군에서 실행되는 MS사의 관계형 데이터베이스 엔진의 이름이다. Jet Engine은 다양한 외부 데이터를 불러올 수 있으며 외부 형식으로 된 파일을 처리할 수 있다. Jet 엔진을 사용하면 개발자들이 매우 공통적으로 프로그램을 작성하여 데이터베이스 파일로부터 다양한 형식의 데이터를 가져와 결합할 수 있다.

본 논문의 실험에서 COM 프로그램으로 Microsoft Access의 JetEngine[6,7]을 접속하고 제어하는 것이 가능함을 보였다.

4. 구현 및 실험결과

프로그램 구성은 그림 1과 같다.[10]

- ① Component 접속[InitEngine()]: 컴포넌트와 접속한다. *DAO DBEngine()의 프로그램 이름을 통해서 ID를 얻어온다.
→hr=CLSIDFromProgID(OLESTR("DAO.DBEngine.35"),&clsidEngine);

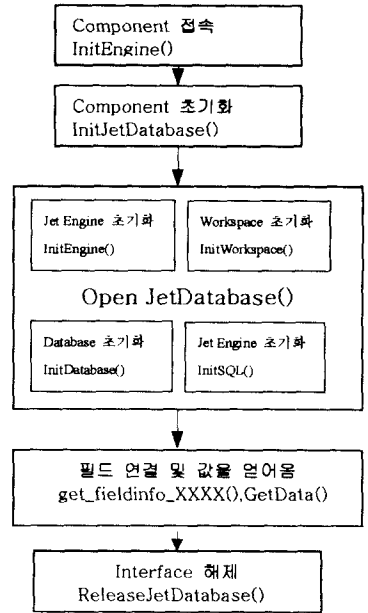


그림 1 프로그램 구성도

- * DAODBEngine의 객체를 생성한다.
→hr=CoGetClassObject(clsidEngine, CLSCTX_INPROC_SERVER, NULL, IID_IclassFactory2, (LPVOID*)&pCF2);
- * 생성하고자 하는 컴포넌트의 인스턴스를 얻어온다.
→hr = pCF2->CreateInstanceLic(NULL, NULL, guidEngine, V_BSTR(&varKey), (LPVOID*)&m_pDAODBEngine);
- ② Component 초기화[InitJetDatabase()]: COM 객체를 초기화 한다. hr = ::CoInitialize(NULL);
- ③ OpenJetDatabase(): password, username과 SQL문을 설정했는지 검사하며, 초기화 여부에 따라서 초기화 할 것인지를 결정한다.
- * Jet Engine 초기화[InitEngine()]
- * Workspace 초기화[InitWorkspace()]: DAODBEngine 으로부터 DAOWorkspace 객체를 얻어온다.
- * Database 초기화[InitDatabase()]: DAOWorkspace 객체에서 DAODatabase객체를 얻어온다.
- * SQL 초기화[InitSQL()]: 이 함수는 QueryDef 객체를 open하고 미리 지정된 SQL 문을 JetEngine에 할당한다. 그리고 Jet Engine에서 DAORecordset 객체를 얻어온다.
PDAOQueryDef->_30_OpenRecordset(ColeVariant((long)dbOpenDynaset), ColeVariant((long)0), &m_pDAORecordset); 여기서 &m_pDAORecordset로 DAORecordset객체를 얻어 올 수 있다.

또한, GetFieldCount(pDAOQueryDef, m_nFields)로 필드의 개수를 얻어온다. BindingField(pDAOQueryDef, m_nFields) 함수에서 필드의 속성들을 얻어온다. Access 에서 어떤 형으로 사용되었는지 알아야만 필드의 값에 맞는 형으로 얻어 올 수 있다.

- ④ 필드연결 [get_fieldinfo_XXX()]. 예를 들어, *get_fieldinfo_Bool→FieldExchange_Bool을 호출하여 Field와 변수를 연결한다. output : bool *pVal => 외부 변수와 FieldExchange_Bool의 반환값을 연결
- ⑤ 필드 값을 얻어옴[GetData()]: FieldExchange_XX 함수들에서 설정된 각각의 값들을 실제로 한 레코드 당 하나씩 데이터를 설정하는 함수이며, pCDAORcordsetGetRows->GetRows()에서 각각 필드의 값을 얻어온다. 역시 이 함수에서 필요한 경우 CallBack 함수를 호출한다.
- ⑥ Interface 해제[ReleaseJetDatabase()] => ::CoUninitialize(); COM 객체 해제

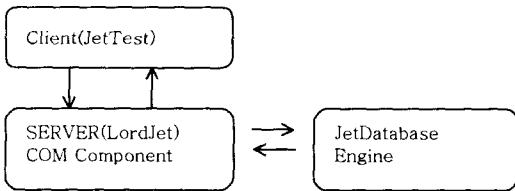


그림 2 프로그램 구조

(그림 3)은 Com을 이용하여 Microsoft Access의 JetEngine에 접근하여 manager.mdb 파일[10]을 불러 온 것이고 (그림 4)는 Access에서 확인 한 것이다.

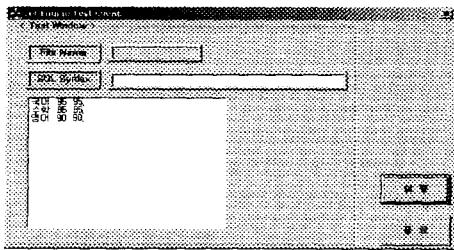


그림 3 실험결과

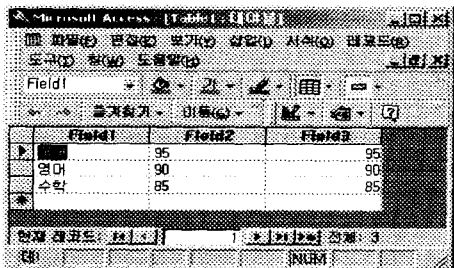


그림 4 Access 에서 확인

5. 결론 및 향후 연구

본 논문의 실험을 통하여 COM으로의 인터페이스가 가능함을 확인하였기 때문에 서버 부분은 DB엔진 내부에 통합화 시킬 필요가 있다. 이것은 다양한 클라이언트로의 DB 접속을 가능하게 할 것이다.

서버와 클라이언트를 통합화 시킬 경우에는 다양한 DB 접속 인터페이스를 추가 시킬 필요가 있다.

이는 하나의 TOOL로서 다양한 DB 접근 및 제어 가능하게 할 것이다. COM의 장점을 이용하여 DB 접근 및 데이터 처리를 효율적으로 설계하는 것이 가능하다.

또한 MS Access외에 Oracle, Informix등 다른 DB와의 인터페이스 모듈을 추가하여 패키지로 활용하는 것도 향후 의미 있는 활용 분야이다.

참고문헌

- [1] Guy Eddon and Henry Eddon., "Inside Distributed COM": Microsoft Corporation. 1997
- [2] Eddon G., "COM+: The evolution of component services": Computer, V.32 N.7, 104-106, 1999
- [3] John Wiley and Sons., "Client/Server Programming with Java and CORBA": simultaneously in Canada 1998
- [4] Robert Signore, John Creamer, Michael O. Stegman "The Odbc Solution" McGraw Hill 1995
- [5] Chuck Wood "Ole Db Odbc Developer's Guide" IDG Books Worldwide 1999
- [6] Dan Haight, Jim Ferguson., "Microsoft Jet Database Engine Programmer's Guide (Microsoft Professional Editions)" Microsoft Press 1997
- [7] Joe Iauca "Database Developer's Guide Using the Jet Engine"
- [8] Bard King, George Shepherd "Inside ATL (Programming Languages/C)" Microsoft Press 1999
- [9] Richard, Dr. Grimes, Alex Stockton, Julian Templeman, George V. Reilly "Beginning ATL COM Programming" Wrox Press Inc 1998
- [10] 박진섭, 강성기, 박상훈 "데이터베이스 COM 인터페이스 개발" 보고서 NetworkLab, Taegon University 2000