데스크탑 기반의 클라이언트 프로그램을 PDA 용 프로그램으로 마이그래이션하기 위한 연구

김태환° , 김종완, 류성열
숭실대학교 컴퓨터학과
thkim126@selab.ssu.ac.kr, wany69@hitel.net
syrhew@computing.soongsil.ac.kr

A Study on DeskTop-based Program to PDA-based Program Migration

Tae-Whan Kim°, Jong-Wan Kim, Sung-Yul Rhew
School of Computing, Soongsil University

요약

기존의 클라이언트/서버 환경에서 클라이언트가 유선으로 이루어져 이동성과 접근성에 제약을 가지고 있는데 이를 해결하기 위해서는 PDA와 같은 무선기기를 이용하여 정보를 조회하고 입력하는 시스템이 필요하다. 그러나 기존의 클라이언트/서버 환경의 클라이언트를 PDA로 마이그래이션하여 통합하기 위해서는 하드웨어, 운영체제, 개발도구 등의 차이로 인해 많은 어려움이 있다.

본 논문에서는 이를 해결하기 위하여 PDA가 지원하지 않는 ODBC, MDI, DAO, OLE와 같은 문제를 해결 할 수 있는 제어컴포넌트를 만들어 여러 서버에 추가하고 재사용 함으로써 클라이언트/서버에서 PDA로 마이그래이션하는 개발과정의 시간과 노력을 절감하게 한다.

1. 서론

과거의 대부분의 컴퓨터들은 유선을 이용하여 일정한 장소에서만 사용 가능한 시스템으로 구성되어 있어 공간에 상당히 제한적인 면을 가지고 있다.

즉 이러한 공간 제한적인 문제를 해결하기 위하여 현재 모바일 컴퓨팅(Mobile Computing)을 이용하여 공간과 위치의 제약 없이 원하는 컴퓨팅 작업을 할 수 있는 환경으로 바꾸어 나가려는 노력이 증가되고 있으나, 대부분 미국을 중심으로한 환경이고, 국내 환경은 미비한 인프라와 PDA(Personal digital assistant)에 대한 낮은 인식률 등의 원인으로 활용도가 극히 저하된 상태로 머물고 있는 실정이다. 그러나 최근 핸드폰을 이용한 무선 인터넷이 이슈가 되고 IMT-2000(International Mobile Telecommunications-2000)이 대두되면서 국내에서도 모바일 컴퓨팅에 많은 관심을 보이게 되었다. 이러한 흐름에 따라 기업에서는 기존의 데스크탑 소프트웨어를 모바일용 소프트웨어로 변환하여 기존의 시스템에 추가하려는 노력을 하고 있다[1].

모바일용 소프트웨어는 데스크탑 소프트웨어와 구성면에서 유사하지만 램IC에 직접 저장 실행되며, 적은 메모리를 사용하고 , 입력장치의 한계 등 하드웨어적 제약 사항을 가지고 있어 기존의 소프트웨어를 모바일용 소프트웨어로 변경하는데는 많은 어려움이 있다[2].

본 논문에서는 기존에 데스크탑용 윈도우즈 플랫폼에서 개발된 어플리케이션을 윈도우CE 플랫폼의 모바일 컴퓨팅 어플리케이션으로 마이그래이션(Migration) 할 때의 문제점을 비교 검토하고, 그 해결 방안을 제시한다.

2. O/S(operation System)의 차이

윈도우CE는 데스크탑 O/S인 윈도우즈에서 가장 핵심적인 기능만을 선택하여 만든 임베디드(embedded)시스템용 O/S로 작고 가벼우며 멀티태스킹, 멀티쓰레드를 실현한 32비트 운영체제로서 여러 가지 디바이스를 동작시킬 수 있도록 오픈 아키텍처를 기초로 하여 설계되었으며, 작은 메모리 환경에서 높은 성능을 실현하여 종래의 모빌, 멀티미디어 제품에도 대응될 수 있도록 설계되었다[4].

[표1]은 데스크탑용 윈도우98과 PDA용 윈도우CE의 기본적인 특징을 비교 분석한 표로 윈도우CE가 작은 운영체제로 구성되어 있음을 나타내고 있다.

[표 1] 윈도우98과 윈도우CE에 관한 비교

종류 \ 항목	윈도우98	윈도우CE	비고
플랫폼 호환성	16/32bit	32bit	완 전 한 32bit
지원 Win32API	약 500개	약 150개	
O/S 크기	약 200M	약 1.5M	
가상주소공간	2GB	32MB	
물리적 공간	하드디스크를 사용한 대용량	RAM	
입력방식	마우스, 키보드	터치스크린	

3. 개발 도구의 차이

3.1 윈도우CE 개발 도구 선택

현재 윈도우CE용으로 개발하기 위한 도구는 Platform Builder3.0 및 eMbedded Visual Tools3.0과 같은 두 가지 기본 개발 도구들이 있다.

Platform Builder는 윈도우CE 자체의 사용자 정의 설정을 만들기 위해 사용할 수 있는 도구로 사용자 정의 SDK를 만들어서 현재 버전의 플랫폼을 사용하는 응용 프로그램 개발자의 노력을 지원하는 기능을 가지고 있다. eMbedded Visual Tool3.0 패키지는 Visual StudioR와 유사한 윈도우CE 기반의 개발을 위한 통합 환경으로 Visual Basic, Win32, MFC(Microsoft Foundation Class)라이브러리 및 ATL(*ActiveX Template Libraries*) 같은 4개의 인터페이스를 지원한다.

3.2 표준MFC와 MFC for windowCE의 차이

MFC는 잘 구조화된 윈도우즈 어플리케이션을 개발하기 위한 훌륭한 도구로 GUI(Graphic User Interface)application을 개발하기 위한 견고하고 포괄적인 클래스 집합이다.

윈도우CE를 위한 MFC는 표준 MFC와 기능 및 특성이 가깝도록 디자인되었지만 데스크탑에 비해 메모리가 적고, CPU속도도 느리며, 하드디스크와 같은 저장매체가 없는 하드웨어적 제약사항 때문에 특정 기능을 위해 추가 및 수정된 클래스는 다음[표2]와 같고, 지원되지 않는 클래스는 [표3]과 같다[3].

[표 2] 윈도우CE에 추가 및 수정된 MFC클래스

항목	클래스
새로 추가된 MFC 클래스	CCeDBDatabase, CCeDBEnum, CCeDBProp, CCeDBRecord, CCeSocket
수정된 MFC 클래스	*CStatic, CString, CSyncObject,* CTabCTrl, CTime, CView, CWinApp CWinTread, CWnd.....

[표 3] 윈도우 CE에서 지원되지 않는 MFC 클래스

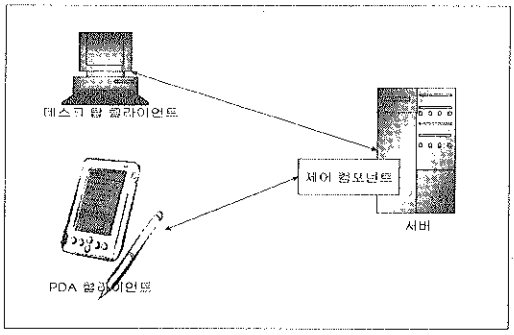
항목	클래스
공통 다이알로그	*CColorDialog, CFontDialog.....*
공통 컨트롤	CAnimateCtrl, CCheckListBox .....
컨트롤바	CControlBar, CDialgBar, CStatusBar.....
DAO	CDaoDatabase, CDaoException .....
인터넷	*CFtpConnection, CFtpfileFind.....*
MDI	CMDIChildWnd, CMDIFrameWnd.....
ODBC	CDatabase, CDBException CDVariant.....
OLE	CAsyncMonikerFile,.....
프린팅	CPrintingDialog, CPrintInfo
리치 에디트 컨트롤	CRichEditCntrl, CRichEditCntrlItem,....
동기화	CMutex, CMultiLock, CSemaphore
그 외에 지원되지 않는 클래스	CDomkState, CFileFind, CMetafileDC, CPalette, CPictureHolder, CSharefile, CStudioFile

데스크탑을 윈도우CE로 변경할 때의 문제는 보통 이러한 서브셋API에서 발생할 것이며, 특히 어플리케이션이 표준MFC로 쓰여졌다면 윈도우CE를 위한 MFC와 호환이 되는지를 파악하기 위해 쓰여진 클래스, 메소드, 프로퍼티등을 세심하게 확인하여 사용해야 한다[1].

4. 제어 컴포넌트를 통한 해결방안

기존의 데스크탑 윈도우의 클라이언트 프로그램을 PDA용 윈도우CE 프로그램으로 마이그래이션 하기 위해서는 O/S의 차이, 개발언어의 차이, 제한된 메모리 및 out-of-memory시의 복구, 제한된 전력, 변화가 큰 하드웨어 특성 등과 같은 문제를 만나게 되므로 어떤 기능은 윈도우CE에서 지원하지 않으므로 삭제하여 사용해야 하거나 다른 API를 사용하여 문제를 해결해야 한다.

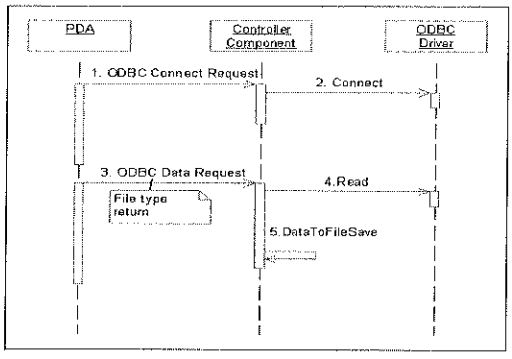
만약 다른 API를 사용하여 새로 개발하려 한다면 이는 모든 문제를 해결해 주지는 않으면서도 많은 시간과 기술적 어려움이 따르게 된다. 그러므로 본 논문에서는 서버와 PDA클라이언트의 제어를 담당하는 컴포넌트를 [그림1]처럼 만들어서 문제를 해결하려고 하는데 이것 역시 모든 문제를 해결 할 수는 없지만 기본적인 요구사항을 컴포넌트로 만들어 재사용 하므로 서버와 PDA의 개발과정에서 시간과 노력을 절감할 수 있다.



[그림 1] 제어컴포넌트 아키텍처

제어컴포넌트가 기본적으로 지원해야 하는 것들은 서버측에서 사용하는 ODBC(Open Database Connectivity), MDI(Multiple Document Interface), OLE등으로 PDA에서 사용 할 수 없는 것을 제어컴포넌트의 게이트웨이 역할을 통해 서버와 PDA사이에 정보를 교환 할 수 있도록 한다.

이 연구는 서버측에서 ODBC를 사용할 경우 PDA의 API함수들이 ODBC를 지원하지 않으므로 기존의 서버에 존재하는 ODBC Driver에 접속하여 데이터들을 사용할 수 없다. 하지만 [그림2]와 같이 제어컴포넌트를 사용하여 서버에 있는 데이터들을 텍스트 형태로 변환하면 PDA용 API는 무선네트워크를 통해 접근하여 자료를 사용할 수 있게 된다.



[그림2] 제어컴포넌트의 ODBC해결

제어컴포넌트의 역할은 실시간 적으로 PDA와 서버의 ODBC를 중개하여 마치 PDA가 ODBC Driver를 사용하는 것처럼 할 수 있게 서버측 ODBC의 데이터를 File로 변경하고 이것을 PDA가 접근하여 사용할 수 있게 한다.

[그림3]은 실제 PDA가 서버측 제어컴포넌트에 접근하는 방법을 구현한 코드이다.

```
Void ODBCConnect()
{
        TCHAR szUNCPath[MAX_PATH +1];
        TCHAR szLocalName[MAX_PATH +1];
        NETRESOURE nr;
        if(!GetTextResponse(_T("Enter UNC to Connect to: "),
                szUNCPath, MAX_PATH))
                return;
        if(!GetTextResponse(_T("Enter local Name: "),
                szLocalName, MAX_PATH))
                return;
        nr.dwType = RESOURCETYPE_DISK;
        nr.lpRemoteName= szUNCPath;
        nr.lpLocalName = szLocalName;
        nr.lpProvider =NULL;
        if(WNetAddConnection3(hwnd,&nr,NULL,NULL,
                CONNECT_UPDATE_PROFILE) != NO_ERROR)
                cout<< _T("Error adding connection: ")
                        <<GetLastError() << endl;
}
```

[그림3] 제어 컴포넌트 구현 코드

5. 결론 및 향후 연구 과제

본 연구에서는 기존의 C/S환경에 PDA를 사용하기 위해 제어 컴포넌트를 추가하여 편리하게 서버측의 클라이언트 프로그램을 PDA로 마이그레이션 하여 개발과정 축소 및 시간을 절약을 할 수 있으며, 또한 다른 개발에서도 제어컴포넌트를 재사용 하여 보다 편리하게 일반 데스크탑의 기능을 PDA서 사용할 수 있도록 하였다.

향후 연구과제로는 제어컴포넌트에 PDA에서 지원하지 않는 MDI,OLE,DAO 같은 기능을 PDA에서 사용할 수 있도록 제어컴포넌트를 확장하는 연구가 계속 되어야 할 것이다.

참고문헌

[1] Nic Grattan, Marshall Brain , Windows CE3.0(Application Programming), Prentice Hall PTR, January, 2001.

[2] modream corporation http://www.modream.co.kr

[3] Mobile Agent & Computing http://my.netian.com/~yuncap/mobile.htm.

[4] Microsoft WindowCE, mobile solutions http://www.microsoft.com/windows/embedded/ce/default.asp.