

GIS 구축을 위한 사용자 인터페이스 개발에 관한 연구[†]

(A Study on Developing an User Interface for GIS Construction)

남인길* 부기동**
(In-Gil Nam) (Ki-Dong Bu)

요약 본 논문에서는 로컬 서버 자동화를 이용한 GIS의 사용자 인터페이스를 개발하는 방법을 제시하였다. 사용자 인터페이스의 개발에 있어서는 윈도우즈 프로그래밍 기술과 컴포넌트 소프트웨어 제작 기술의 적용이 중요하다. 본 논문에서는 컴포넌트 소프트웨어 제작 기술인 로컬 서버 자동화와 비주얼 베이직 언어를 이용하여 사용자 인터페이스를 개발하는 방법을 제시하고, 사례 연구로서 주제도 중첩, 속성 테이블 조회, 그래프 분석, 분포도 분석 등의 기능을 수행할 수 있는 사용자 인터페이스를 개발하였다.

Abstract This paper proposes an easy method to develop user interfaces for the GIS using local server automation. In the developing stage of user interface, the most important thing is to apply effective windows programming techniques and component software supporting techniques. This study shows an user interface developing method using the local server automation and Visual Basic programming. For a case study, the study constructs its user interface which performs map overlaying, referencing attribute tables, graph analysis, drawing up of thematic map.

1. 서론

최근 대용량의 다양한 지리 자료들을 효율적으로 저장 및 검색하고 분석할 수 있는 GIS의 구축이 활발하게 이루어지고 있다. 아울러 그래픽 환경에서 방대한 지리 자료들을 손쉽게 접근할 수 있도록 해주는 GIS의 사용자 인터페이스의 개발은 GIS 구축에 있어서 필수적인 과제로서 대두되고 있다.

이러한 사용자 인터페이스 개발에 있어서는 GIS의 운용에 필요한 기능을 효과적으로 프로그래밍 할 수 있는 윈도우즈 프로그래밍 기술과 컴포넌트 소프트웨어 제작 기술의 적절한 연동이 중요하다. 왜냐하면 사용자가 단순한 윈도우즈 프로그래밍만으로 GIS에서 필수적인 첨단 매핑 기능과 각종 분석 기능을 수용할 수 있는 인터페이스를 개발

하는 것은 용이한 일이 아니기 때문이다.

따라서 사용자 인터페이스의 효율적인 개발을 위해서는 객체지향 기술을 기반으로 하는 컴포넌트 소프트웨어 제작 기술을 적용하는 것이 바람직하다. 컴포넌트 소프트웨어 제작 기술은 다양한 컴포넌트들을 필요할 때마다 호출하여 사용할 수 있도록 객체 공유를 위한 표준 플랫폼을 제공해주기 때문에 이를 윈도우즈 프로그램 작성 과정에서 효과적으로 활용함으로써 프로그램 코딩에 필요한 인적, 시간적 경비를 대폭 절감할 수 있다[1].

최근에 주목을 받고 있는 대표적인 컴포넌트 소프트웨어 제작 기술로는 Microsoft사의 OLE /COM(Object Linking & Embedding/Component Object Model)과 CILAB의 OpenDoc, OMG의 CORBA, Java Applet 등을 들 수 있다. 이 중에서 특히, Microsoft사의 OLE/COM은 비주얼 언어와 윈도우를 기반으로 하는 개인용 운영체제의 확산에 힘입어 날이 갈수록 그 사용 범위가 확산되고 있으며, 접근의 용이성과 편리성 등의 측면에서도 그 가치가 높게 인정받고 있다.

† 이 논문은 1999학년도 대구대학교 학술 연구비 지원에 의한 논문임.

* 대구대학교 컴퓨터정보공학부 교수

** 경일대학교 컴퓨터공학과 부교수

본 논문에서는 컴포넌트 소프트웨어 제작 기술인 OLE/COM 환경 하에서 로컬서버를 자동화하는 방법을 사용함으로써 GIS 구축 시에 필수적인 사용자 인터페이스를 개발하는 방법을 제시하고, 사용자 인터페이스를 사례로서 직접 구현해 봄으로써 그 결과를 평가하고자 한다.

2. 기술배경

2.1 OLE/COM

Microsoft사에서는 OLE 1.0 기술을 1990년에 발표한 이래, 1993년에는 OLE 2.0, 1995년에는 OLE 기술을 확장한 ActiveX 기술을 발표한 바가 있다. 이러한 OLE 혹은 ActiveX 기술은 당초에는 서로 다른 응용프로그램들이 생산해 내는 다수의 정보 소스 객체를 모아 하나의 문서, 즉 복합 문서를 만들어 내기 위해서 개발되었지만, 현재는 이러한 기능들이 데이터뿐만 아니라 함수까지도 컴포넌트로서 공유할 수 있는 총체적인 컴포넌트 소프트웨어 제작 기술로 발전하게 되었다[1][2].

이러한 OLE의 인터페이스들을 정의하고 명세하는 Microsoft사의 표준이 COM이라고 할 수 있다[3]. 따라서 OLE는 COM을 기반으로 하여 만들어진 COM 객체간의 연결 및 내장을 구현하는 상위 기술이 된다.

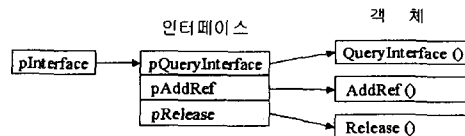
전통적으로 애플리케이션은 파일, 모듈, 혹은 클래스로 분리된다. 이 것들은 하나의 애플리케이션을 형성하기 위해서 컴파일되고 링크된다. 컴포넌트는 컴파일되고, 링크되고 사용될 수 있는 코드의 2진 파일 형태로 구성된다. 접점에서 애플리케이션과 동일하지만 애플리케이션은 여러 개의 컴포넌트로 분리될 수 있다. 즉, COM은 한 덩어리의 애플리케이션을 여러 개의 컴포넌트로 분리할 수 있는 도 구이며, 컴포넌트로부터 애플리케이션을 구축하는 방법에 대한 명세를 제공해 준다. COM은 객체를 정의하고, 관련된 함수들을 인터페이스로 묶을 수 있도록 해준다. 인터페이스는 사용자가 이름을 짓고 등록할 수 있으며 만들어진 COM 객체는 COM 클래스로부터 인스턴스를 생성하여 만들어지며, 고유한 인스턴트 데이터와 결합하게 된다.

이러한 COM 객체간의 통신은 인터페이스를 통하여 이루어지며, 클라이언트에게 컴포넌트는 인터페이스의 집합이다. 즉, 클라이언트는 오직 인터페이스만을 통해서 COM 컴포넌트와 통신한다. 모든 COM 객체는 IUnknown이라는 인터페이스를 반드시 구현해야 하는데 그 이유는 이 IUnknown이 객체의 수명주기(Life Cycle)을 관리하며, 함수에 대한 객체를 조회하는 기능을 갖고있기 때문이다. 클라이언트는 COM 객체에게 특정한 인터페이스를 지원하는지 물을 수 있으며, 만일 지원이 가능하다면 인터페이스

의 포인터를 클라이언트에게 반환한다. 따라서 COM하에서 작성되는 모든 인터페이스는 IUnknown이라는 인터페이스를 상속받는다. IUnknown의 정의는 Win32 SDK의 UNKNOWN.H라는 헤더 파일에 포함되며 IUnknown의 선언은 다음과 같다.

```
interface IUnknown
{
    virtual HRESULT __stdcall
    QueryInterface(const IID& iid,
        void **ppv) = 0 ;
    virtual ULONG __stdcall AddRef() = 0 ;
    virtual ULONG __stdcall Release() = 0 ;
};
```

위 선언에서 보듯이 IUnknown은 QueryInterface와 AddRef, Release라는 세 개의 함수를 정의한다. 클라이언트는 컴포넌트가 임의의 인터페이스를 지원하는지 여부를 결정하기 위해서 QueryInterface라는 함수를 호출한다. 컴포넌트가 특정 인터페이스를 지원한다면 QueryInterface는 그 인터페이스에 대한 포인터를 리턴한다. 그리고 클라이언트는 다른 인터페이스를 요청하거나 그 컴포넌트를 언로드 시킬 수 있다.



<그림 1> 포인터 테이블로 구현된 COM 인터페이스

<그림 1>에서는 인터페이스의 메소드가 객체 내에서 구현된다는 사실과 이러한 메소드들에 대한 함수 포인터를 사용해서 테이블을 구성함으로써 멤버에 액세스 할 수 있다는 사실을 보여준다.

2.2 로컬서버의 자동화

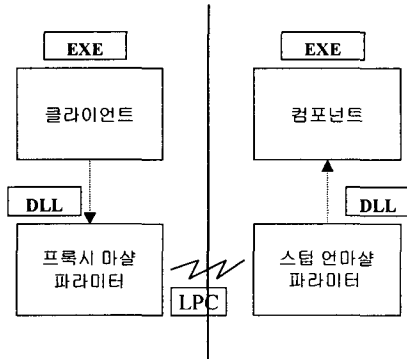
전술한바와 같이 COM 인터페이스는 DLL 혹은 EXE의 내부에 구현할 수 있다. 여기서 로컬 서버란 인터페이스가 포함된 EXE를 의미한다. 클라이언트가 로컬서버에게 인터페이스를 통해 컴포넌트를 요청하면 COM이 반환하는 컴포넌트는 또 다른 프로세스 내에서 사용될 수 있다. 즉 클라이언트가 인터페이스 포인터를 통해 직접적으로 객체의 함수를 호출할 수 있으며, 서버는 그 데이터를 액세스할 수 있고 변경할 수도 있다. 단지 서버는 윈도우즈 운영체제의 레지스트리를 통해 자신이 특정 객체를 공급하는 로컬 서버임을 알려야 한다. 레지스트리의 LocalServer32

키는 다음과 같이 로컬서버에 대한 패스를 포함하고 있다.

```
HKEY-CLASS_ROOT\Some CLSID\
LocalServer32="C:\some-pass\my-server.exe"
```

EXE에 있는 메소드를 호출할 경우에는 클라이언트의 주소 영역의 함수에 넘겨줄 파라미터를 컴포넌트의 주소 영역으로 옮겨주는 문제가 가장 중요하다. 이러한 행위는 마샬링(marshaling)이라고 하며 어떤 프로세스에서 다른 프로세스로 파라미터를 복사하는 일을 책임진다. 컴포넌트를 마샬링하기 위해서는 IMarshal이라는 이름을 가진 인터페이스를 구현해야 한다. COM은 컴포넌트 생성의 일부로써 그 컴포넌트에 IMarshal을 요구한다. 그 후 함수의 호출 전이나 후에 파라미터를 마샬 혹은 언마샬하기 위해 IMarshal에 있는 멤버 함수를 호출한다.

또한, COM에서는 프로세스간의 통신을 하고자 할 때 LPC(Local Procedure Call)를 사용한다. 클라이언트는 컴포넌트처럼 동작하는 프록시(proxy) DLL과 통신을 하며, 프록시 DLL은 클라이언트를 위해 마샬링을 하고 LPC를 호출하는 역할을 담당한다. LPC를 통해 프록시는 클라이언트에서 전송된 데이터를 언마샬하기 위한 스텝(stub)이라는 이름의 DLL을 요구한다. 스텝은 또한 컴포넌트를 클라이언트로 되돌리기 위해서 데이터를 마샬링한다. 그림 2는 이러한 LPC를 이용한 마샬링 과정을 보여준다.



<그림 2> COM에서 LPC를 이용한 마샬링 과정

3. GIS를 위한 사용자 인터페이스의 개발 방법

3.1 사용자 인터페이스 개발의 필요성

GIS는 지리 형태에 관한 자료를 수집, 저장하고 이를 분석 및 가공하여 각종 지리 관련 응용분야에서 활용할 수 있는 정보를 제공하는 컴퓨터 소프트웨어 시스템이라 할 수 있다. GIS에서는 대용량의 다양한 지리자료들이 벡터

지도, 래스터 이미지, 속성자료 등의 형태로 데이터베이스 내부에 탑재되게 된다. 이러한 다양한 형태의 지도자료와 속성 데이터베이스를 연계 시켜서 강력한 공간분석을 행할 수 있는 사용자 인터페이스는 GIS 응용에 있어서 필수적인 도구라고 할 수 있다[4].

그러나 대부분의 실무현장에서는 사용자 인터페이스 개발에 소요되는 막대한 인적, 경제적 손실 때문에 상업적으로 유통되고 있는 GIS 패키지를 사용하여 GIS를 구축한다. 이러한 패키지 자체를 사용자 인터페이스에 대신하여 사용하는 경우가 많다. 하지만 이러한 GIS 패키지들은 사용자의 업무 현실과 요구 조건에 맞는 인터페이스를 제공해 주지 못 할 뿐만 아니라 본격적인 개발 도구로서 그 기능도 복잡하고 사용법도 까다로우며 실무 부서의 사용자가 이를 배워서 응용하기에는 많은 어려움이 따른다.

뿐만 아니라 출력 양식을 현장의 실정에 맞도록 변경하거나 속성 데이터베이스의 디스플레이 화면을 임의로 변경하고자 할 경우, 기존의 GIS 패키지에서 사용자의 요구에 부합하는 기능들을 지원할 수 없기 때문에 불편을 느끼게 된다.

따라서 필요 없이 복잡한 기능들을 배제하고 현장에서 필요한 기능들만을 포함시킴으로써 보다 조작이 쉽고 간편하며, 필요할 경우에는 GIS 도구에서 지원하지 않는 기능들도 추가하여 효율성을 높일 수 있는 사용자 인터페이스를 직접 개발하여 사용하는 것이 바람직하다. 본 연구에서는 이러한 취지에서 컴포넌트 소프트웨어 제작 기술인 로컬서버 자동화 기법을 사용하여 벡터지도와 속성 정보를 연계한 맵핑/디스플레이 및 첨단 공간분석이 가능한 GIS의 사용자 인터페이스를 개발하는 방법을 제시하고, 사용자 인터페이스를 사례로서 직접 구현해 봄으로써 그 결과를 평가하고자 한다.

3.2 로컬서버 자동화를 이용한 사용자 인터페이스의 개발 방법

사용자 인터페이스의 작성에 있어서 로컬서버 자동화를 이용하기 위해서는 컴포넌트를 공급해 주는 로컬서버가 있어야 한다. GIS 응용을 위한 로컬서버는 OLE/COM 표준을 준수하는 GIS 도구이면 어떠한 제품이라도 무방하다. 상업용으로 유통되고 있는 대부분의 데스크톱 제품들은 로컬서버로서의 표준을 준수하고 있기 때문에, 현재의 GIS를 구축하기 위해서 사용한 패키지를 로컬서버로서 사용하면 된다. 본 논문에서는 데스크톱 전용 GIS 패키지인 MapInfo 4.0 [5][6]을 로컬서버로 사용하여 인터페이스 프로그래밍 방법을 설명하고자 한다.

COM 환경에서 로컬서버는 IDispatch 인터페이스를 구현한 COM 컴포넌트의 집합이다. 자동화 클라이언트는

IDispatch 인터페이스를 통해 로컬서버와 통신하는 COM 클라이언트이다. 클라이언트는 로컬서버에 의해 구현된 함수를 직접 호출하지 않고 IDispatch 인터페이스의 멤버 함수를 사용한다.

예를 들어 비주얼 베이직[7]으로 로컬서버의 함수 do 를 호출하는 방법은 다음과 같다.

```
Dim mi As Object
Set mi = CreateObject(MapInfo.Application)
msg="open table " "world.tab"
mi.do
```

이 코드는 COM 컴포넌트를 만들고 컴포넌트에 구현되어있는 IDispatch 인터페이스를 통해서 함수 do를 호출한다. 이 과정에서 ProgID를 사용하여 컴포넌트를 만드는 C++ 코드는 다음과 같다.

```
// Initialize the OLE Library
HRESULT hr = OleInitialize(NULL) ;
// Get the CLSID for the application.
wchar_t progid[] = L"MapInfo.Application) ;
CLSID clsid ;
::CLSIDFromProgID(progid, & clsid) ;
// Create the component.
IDispatch* pIDispatch = NULL ;
::CoCreateInstance(clsid,
    NULL,
    CLSTX_INPROC_SERVER,
    IID_IDispatch,
    (void**)&pIDispatch) ;
```

이때, 개별적인 QueryInterface 호출을 하지 않기 위해서, CoCreateInterface에서 IDispatch 포인터를 저장한다. IDispatch 인터페이스 포인터를 가지게 되면, 디스패치 함수 do에 대한 DISPID를 구할 수 있다. IDispatch::GetIDsOfNames 함수는 이름 문자열을 가지고 다음과 같이 DISPID를 구한다.

```
DISPID dispid ;
OLECHAR* name = L"do" ;
pIDispatch->GetIDsOfNames(
    IID_NULL,
    //Must be IID_NULL
    &name
    //Name of function
    1,
    //Number of names
    GetUserDefaultLCID(),
    //Localization info
```

```
&dispid) ;
//Dispatch ID
```

DISPID는 클라이언트가 호출하기를 원하는 함수를 가리키며, 클라이언트에게 문자열 전달을 피함으로써 최적화를 수행할 수 있다. 또한, do에 대한 DISPID를 가지고 있으면, DISPID를 복잡한 함수인 IDispatch::Invoke로 넘기는 것으로 함수를 실행시킬 수 있다. 다음 코드는 Invoke 호출을 구현한 것이다. Invoke 호출에서는 어떤 파라미터도 없이 함수를 호출한다.

```
//Prepare the arguments for do.
DISPARAMS dispparamsNoArgs = {
    NULL,
    NULL,
    0, //Zero arguments
    0 //Zero named arguments
};
// Invoke call.
pIDispatch->Invoke(dispid,
//DISPID
    IID_NULL,
    //Must be IID_NULL
    GetUserDefaultLCID(),
    DISPATCH_METHOD,
//Method
    &dispparamsNoArgs,
//Method arguments
    NULL,
//Results
    NULL,
//Exception
    NULL) ;
//Arg error
```

비주얼 베이직 언어에서 IDispatch 인터페이스를 통해 MapInfo 로컬서버로부터 컴포넌트를 실행시키기 위해서는 다음과 같은 코드를 사용한다[8].

먼저, 사용자 인터페이스가 실행되면 필요한 컴포넌트를 제공해줄 로컬서버를 백그라운드로 수행될 수 있도록 호출한다. 로컬 서버로 MapInfo를 호출하였을 경우 이에 대응되는 코드는 다음과 같다.

```
Dim mi As Object
Set mi = CreateObject(MapInfo.Application)
```

로컬서버가 백그라운드로 실행된 후에 클라이언트로 부터 로컬서버에게 메시지를 전송하는 방법은 다음과 같다

```
msg="open table " "world.tab"
mi.do msg
```

그리고 로컬서버가 제공하는 대화상자나 에러 메세지들을 클라이언트에서 받아보기 위해서는 다음과 같은 코드를 사용한다.

```
mi.do "Set Application Window " &
UserControl.hWnd
```

이때, "UserControl"은 비주얼 베이직에서 사용하는 품의 명칭이 된다. 그리고 로컬서버 윈도우의 소유권을 현재의 클라이언트 윈도우에 귀속시키기 위해서 다음과 같은 코드를 추가한다.

```
mi.do "Set Next Document Parent " &
Frame1.hwnd & " Style 1"
```

위 코드는 로컬서버가 출력한 윈도우를 클라이언트의 현재 품인 "Frame1"의 자식(child) 윈도우로 설정하게 됨을 의미하고, 로컬서버의 윈도우를 경계선 없이 클라이언트의 품의 내용으로 넘겨받기 위해서 "Style 1"을 지정한 것이다.

또한, modeless 윈도우는 현재 윈도우가 열려있는 채로 다른 윈도우로 전환이 가능한 것이다. 이러한 로컬서버의 modeless 윈도우를 클라이언트가 소유하기 위해서는 다음과 같은 코드를 사용한다.

```
mi.do "Set Window "윈도우명" Parent " &
Frame1.hWnd
```

비주얼 베이직 응용에서 로컬서버의 표준 툴바 버튼 기능을 에뮬레이션하기 위해서는 "Run MenuCommand" 메소드를 사용하며 툴바 버튼의 식별번호 혹은 식별 코드를 첨가한다. 예를 들어 로컬서버의 "Zoom In" 기능을 에뮬레이션 하기 위해서는 다음과 같은 코드를 사용한다.

```
mi.RunMenuCommand 1705 혹은
mi.RunMenuCommand M_Tools_Expand
```

끝으로 클라이언트가 종료될 때 로컬서버를 자동적으로 종료되게 하기 위해서는 다음과 같은 코드를 사용한다.

```
Private Sub UserControl_Terminate()
mi.do "End MapInfo"
End Sub
```

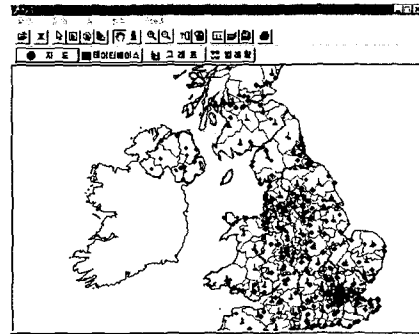
4. GUI 개발 사례연구

이 절에서는 전술한 로컬 서버와의 통신 방법을 이용하여 GIS의 표준 기능을 수행할 수 있는 사용자 인터페이스를 프로그래밍하는 방법을 기술한다. 사례로서 제시하는 지도 데이터와 속성 데이터는 MapInfo Professional 4.0

에서 제공하는 샘플 파일 중 영국지도를 이용하였다. 사용자 인터페이스의 프로그래밍 방법을 기능별로 세분하여 설명하면 다음과 같다.

4.1 지도창에서 주제도 중첩

별도의 주제도들을 중첩하여 디스플레이 하는 기능은 사용자 인터페이스의 가장 기본적인 기능이라 할 수 있다. 이를 위해서는 먼저 "테이블 열기" 기능이 필요하며 테이블을 열 때마다 지도들이 자동으로 중첩되도록 한다.



<그림 3> 주제도의 중첩

<그림 3>은 "테이블 열기" 기능으로 영국지도와 도시 및 버스 터미널을 나타내는 점지도와 행정 경계지도를 중첩시킨 후 이를 지도창을 통해 디스플레이한 화면이다.

4.2 데이터베이스 창에서 속성 자료의 조회

<그림 3>의 지도 창에서 선택한 객체에 대해서는 데이터베이스 창을 통하여 연결된 속성 정보를 확인 할 수 있다. 여러 개의 지도가 중첩된 상태에서는 사용자가 필요로 하는 데이터베이스 테이블을 선택할 수 있어야 한다.

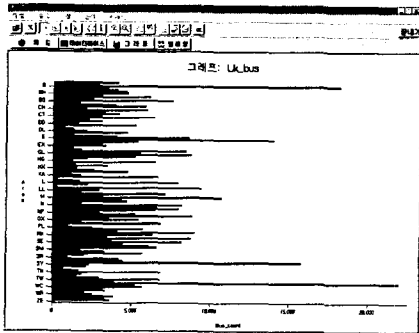
Name	Country	Longitude	Latitude
Stratford	United Kingdom	12.2825 W	51.1477 N
Northam	United Kingdom	12.6283 W	51.1487 N
Adwick	United Kingdom	12.8946 W	51.1522 N
Donkey	United Kingdom	12.9927 W	51.1577 N
Path	United Kingdom	12.9928 W	51.2114 N
Exticely	United Kingdom	12.7477 W	51.2089 N
Swing	United Kingdom	12.7387 W	51.1677 N
Falst	United Kingdom	12.8207 W	51.1717 N
Dumbarton	United Kingdom	12.8707 W	51.1841 N
Greenock	United Kingdom	12.9627 W	51.1817 N
Easthouse	United Kingdom	12.9527 W	51.1807 N
Port Glasgow	United Kingdom	12.9527 W	51.2047 N
Arnie	United Kingdom	12.9527 W	51.2047 N
Gassary	United Kingdom	12.9146 W	51.1817 N
Obanrock	United Kingdom	12.9127 W	51.2027 N
Ayr	United Kingdom	12.9727 W	51.1817 N
Ardschoy	United Kingdom	12.1115 W	51.1817 N
Blun	United Kingdom	12.7227 W	51.1817 N
Dunbar	United Kingdom	12.6127 W	51.1817 N
Widley Bay	United Kingdom	12.7227 W	51.1817 N
Lownders	United Kingdom	12.9846 W	51.1817 N
Lams	United Kingdom	12.1115 W	51.1817 N

<그림 4> 속성 DB의 조회

<그림 4>는 도시 지도와 연결된 속성 테이블을 선택하여 그 내용을 데이터베이스 창을 통해 디스플레이한 화면이다.

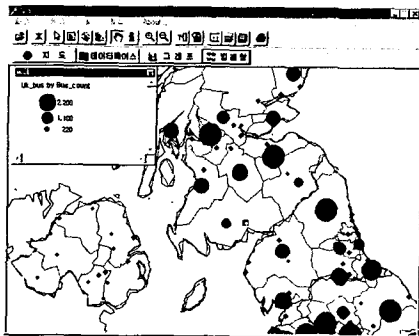
4.3 그래프창에서 그래프 분석

지도 창에서 선택된 객체들과 연결된 속성 자료들 중에서 수치형 자료는 그 값의 크기에 따라 그래프 분석이 가능하다. <그림 5>에서는 영국의 버스 터미널 당 보유하고 있는 버스 대수를 횡축으로 하고 터미널 명을 종축으로 하여 그래프 분석을 시행한 결과를 디스플레이한 결과를 보여준다.



<그림 5> 속성 DB의 그래프 분석

4.4 표준 툴바 기능의 에몰레이션



<그림 6> 분포도 분석

지금까지 살펴본 지도 창, 데이터베이스 창, 그래프 창의 처리는 탭을 눌렀을 때의 화면 전환과 테이블 리스트에서 선택한 테이블을 처리하기 위해 다소 복잡한 코딩 과정을 거쳐야 하지만 표준 툴바의 기능을 에몰레이션 하기

위해서는 로컬 서버와의 간단한 통신만으로 구현이 가능하다. 즉 표준 툴바 기능을 수행할 매소드를 호출하는 방법만 알면 손쉽게 표준 툴바 기능을 구현할 수 있다.

<그림 6>은 분포도 분석으로 버스 터미널당 버스 보유 대수를 크기 심볼로 나타낸 후 범례 버튼을 눌러 범례에 대한 modeless 윈도우가 디스플레이 된 화면이다.

5. 평가

서론에서 서술한 바와 같이 사용자가 단순한 윈도우즈 프로그래밍만으로 GIS에서 필수적인 첨단 매핑 기능과 각종 분석 기능을 수용할 수 있는 사용자 인터페이스를 개발하는 것은 용이하지 않다. 그러나 로컬 서버 자동화를 이용한 컴포넌트 소프트웨어 제작 기술을 이용하면, 초보자도 손쉽게 실무에 필요한 사용자 인터페이스를 구현할 수 있을 뿐 아니라 개발에 소요되는 인적, 경제적 비용을 상당히 절감할 수 있다.

본문에서 기술한 개발 방법을 적용하여 사례 인터페이스를 개발해 본 결과, 사용자 인터페이스의 표준 기능인 주제도 중첩, 속성 테이블 조회, 그래프 분석, 분포도 작성 등을 완벽하게 수행함을 확인할 수 있었다.

아울러 본문에서 제시한 모든 표준 툴바 기능을 수행하는 사용자 인터페이스를 완성하는데 소요된 시간은 3주일 정도에 불과했으며, 사용된 Visual Basic 코드의 분량도 350 스텝 정도에 불과하였다. 이는 컴포넌트 소프트웨어 제작 기술을 적용하는 것이 프로그래밍 과정에서의 생산성을 매우 향상시켜주는 한가지 사례라고도 할 수 있다.

그러나 사용자 인터페이스의 원형 선택 기능에서 동적으로 변화하는 반경 즉, 축적과 일치하는 현재의 선택 범위에 대한 반경 값을 로컬서버로부터 넘겨받아 실시간으로 디스플레이하는 기능을 구현하기 위해 많은 노력을 기울였으나 "OLE callback" 함수 적용에 있어 원인 불명의 에러가 발생하여 현재 이에 대한 문제 해결 방법을 계속 모색하고 있다. 이 기능은 표준 툴바에 제공하는 출자 기능과 버퍼 기능의 반경 값을 이용한 범위 선택 기능으로 대신 할 수 있기 때문에 사용자의 입장에서서는 그다지 불편하지 않을 것으로 보인다.

6. 결론

GIS 구축을 위한 사용자 인터페이스 개발에 있어서는 GIS의 운용에 필요한 기능을 효과적으로 프로그래밍 할 수 있는 윈도우즈 프로그래밍 기술과 컴포넌트 소프트웨어

제작 기술의 연동이 중요하다. 본 연구에서는 초보자라도 손쉽게 개발할 수 있는 컴포넌트 소프트웨어 제작 기술로서 로컬 서버 자동화와 비주얼 베이직 프로그래밍 기법을 사용하여 GIS의 사용자 인터페이스를 개발하는 방법을 제시하였다.

아울러 개발과정을 상세히 설명하기 위하여 사례 시스템의 구축과정을 프로그램 코드와 함께 제시하였다. 그 결과, 구현된 사용자 인터페이스는 GIS의 표준 기능인 주제도 중첩, 속성 테이블 조회, 그래프 분석, 분포도 작성 등을 완벽하게 수행함을 확인할 수 있었다.

본 논문의 후속연구는 또 다른 컴포넌트 소프트웨어 제작 기술인 OpenDoc과 Cyber Dog 등과 다양한 OLE 서버를 사용하여 GIS의 사용자 인터페이스를 개발하는 방법을 연구하는 것이다.

참 고 문 헌

- [1] 강민석, "윈도우즈 세계에 부는 OLE 바람", 마이크로 소프트웨어, pp. 177-187, 1994.
- [2] 이이표, 김병세, Inside ActiveX & OLE 실무 프로그래밍, 삼양출판사, 1998.
- [3] 송혜원, 곽준기, Inside COM, 삼각형, pp. 246-249, 1997.
- [4] 부기동, 조명희, 김광주, 서준석, "GIS를 이용한 은행 마케팅 DB의 설계 및 구현", 한국 지리정보학회 추계 학술발표대회 논문집, pp. 95-104, 1998.
- [5] MapInfo Co., MapInfo Professional 4.0, 사용자설명서, 1995.
- [6] MapInfo Co., MapInfo Professional 4.0, 사용자참고서, 1995.
- [7] 주경민, 박성완, 정동길, Visual Basic Programming Bible Ver 5, 영진출판사, 1997.
- [8] MapInfo Co., MapBasic professional, 사용자설명서, 1995.



부 기 동

- 1984년 경북대학교 전자공학과졸업 (전자계산기전공 공학사)
- 1988년 경북대학교 대학원 전자공학과 (전산공학전공 공학석사)
- 1996년 경북대학교 대학원 전자공학과 (전산공학전공 공학박사)

1983년~1985년 포항종합제철 시스템개발실
 1988년~현재 경일대학교 컴퓨터공학과 부교수
 관심분야 : 데이터베이스, GIS



남 인 길

- e-mail: ignam@biho.taegu.ac.kr
- 1978년 경북대학교 전자공학과 (공학사)
- 1981년 영남대학교 대학원 전자공학과계산기전공 (공학 석사)
- 1992년 경북대학교 대학원 전자공학과 전산공학 전공(공학박사)

1978년~1980년 대구은행 전산부
 1980년~1990년 경북산업대학 전자계산학과 부교수
 1990년~현재 대구대학교 컴퓨터정보공학부 교수
 1996년~1997년 미국 루이지애나 주립대학 교환 교수
 관심분야 : 데이터베이스, GIS, 이동컴퓨팅