

OLE 자동화를 이용한 GIS의 사용자 인터페이스 개발에 관한 연구

부기동¹

A Study on Developing GIS User Interface using OLE Automation

Ki-Dong Bu¹

요 약

본 논문에서는 OLE 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 OLE 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.

주요어: GIS, 사용자 인터페이스, OLE 자동화

ABSTRACT

This study proposes an easy method to develop user interfaces for the GIS using OLE automation. In the developing stage of user interface, the most important thing is to make the best use of effective windows programming techniques and component software supporting techniques. Using the OLE automation and Visual Basic programming, the study constructs an user interface as case study which performs map overlaying, referencing attribute tables, graph analysis, drawing up of thematic map.

KEYWORDS : GIS, User Interface, OLE Automation

서 론

최근 대용량의 다양한 지리 자료들을 효율적으로 저장 및 검색하고 분석할 수 있는 GIS의 구축이 활발하게 이루어지고 있다. 아

울러 그래픽 환경에서 방대한 지리 자료들을 손쉽게 접근 할 수 있도록 해주는 GIS의 사용자 인터페이스의 개발은 GIS 구축에 있어서 필수적인 과제로서 대두되고 있다.

1999년 3월 29일 접수 Received on March 29, 1999
1 경일대학교 컴퓨터공학과(kdbu@bear.kyungil.ac.kr)

이러한 GIS의 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 개발하는 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(송혜원과 곽준기 1997).

대부분의 실무현장에서는 사용자 인터페이스 개발에 소요되는 막대한 인적, 경제적 손실 때문에 상업적으로 유통되고 있는 GIS 패키지를 사용하여 GIS를 구축한 다음, 이러한 패키지 자체를 사용자 인터페이스에 대신하여 사용하는 경우가 많다.

그러나 이러한 GIS 패키지들은 사용자의 업무 현실과 요구 조건에 맞는 인터페이스를 제공해 주지 못 할 뿐만 아니라 본격적인 개발 도구로서 그 기능도 복잡하고 사용법도 까다로워 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 OLE 자동화(OLE automation)를 사용하여 사용자 인터페이스를 개발하는 방법을 제시하고, 사용자 인터페이스를 사례로서 직접 구현해 봄으로써 그 결과를 평가하고자 한다.

OLE 자동화의 기술배경

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 함수(메소드) 까지도 컴포넌트로서 공유할 수 있는 총체적인 컴포넌트 소프트웨어 제작 기술로 발전하게 되었다(이이표와 김병세 1998).

OLE 기술은 윈도우즈 환경에서 객체들을 공유하기 위한 COM 기술을 기반으로 하고 있다(송혜원과 곽준기 1997). COM은 윈도우즈 환경에서 객체를 정의하고, 객체를 구성하는 방법 및 함수를 호출하는 방법에 대한 일관적인 표준을 제공해주고, 서버 프로그램과 클라이언트 프로그램간에 상호 통신을 통해 OLE 컴포넌트를 공유할 수 있도록 해주는 객체 기반 기술의 명칭이다. 따라서 OLE 컴포넌트를 공유하고자 하는 응용 프로그램들은 OLE/COM에서 제공하는 표준을 따라서 작성해야 한다.

OLE 자동화는 이러한 OLE/COM의 표준 하에서 작성된 서버 프로그램의 객체들을 호출하기 위한 문장을 클라이언트 프로그램 내에 삽입

함으로써 서버로부터 필요한 메소드를 실행시키고 클라이언트에게 그 결과 값을 넘겨 줄 수 있는 OLE/COM 기술의 한가지 적용 분야라 할 수 있다.

2. OLE 자동화

OLE 자동화는 이러한 OLE/COM 기술의 일부분으로서 응용 프로그램을 작성할 때 필요한 기능을 일일이 코딩하지 않고 자동화 서버에서 제공해주는 각종 함수들을 실행함으로써 손쉽게 클라이언트 소프트웨어를 작성할 수 있게 해주는 기술이다. 따라서 사용자 프로그램은 OLE 자동화 서버로부터 필요한 함수를 호출하여 그 결과 값을 반환 받는 클라이언트로서의 역할을 수행한다(강민석, 1994).

다음의 그림 1은 OLE 라이브러리를 통한 OLE 자동화 서버와 클라이언트 프로그램간의 상호통신을 위한 구조를 나타낸 것이다. 그림에서 보는 것처럼 클라이언트 프로그램 사용자가 서버 프로그램에서 제공하고 있는 내장 함수의 기능을 이용하고 싶을 때는 서버 프로그램의 인터페이스를 호출하여 내장 함수의 다양한 서비스를 받을 수 있다. 반대로 서버 프로그램이 처리 결과를 반환하기 위해 필요할 경우에는 클라이언트 프로그램의 인터페이스를 호출하여 클라이언트가 가지고 있는 내장 함수를 호출하게 된다. 이와 같은 인터페이스를 통한 상호 호출은 중간에 OLE 라이브러리를 경유하여 진행된다.

따라서 OLE 프로그램들이 제공하는 OLE 기능은 이 인터페이스와 라이브러리를 통하여 구현된다고 할 수 있다. 인터페이스는 OLE 프로그램이 다른 OLE 프로그램이 기능을 호출할 때 필요한 함수들을 의미상으로 비슷한 것끼리 모아 놓은 것이지만, 함수들을 모아 놓았다고 해서 실제 기능이 구현된 코딩까지 들어 있는 것은 아니고, 단순한 함수 포인터 리스트에 불과하다.

OLE의 인터페이스는 단순한 함수 이름의 리스트이므로 누구나 함수에 대한 기능 코딩을 제공할 수 있고, 어떤 프로그램이라도 그 인터페

이스를 사용할 수 있어 개방성이 뛰어나다. 또한 기존의 인터페이스는 건드리지 않고 새로운 인터페이스를 추가함으로써 쉽게 기능 확장을 할 수 있으므로 확장성 또한 뛰어나다.

OLE 인터페이스에 대한 구현은 DLL이나 EXE 확장자를 갖는 형태로 라이브러리에 존재하게 된다. 즉, OLE 자동화 서버는 윈도우 레지스트리에 DLL이나 EXE로 등록된 함수들을 라이브러리를 통해 제공하며, 클라이언트에서는 이를 컴포넌트로서 활용하게 되는 것이다.

OLE 라이브러리에서는 각 인터페이스에 대한 내장된 코드를 제공하므로 프로그램에서 따로 인터페이스에 대한 코딩을 하지 않고 포인터를 사용하여 인터페이스를 구현할 수 있으며, OLE가 제공하는 인터페이스의 기능이 적합하지 않을 때는 프로그램 자신의 특별한 인터페이스를 새로 추가하거나 또는 아예 완전히 바꿀 수도 있다.

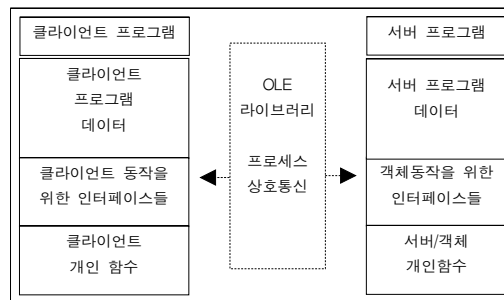


FIGURE 1. Mutual communication under the OLE automation

사용자 인터페이스의 개발 방법

이 장에서는 사용자 인터페이스 개발 방법을 인터페이스의 설계와 OLE 자동화 서버와의 통신 방법, 그리고 인터페이스 프로그래밍 단계로 구분해서 기술하고자 한다.

1. 사용자 인터페이스의 설계

사용자가 환경에 적합한 사용자 인터페이스를 설계 할 때 중요하게 다루어져야 할 사

항은 사용자 인터페이스가 갖추어야 할 기능에 대한 정확한 명세를 도출하는 일이다. 나중에 잘 사용되지 않을 불필요한 기능들은 가급적 배제하고 업무에 필수적인 기능들이 빠르게 동작할 수 있도록 정확한 기능 명세서를 작성하여야 한다. GIS 구축을 위한 사용자 인터페이스의 대표적인 기능들을 나열하면 다음과 같다(조명희 등, 1998).

1) 중첩 및 매핑 디스플레이 기능

사용자 인터페이스의 장점은 사용자에게 테이블 자료들을 다양한 형태로 보여 줄 수 있다는 것이다. 사용자가 자신의 데이터를 지도나 데이터베이스 창, 혹은 그래프로도 보고자 할 때, 이를 효과적으로 지원해주는 매핑 디스플레이 기능은 GUI의 필수적 기능이라 할 수 있다. 지도가 중첩되어 디스플레이 된 상태에서 연결된 데이터베이스 내용 보기, 확대, 축소 기능, 원형 및 사각형 선택, 지도창 드래그, 속성정보보기, 속성정보찾기 기능 등이 여기에 포함된다.

2) 계층제어 기능

여러 주제도가 중첩되어 있는 상태에서 각각의 주제도를 보이게 혹은 안보이게 할 수도 있고, 주제도를 개별적으로 수정가능 혹은 수정 불가능하게 할 수 있는 기능이다. 또한, 계층의 추가 및 삭제 기능, 심볼 표시하기, 계층의 배율 조정, 라벨링 등의 기능이 여기에 포함된다.

3) 데이터 분석을 위한 분포도 작성 기능

사용자 인터페이스에서는 공간데이터와 연결된 속성 데이터의 양식이나 경향을 알기 쉽게 분포도로 나타내주는 기능이 필수적이다. 속성 데이터의 성격이나 분석 방법에 따라 범위, 막대도표, 파이도표, 크기심볼, 점밀도, 개별값 등 다양한 종류의 분포도를 작성할 수 있어야 한다.

4) 버퍼링에 의한 지리적 공간분석 기능

한 객체에 대한 반경으로 버퍼를 설정해 놓고 버퍼 내에 속하는 객체들에 대한 지리적 공간 분석을 가능하게 하는 기능이다. 즉 지도상

의 한 지점을 중심으로 반경 내에 속하는 다른 객체에 대한 정보 검색을 가능하게 하며, 필요할 경우에는 통계 분석, 분포도 작성 등을 병행하여 공간 분석이 가능하도록 하는 기능이다.

5) 데이터베이스 질의 기능

사용자 인터페이스에서 데이터베이스 질의어를 사용하여 저장된 데이터를 효율적으로 검색하고 분석하는 기능은 매우 중요하다. 사용자가 수준 높은 데이터 분석을 하기 위해서는 테이블 간의 조인(join), 그룹화(group by), 정렬(order by), 집계함수(aggregation function) 등 기존의 데이터베이스 시스템에서 사용하는 질의 기능들을 모두 사용할 수 있어야 한다.

6) 속성 데이터베이스의 통계치 계산기능

속성 데이터베이스의 수치형 칼럼에 대해 통계치 계산을 수행하거나 수치지도에서 선택된 객체의 데이터베이스를 검색하여 자동으로 통계 처리 할 수 있는 기능이다.

사용자 인터페이스가 갖추어야 할 기능들을 수용하는 표준 툴바의 작성 사례는 그림 2와 같다.



FIGURE 2. An example of the standard tool bar

또한, 표준 툴바를 수용하는 윈도우즈 프레임의 작성 사례는 그림 3과 같다. 그림 3에서는 그림 2에서 사례로 예시한 표준 버튼들을 윈도우 상단에 한 줄로 배치시켜 보았다.

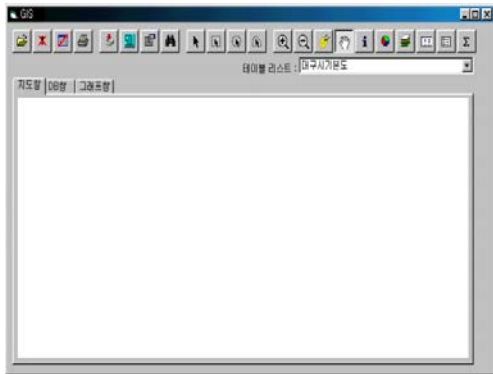


FIGURE 3. An example of the windows frame

윈도우에는 기본적으로 세개의 판넬이 노트 탭 형식으로 설정되며, 각 탭은 지도 창과 데이터베이스창 그리고 그래프 창으로 구성된다. 지도 창은 “테이블 열기”로 연 백터지도를 볼 수 있는 지도 뷰어(viewer) 역할을 담당하며, 데이터베이스 창은 지도와 연결된 속성 데이터베이스의 브라우저 역할을, 그리고 그래프 창은 속성 데이터베이스의 수치 칼럼에 대한 그래프 분석한 결과를 디스플레이하는 역할을 담당하게 된다.

따라서 한 개의 테이블을 열게되면 지도와 속성 데이터베이스가 함께 열리게 되고 지도창과 데이터베이스 창의 노트 탭을 선택함으로써 지도 혹은 데이터베이스를 부라우징 할 수 있다.

화면 우측 상단에 있는 “테이블 리스트”는 콤보 박스로서 현재 열려 있는 테이블의 목록을 보여준다. 만약 여러 개의 테이블을 열 경우 “테이블 리스트”에서 목록을 선택하여 지도창과 데이터베이스 창의 내용을 선택한 지도와 속성데이터베이스로 전환 할 수 있다.

2. OLE 서버와의 통신 방법

사용자 인터페이스의 작성에 있어서 OLE 자동화를 이용하기 위해서는 컴포넌트를 공급해 주는 OLE 서버를 결정하여야 한다. GIS 응용을 위한 OLE 서버는 OLE/COM 표준을 준수하는 GIS 도구이면 어떠한 제품이라도 무방하다. 상업용으로 유통되고 있는 대부분의 데스크톱 제품들은 OLE 서버로서의 표준을 준수하고 있기 때문에, 현재의 GIS를 구축하기 위해서 사용한 패키지를 OLE 서버로서 사용하면 된다.

본 논문에서는 데스크톱 전용 GIS 패키지인 MapInfo(4.0)(MapInfo 사용자 설명서, 1995)을 OLE 서버로 사용하고 비주얼 베이직 언어를 이용한 인터페이스 프로그래밍 방법을 설명하고자 한다. 제품마다 다소의 차이는 있을 수 있지만 OLE 자동화를 이용하여 OLE 서버와 통신하는 방법(MapBasic 사용자 설명서, 1995)은 근본적으로 대동소이하다고 할 수 있다.

1) OLE 서버의 백그라운드 호출

먼저, 사용자 인터페이스가 실행되면 필요한 메소드를 제공해줄 OLE 서버를 백그라운드로 수행될 수 있도록 호출한다. OLE 서버로 MapInfo를 호출하였을 경우 이에 대응되는 비주얼 베이직(주경민 등, 1997) 코드는 다음과 같다.

```
Dim mi As Object
```

```
Set mi = CreateObject(MapInfo.Application)
```

2) OLE 서버에게 메세지 전송

클라이언트로부터 OLE 서버에게 메세지를 전송하는 방법은 다음과 같다

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

```
mi.do msg
```

3) 클라이언트에서 OLE 서버 윈도우의 소유

OLE 서버가 실행된 이후에 OLE 서버가

제공하는 대화상자나 에러 메시지들을 클라이언트에서 받아보기 위해서는 다음과 같은 코드를 사용한다.

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

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

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

위 코드는 OLE 서버가 출력한 윈도우를 클라이언트의 현재 폼인 "Frame1"의 자식(child) 윈도우로 설정하게 됨을 의미하고, OLE서버의 윈도우를 경계선 없이 클라이언트의 폼의 내용으로 넘겨받기 위해서 "Style 1"을 지정한 것이다.

4) 클라이언트에서 OLE 서버의 modeless 윈도우의 소유

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

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

5) 클라이언트에서 OLE 서버의 툴바 버튼 기능에 대한 에뮬레이션

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

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

6) OLE 서버의 종료

클라이언트가 종료될 때 OLE 서버를 자동적으로 종료되게 하기 위해서는 다음과 같은 코드를 사용한다.

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

3. 사용자 인터페이스의 프로그래밍

이 절에서는 전술한 OLE 자동화 서버와의 통신 방법을 이용하여 GIS의 표준 기능을 수행할 수 있는 사용자 인터페이스를 프로그래밍하는 방법을 기술한다. 사례로서 제시하는 지도 데이터와 속성 데이터는 MapInfo Professional 4.0에서 제공하는 샘플 파일을 이용하였다.

사용자 인터페이스의 프로그래밍 방법을 기능별로 세분하여 설명하면 다음과 같다.

1) 지도창에서 주제도 중첩

별도의 주제도들을 중첩하여 디스플레이하는 기능은 사용자 인터페이스의 가장 기본적인 기능이라 할 수 있다. 이를 위해서는 먼저 "테이블 열기" 기능이 필요하며 테이블을 열 때마다 지도들이 자동으로 중첩되도록 한다. 또한 테이블명이 우측 상단의 "테이블 리스트" 콤보 박스에 추가되도록 한다.

이때, "테이블 열기"에 대한 프로그램 코드는 다음과 같다.

```
CommonDialog1.DialogTitle = "테이블 열기"
CommonDialog1.Filter = "MapInfo 파일*.TAB"
CommonDialog1.ShowOpen
If CommonDialog1.FileTitle <> "" Then
    TableName = mi.eval("PathToTableName$( "" &
        CommonDialog1.filename & "" )")
    mi.do "Open Table """
```

```

&CommonDialog1.filename & "" As " &_
TableName
ExistMapper = mi.eval("TableInfo(0,5)")
If ExistMapper = "T" Then
  If NewMapper Then
    mi.do "Set Next Document Parent "
    & Frame1(0).hWnd & " Style 1"
    mi.do "Map From " & TableName
    MapperID = mi.eval("FrontWindow()")
    NewMapper = False
  Else
    mi.do "Set Next Document Parent "
    & Frame1(0).hWnd & " Style 1"
    mi.do "Add Map Window " &
    MapperID & " Layer " & TableName
    End If Frame1(0).Visible = True
  Else If NewMapper Then
    'mi.do "Set Next Document parent "
    & Frame1(1).hWnd & " Style 1"
    'mi.do "Browser * From " & TableName
    'BrowserID = mi.eval("FrontWindow()")
    Frame1(0).Visible = True
    lblMap.Visible = True
  Else
    mi.do "Set Next Document parent "
    & Frame1(1).hWnd & " Style 1"
    mi.do "Browser * From " & TableName
    BrowserID = mi.eval("FrontWindow()")
    lblMap.Visible = False
    End If
  End If
  TabStrip1.Enabled = True
  cboTableList.Enabled = True
  cboTableList.AddItem (TableName)
  cboTableList.Text = TableName

```

다음의 그림 4는 위 프로그래밍에 의해 “테이블 열기” 기능으로 세계지도와 각국의 수도를 나타내는 점지도를 중첩시킨 후 이를 지도 창을 통해 디스플레이한 화면이다.

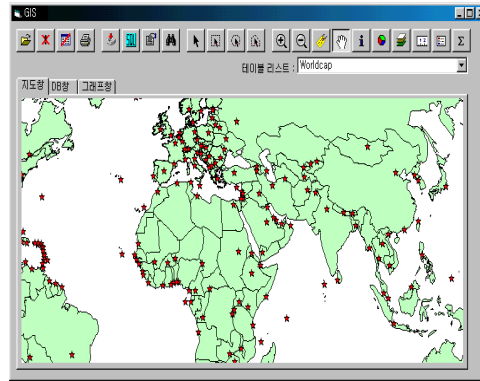


FIGURE 4. A displayed screen of overlaying thematic maps

2) 데이터베이스 창에서 속성 자료의 조회

그림 4의 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. 화면의 우측 상단에 나타나는 콤보 박스인 “테이블리스트”에서 원하는 테이블을 선택한 후 이를 데이터베이스 창을 통하여 디스플레이하기 위한 코드는 다음과 같다.

```

BrowserEnableButton
If cboTableList.Text <> "" And
cboTableList.Text <> OldTableName
Then If BrowserID <> 0 Then
  mi.do "Close Window " &_
  BrowserID
  End If
  mi.do "Set Next Document Parent"
  & Frame1(1).hWnd & " Style 1"
  mi.do "Browse * From " &_
  cboTableList.Text
  BrowserID =
  mi.eval("FrontWindow()")
  OldTableName = cboTableList.Text
  End If
  Frame1(0).Visible = False

```

```
Frame1(1).Visible = True
Frame1(2).Visible = False
```

그림 5는 위 코드를 이용해 각국 수도지도와 연결된 속성 테이블을 선택하여 그 내용을 데이터베이스 창을 통해 디스플레이한 화면이다.

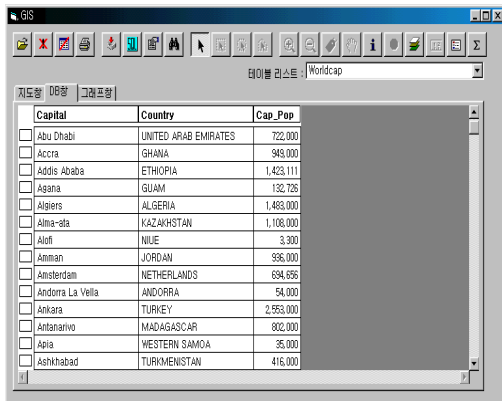


FIGURE 5. A displayed screen of attribute DB

3) 그래프창에서 그래프 분석

지도 창에서 선택된 객체들과 연결된 속성 자료들 중에서 수치형 자료는 그 값의 크기에 따라 그래프 분석이 가능하다. 그림 6에서는 각국 수도 속성 테이블에서 수도 명을 종축으로 하고 인구 수를 횡축으로 하여 그래프 분석을 시행한 결과를 디스플레이한 결과를 보여준다.

그래프 창에서 그래프 분석 기능을 활성화시키기 위한 코드는 다음과 같다.

```
WindowNumber =
mi.eval("NumWindows()")
For i = 1 To WindowNumber
    If mi.eval("WindowInfo(" & Str(i)
    & ",3)") = "4" Then
        GrapherID =
        Val(mi.eval("WindowInfo(" & Str(i)
        & ",13)"))
mi.do "Close Window " & GrapherID
    End If
Next
```

```
NowFrameHandle = Frame1(2).hWnd
NowTabStrip = 3
Frame1(0).Visible = False
Frame1(1).Visible = False
Frame1(2).Visible = True
mi.do "Set Next Document Parent " &
Frame1(2).hWnd & " Style 1"
mi.RunMenuCommand 603
```

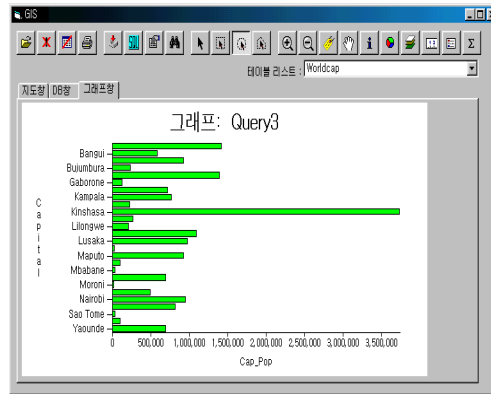


FIGURE 6. A graph presentation of attribute DB

4) 표준 툴바 기능의 에뮬레이션

지금까지 살펴본 지도 창, 데이터베이스 창, 그래프 창의 처리는 탭을 눌렀을 때의 화면 전환과 테이블 리스트에서 선택한 테이블을 처리하기 위해 다소 복잡한 코딩 과정을 거쳤지만 표준 툴바의 기능을 에뮬레이션 하기 위해서는 OLE 자동화 서버와의 간단한 통신만으로 구현이 가능하다. 즉 표준 툴바 기능을 수행할 메소드를 호출하는 방법만 알면 손쉽게 표준 툴바 기능을 구현할 수 있다. 그러나 범례 버튼과 정보 버튼처럼 현재의 윈도우 상에 다른 윈도우를 열고 마우스를 양쪽 창에서 활성화시키기 위한 modeless 윈도우의 경우는 OLE 서버의 modeless 윈도우를 클라이언트가 소유하기 위해서는 다음과 같은 코드를 추가해야만 한다.

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


예를 들어 분포도 분석을 하기 위한 표준 툴바 기능을 수행한 후 modeless 윈도우인 에플리케이션 하기 위해서는 다음과 같은 코드를 사용해야한다.

```
If TgBtn(1).Value = True Then
    mi.do "Set Window Legend Parent " &
    NowFrameHandle
    mi.do "Set Window " & MapperID &
    Front"
    mi.RunMenuCommand 606
Else
    mi.do "Close Window Legend"
End If
```

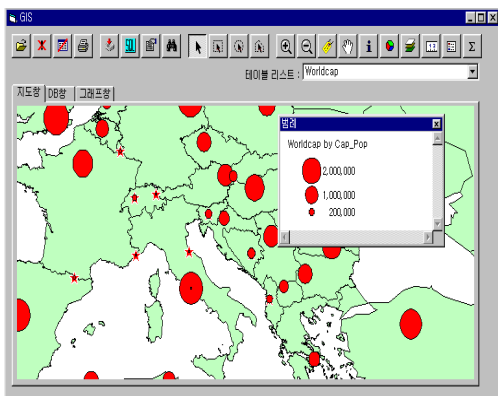


FIGURE 7. A distribution map including the legend window

그림 7은 분포도 분석으로 각국의 수도별 인구를 크기 심볼로 나타낸 후 범례 버튼을 눌러 범례에 대한 modeless 윈도우가 디스플레이 된 화면이다. 이러한 방법으로 전술한 모든 표준 툴바 기능을 에플리케이션 할 수 있다.

평 가

서론에서 서술한 바와 같이 사용자가 단순한 윈도우 프로그래밍만으로 GIS에서 필수적인 첨단 매핑 기능과 각종 분석 기능을 수용할 수 있는 사용자 인터페이스를 개발하는

것은 용이하지 않다. 그러나 OLE 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 2주일 정도에 불과했으며, 사용된 비주얼 베이직 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 OLE서버로부터 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 줄자 기능과 버퍼 기능의 반경 값을 이용한 범위 선택 기능으로 대신 할 수 있기 때문에 사용자의 입장에서는 그다지 불편하지 않을 것으로 보인다.

다만, 사용자 인터페이스의 동작환경을 고려해보면, OLE 자동화 서버가 윈도우즈 환경에서 백그라운드로 실행되기 때문에 32M의 메모리와 100Mhz의 CPU 성능을 갖춘 Pentium PC 이상에서만 동작이 가능했으며, 대체로 MMX 200 Mhz 정도 이상의 Pentium PC이면 좋은 성능을 보임을 확인할 수 있었다.

결 론

GIS 구축을 위한 사용자 인터페이스 개발에 있어서는 GIS의 운용에 필요한 기능을 효과적으로 프로그래밍 할 수 있는 윈도우 프로그래밍 기술과 컴포넌트 소프트웨어 제작 기술의 연동이 중요하다. 본 연구에서는 초보자라도 손쉽게 개발할 수 있는 컴포넌트 소프트웨어 제작 기술로서 OLE 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. **KAGIS**

참고문헌

- 강민석. 1994. 윈도우즈 세계에 부는 OLE 바람. 마이크로 소프트웨어. 177-187.
- 송혜원, 박준기. 1997. Inside COM. 삼각형. 246-249.
- 이이표, 김병세. 1998. Inside ActiveX & OLE 실무 프로그래밍. 삼양출판사.
- 조명희, 부기동, 김광주, 서준석. 1998. GIS를 이용한 대구은행의 점포단위 마케팅 데이터베이스 구축에 관한 연구. 대구은행용역 보고서.
- 주경민, 박성완, 정동길. 1997. Visual Basic Programming Bible Ver 5. 영진출판사.
- MapInfo Co. 1995. MapInfo Professional 4.0. 사용자설명서.
- MapInfo Co. 1995. MapInfo Professional 4.0. 사용자참고서.
- MapInfo Co. 1995. MapBasic professional. 사용자설명서. **KAGIS**