

출입자 통제 시스템을 위한 객체지향 그래픽 시스템 구성편집기의 설계 및 구현

Design and Implementation of an Object-Oriented  
Graphic System Configuration Editor for Access Control Systems

이정호\*, 정선태\*\*, 이명호\*\*\*, 설철규\*\*\*

\*숭실대학교 전자공학과 (Tel: 817-5987; Fax: 821-7968; E-mail: sandkby@syscon.soongsil.ac.kr)

\*\*숭실대학교 전자공학과 (Tel: 820-0638; Fax: 814-1652; E-mail: cst@syscon.soongsil.ac.kr)

\*\*\*LG-Honeywell (Tel: 0417)559-4674; Fax: 0417)559-4708)

**Abstracts** : In this paper, we design and implement an Object-Oriented Graphic System Configuration Editor for Access Control Systems, 'Visual Builder', with such functions as graphic drawing, system configuration, graphic management, processing of user input through graphics. Visual Builder is designed to be well separated from and independent of S/W parts closely coupled with the underlying H/W and thus to improve its reusability and maintainability.

**Keyword** : System Configuration Editor, Access Control Systems, Object-Oriented programming, GUI, UIMS

1. 서 론

출입자 통제 시스템 (Access Control Systems)은 전산실, 연구소, 주요 사무실 등 극히 중요한 장소에 출입자를 선별하여 출입시키는 시스템으로서, 열쇠 역할을 하는 카드를 소지한 허가된 사람만이 각기 인가된 등급에 따라 허가된 출입문을 허가된 시간대에 출입할 수 있게 하고, 출입자의 인적사항, 출입시간, 출입장소 등을 중앙통제실에 설치된 퍼스널 컴퓨터의 운용 프로그램에 의하여 자동기록, 통제, 디스플레이하는 시스템이다[3]. 시스템의 구성은 크게 하부단에 설치되어 직접적으로 출입자를 통제하는 H/W 장치들과 상부에서 H/W 장치의 설정(system configuration) 및 출입통제 상황을 모니터링하는 오퍼레이팅 스테이션으로 구성된다.

마이크로프로세서의 급격한 발달은 H/W 장치들을 급속하게 발전시켜왔으며, 이로 인하여 시스템의 개발 환경은 H/W 개발 전문가들을 중심으로하던 시대에서 벗어나 S/W 개발자들을 위한 프로그램의 효율적인 개발 방법 및 최종 사용자 (end user)를 대상으로 하는 편리한 사용자 인터페이스를 중심으로 연구되어지고 있다[8]. 특히 개발된 S/W의 재사용성 증대를 위한 객체지향(Object-Oriented) 개념 및 사용자 인터페이스를 효율적으로 개발하기 위한 UIMS (User Interface Management System, 사용자 인터페이스 관리 시스템) 개념들이 구체화 되어지고 있다.

그러나, 이제까지 개발된 출입자 통제 시스템들은 출입 통제라는 기능상의 특수성 때문에 특정 기업을 중심으로 자사의 H/W와 그 H/W용으로 작성된 S/W로 이루어진 전용시스템으로 개발·공급되어 왔다. 따라서 H/W 및 S/W들간의 호환성이 거의 없으며, 개발된 S/W의 재사용 또한 거의 불가능하다. 또한 시스템 관리자를 위한 사용자 인터페이스는 H/W 및 S/W의 최적화를 통한 빠른 속도위주로 개발되어 사용이 불편하고, 사용자의 다양한 요구를 수용할 수 있게 개방적이지 못하다[5].

따라서 본 논문에서는 사용자 인터페이스 부분이 출입통제용 H/W 장치에 의존적이지 않다는 사실에 착안하여 출입자 통제 시스템용 S/W가 가져야 할 기능들 중 사용자 인터페이스 부분

들을 하부단 H/W와 밀접한 연관을 맺는 부분들과 잘 분리·독립화시키고, 분리된 사용자 인터페이스 S/W 부분의 재사용성 및 유지보수성을 증대시킨 출입자 통제 시스템용 객체 지향 그래픽 시스템 구성 편집기인 'Visual Builder'를 설계·구현하였다. 개발된 Visual Builder는 출입자 통제 시스템에 필요한 그래픽 도면의 작성 기능, 관계점 설정 기능, 도면 관리 기능, 도면을 통한 사용자 입력 처리 기능 등이 있다.

2. 출입자 통제 시스템

2.1. 출입자 통제 시스템의 구조

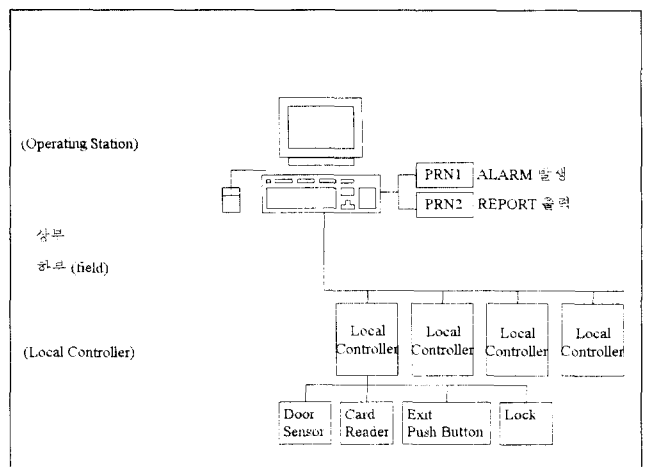


그림 1. 출입자 통제 시스템의 구조

출입자 통제 시스템은 그림 1과 같이 하부단에 직접적인 처리 장치인 Local Controller를 가지고 있으며 상부에서는 H/W 시스템 설정 및 모니터링을 하는 오퍼레이팅 스테이션으로 구성된다.

오퍼레이팅 스테이션은 시스템을 설정하는 작업으로서 빌딩 도면 작성 기능과 작성된 도면과 실제 설치된 H/W와의 관계를 설정하는 관계점 설정 기능이 있어야 하며, 운영하는 작업으로는 현재 출입자에 대한 정보 및 과거 일정기간의 출입상황 등을 관리하는 Database 기능 및 H/W와 통신을 수행하는 기능 등이 필요하다. 현장에서 실제적인 출입을 통제하는 H/W 장치인 Local Controller는 Card Reader 및 Lock 등을 이용하여 출입자의 출입 허용 및 출입문 관리의 기능을 수행 한다.

## 2.2. 출입자 통제 시스템 S/W의 분석

출입자 통제 시스템 S/W는 첫째, 시스템을 설정하는 기능으로 빌딩 도면을 작성하고, 도면상에 그려진 각각의 Door 및 Lock을 실제 설치된 H/W와 매핑(mapping, 관계점 설정)을 할 수 있어야 한다. 둘째로 시스템 운용시에는 시스템 설정시에 그려진 도면을 관리하며, 현재 출입할 수 있는 출입자의 출입허용, 출입자 추가, 출입자 삭제에 관한 출입자 정보와 과거 일정 기간의 출입 정보를 관리할 수 있어야 한다. 또 H/W와 통신을 통하여 출입자를 통제 하여야 한다.

초기의 문자기반(text base) 사용자 인터페이스는 본 논문에서 제외하기로 하고, 그래픽 사용자 인터페이스 기반의 출입자 통제 시스템 S/W를 분석해 보면 다음과 같다.

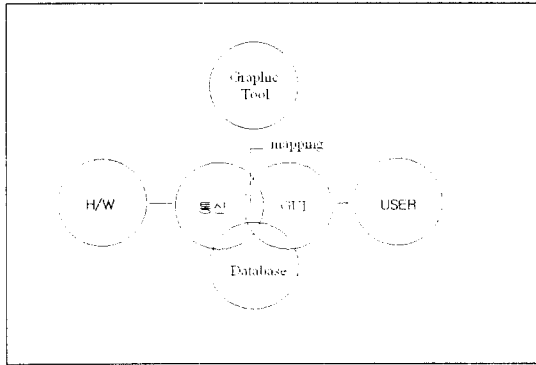


그림 2. 기존 출입자 통제 시스템의 S/W 구조

기존의 출입자 통제 시스템 S/W는 그림 2에서와 같이 통신, 데이터 베이스 및 사용자 인터페이스의 여러 가지 기능이 복합적으로 구성된 메인프로그램과 도면을 작성하는 상용의 그래픽 툴의 두 부분으로 나눌 수 있다.

메인 프로그램부분은 통신부분을 이용하여 H/W로부터 출입 상황을 받아들여 데이터베이스에 기록하고 사용자 인터페이스 부분을 통해서 결과를 사용자에게 보여준다. 또한 사용자의 입력을 받아 현재까지 데이터베이스에 기록된 사항들을 처리하여 사용자에게 결과를 보여준다. 즉 위의 그림 2와 같이 통신, 데이터 베이스 및 사용자 인터페이스의 여러 가지 기능이 복합적으로 구성된 메인프로그램을 중심으로 모든 기능이 모여 있으며, 각각의 기능들이 저수준의 라이브러리로 구성되어 있어 각각의 부분을 분리하기가 쉽기 않다. 이렇게 복합적으로 구성되어 빠른 처리 속도의 장점을 얻을수 있으나, S/W 부분들 중 H/W와 관련 깊은 부분들이 많아 새로운 H/W 적용시 S/W 전체를 수정해야 하며, 새로운 기능의 추가가 쉽지 않으며, 재사용성이 떨어진다.

또한, 도면의 작성을 상용의 그래픽 툴을 이용함으로써 출입자 통제 S/W의 사용자의 인터페이스인 도면의 변경 및 관계점 설정 등의 시스템 관리사용 인터페이스 부분의 변경이 용이하지 않다.

본 논문에서는 기존 출입자 통제 S/W의 H/W 종속성 배제, S/W의 재사용성 증대 및 사용자의 인터페이스의 변경 요구를 쉽게 수용하기 위하여 기존 출입자 통제 S/W 부분들 중 상용의

그래픽 툴이 담당하던 그래픽 도면의 작성 기능과 메인 프로그램 중 관계점 설정 기능, 도면 관리 기능, 도면을 통한 사용자 입력 처리 기능 등을 H/W와 밀접한 관련을 맺는 통신 및 Database 기능들과 분리하고, 객체지향 프로그래밍 기법을 이용하여 이 부분의 재사용성을 높인 출입자 통제 시스템용 그래픽 시스템 구성 편집기인 'Visual Builder'를 구현한다.

## 3. Visual Builder의 설계 및 구현

### 3.1. Visual Builder와 H/W 관련 기능중심부의 분리

Visual Builder를 설계하기 위하여 우선 출입자 통제 S/W에 필요한 각 부분의 기능들을 명세하고, 이들 중 H/W 의존적인 기능중심 부분들과 H/W 교체시에도 재 사용할수 있는 Visual Builder 부분을 분리한다.

출입자 통제시스템의 대표적인 기능을 명세하면 다음과 같다.

- ▶ 도면 작성 및 관리 기능
- ▶ 사용자의 입력 관리 기능
- ▶ 관계점 설정 기능
- ▶ 데이터 베이스 관련 기능
- ▶ 통신 기능

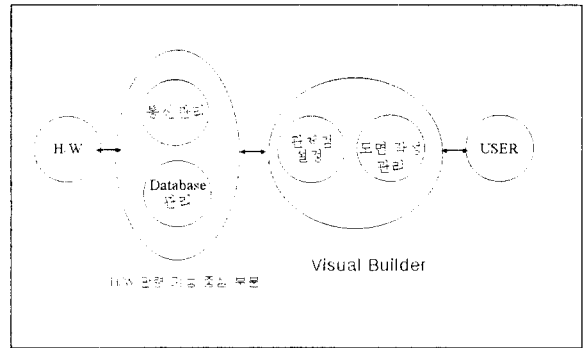


그림 3. Visual Builder와 기능중심부로 분리된 S/W 구조

다섯가지 기능의 명세로부터 H/W 교체시에도 다시 사용할 수 있는 기능인 도면의 작성기능, 사용자의 입력관리 기능 및 관계점 설정 기능을 중심으로 시스템 구성 편집기를 구성하고, H/W 의존적인 통신기능 및 Database 관련 기능을 기능중심부분으로 구성한다. 즉 Visual Builder는 크게 도면 작성 부분과 관계점 설정 부분으로 구성되며, 관계점 설정 기능을 이용하여 분리된 기능중심부와 관계를 맺는다.

이렇게 S/W를 두 부분으로 분리함으로써 분리된 두 부분은 서로에게 최소한의 영향을 주면서 각각 개발될수 있으며, Visual Builder 중 관계 설정 부분에 해당하는 관계점 설정에 대한 정보만을 공유하면 Visual Builder와는 상관없이 개별적으로 개발된 H/W 및 H/W 관련 기능 중심부분의 S/W와도 호환성을 유지할수 있다.

다음절에서는 Visual Builder의 기능 중 도면의 작성 및 관리를 담당하는 부분과 그려진 도면을 실제 설치된 H/W와 관계를 맺는 관계점 설정 부분에 대하여 각각의 설계 및 구현 내용을 살펴본다.

### 3.2. Visual Builder 중 도면작성 부분의 설계 및 구현

Visual Builder는 출입자 통제 시스템의 사용자 인터페이스에 사용되는 건물의 도면 및 건물에 설치된 H/W 장치들을 그려주는 기능이 필요하다. 이런 기능을 구현하기 위하여 Visual Builder의 기능을 명세하고, 이들 명세로부터 객체들을 추출하고

객체의 기능들을 찾아 클래스를 구성한다. 이렇게 클래스로 구현함으로써 객체지향의 장점인 재사용성의 향상, 쉬운 유지보수 등의 여러 장점을 얻을 수 있다.

우선 Visual Builder의 도면 작성 부분에 필요한 기능을 명세하면 다음과 같다.

- 도면에 사용되는 선, 원, 사각형 등의 그래픽 객체를 그리는 기능
- 그래픽 객체들의 속성 설정 기능
- 그래픽 객체들을 관리하는 기능
- 상용 그래픽 툴에서 그린 그래픽 파일 이용 기능
- 자주 사용되는 그래픽을 라이브러리로 사용할수 있는 기능
- 그래픽 객체들로 구성된 도면을 관리하는 기능

기능의 명세로부터 객체들을 추출하면 선, 원 등의 그래픽 객체들을 추출할 수 있으며, 그 객체들의 기능들은 주로 마우스를 이용한 사용자의 입력을 받아들이는 기능들이다. 본 논문에서는 그래픽 객체들과 사용자의 입력처리 기능들을 분리하여 각각의 클래스를 구현함으로써, 사용자의 입력장치의 변화 및 새로운 입력 기능의 추가 및 삭제를 쉽게 할수 있게 설계한다.

우선 이들중 그래픽 객체에 해당하는 선, 원, 타원, 사각형, 둥근 사각형, 문자열, 그래픽 파일, 다각형 등은 단일 객체들로 구성할수도 있지만 모두 두 개의 좌표점 사이에 표현되며, 내부 색깔 등을 가진다는 공통점을 이용하여 기본이 되는 클래스(CDrawObj)를 설계하고 각각의 특징들을 기본 클래스로부터 계승하여 구현한다. 이렇게 함으로써 추후 새로운 사용자 인터페이스에 필요한 그래픽 구성요소들을 추가할 때 기본 클래스로부터

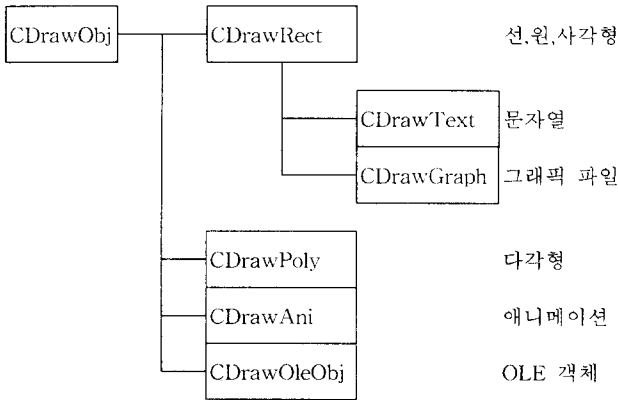


표 1. 그래픽 객체들의 계승 구조

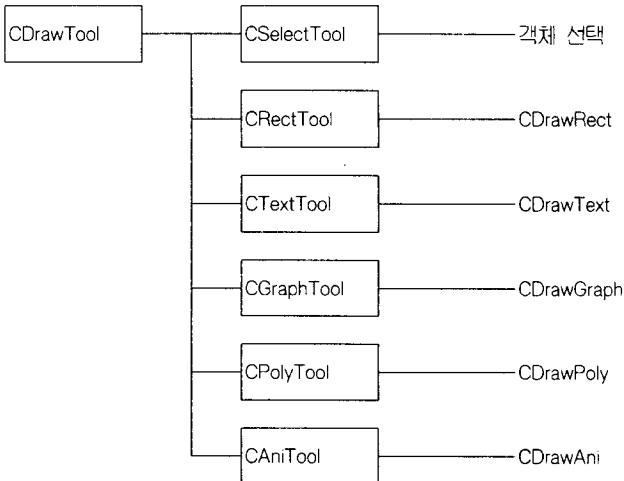


표 2. Drawing Tool의 계승도

계승하여 쉽게 기존 구조에 삽입할 수 있다. 계층적으로 설계된 그래픽 객체 관련 클래스들의 계승관계는 표 1과 같다. 이들 그래픽 객체들을 그려주는 기능들은 마우스 동작을 중심으로 Click, Double Click, Drag 등 한정된 몇가지 동작들로 이루어지므로 기본적인 동작들을 추출하여 기본 클래스(CDrawTool)로 구현하고, 각각의 기능들을 기본클래스로부터 계승 구현한다. 이들 기능들을 그래픽 객체 하나씩과 연결된다. 계층적으로 설계된 그래픽 객체 관련 클래스들의 계승관계는 표 2와 같다.

이렇게 설계된 그래픽 툴을 이용하여 그려진 그래픽 객체는 객체의 속성 설정(Property Setting Dialogue)을 통하여 색깔 등의 속성을 설정한다. 각각의 그래픽 객체들은 도면이라는 자료구조에 저장되고 관리된다. 이렇게 관리됨으로서 각각의 그래픽 객체들은 한 장의 도면으로 인식된다. 또한 자주 사용되는 출입문, 계단 등의 그림은 라이브러리로 등록하여 쉽게 도면에 삽입할수 있으며(Library Manager), 상용의 그래픽 툴에서 그려진 그림들도 도면에 삽입할 수 있다(Graphic File Load & Display). 이 과정에 기능의 명세에는 없지만 많은 메모리를 차지하는 그래픽 파일들을 효율적으로 관리하기 위하여 그래픽 파일 버퍼(Graphic File Buffer)를 추가로 설계하였다.

기능의 명세로부터 각각 설계된 클래스들간의 관계를 도면 작성 과정을 통하여 살펴보면 그림 4와 같다.

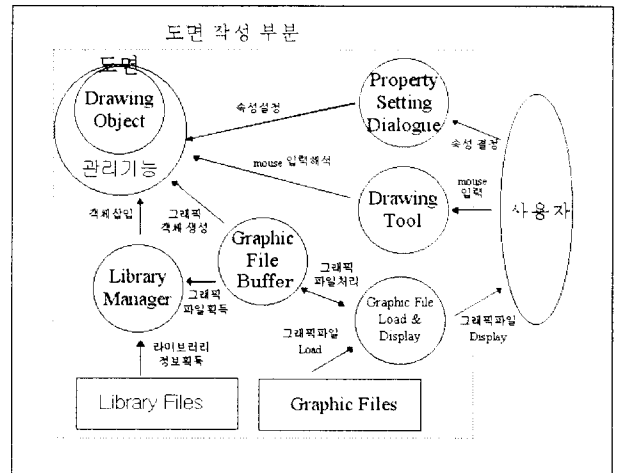


그림 4. 도면 작성에 관련된 객체들의 관계도

실제로 구현된 Visual Builder로 그린 도면의 예는 아래 그림 5와 같다.

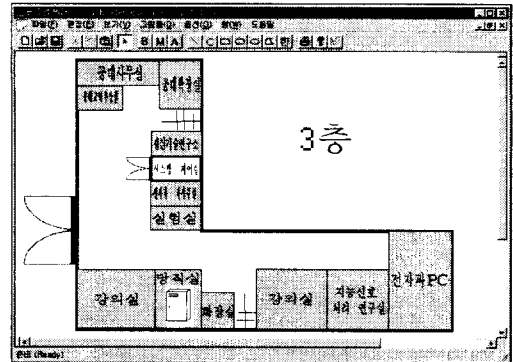


그림 5. 구현된 Visual Builder로 그린 도면의 예

### 3.3. Visual Builder의 관계 설정 부분의 설계 및 구현

Visual Builder 부분과 H/W 관련 부분들의 분리로 얻을수 있

는 가장 큰 장점은 많은 시간과 노력을 요하는 그래픽 사용자 인터페이스의 독립적이고 신속한 개발이다. 이것은 Visual Builder 부분의 독립적인 설계에서도 많은 영향을 받지만 분리된 두 부분을 연계시키는 관계 설정 부분이 분리된 양쪽을 어떻게 연계시키느냐 하는 것에도 많은 영향을 받는다. 따라서 관계 설정 부분의 설계에는 분리된 두 부분이 최소한의 관계만을 맺도록 하는 것을 목표로 설계·구현 한다.

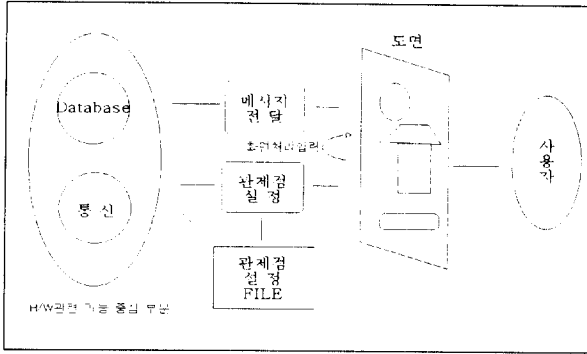


그림 6. 관계점 설정 및 메시지 전달

관계 설정 부분은 초기 H/W의 상태를 설정하는 관계점 설정 기능과 프로그램 실행시 설정된 내용에 따라 사용자의 입력을 기능중심으로 짜여진 부분에 메시지를 전달하는 기능으로 구성된다. 우선 사용자는 작성된 도면과 실제 설치된 H/W의 관계를 설정하고, 설정된 정보를 기능중심 부분에 공개된 형식으로 저장한다. 사용자 입력 전달 부분은 관계점 설정 기능을 통하여 공개된 설정 정보를 바탕으로 S/W 운영시에는 사용자의 입력이 도면의 어떤 그래픽 객체를 선택했는지의 정보만을 전달해 주면 된다. 또한 정보 전달시 기능 중심부의 각각의 기능과 함수에 신경쓰지 않고 사용자의 입력에 따라 적당한 메시지를 발생하여 전달하고, 기능중심부는 공개된 메시지에 따라 동작하는 메시지 호출 방법을 사용하면 분리된 두 부분의 독립성을 높이고, 호출되는 메시지 구조만을 공유하면 어떤 기능적인 부분 및 H/W와도 호환성을 가질수 있다.

관계점 설정에 필요한 기능을 명세하면 다음과 같다.

- 영역의 설정 (도면에 표시되는 영역들을 설정)
  - 도어의 설정 (도어에 설치된 센서 및 Lock 설정)
  - LPM의 설정 (Local Controller의 정보 설정)
  - 터미널 설정 (Card Reader의 설정)
  - 그림 단축키 (다른 도면으로 이동하기 위한 설정)
- 이들 기능의 명세로부터 각각의 기능에 해당하는 설정 기능을

메시지 발생 부분	가상함수 제공 Send_mess(CString &Name, int Type)
메시지 전달 부분	Window구조로 메시지 발생 PostMessage(m_HO_HWND, WM_USER+7, Type 0) 메시지 받는 퍼블릭 함수 설정 ON_MESSAGE(WM_USER+7, Management)
메시지 처리 부분	메시지의 내용에 따라 처리 Management(int Type, LONG) { switch(Type) { case 1: // LPM case 2: // Door ... } }

표 3. 메시지 전달 구조

관계점 설정 다이얼로그 박스로 구성했다. 이렇게 설정된 관계점 설정 내용은 기능중심부분에서 알 수 있는 File의 형식(format)으로 저장하여 개별적으로 개발된 기능중심부에도 File의 형식만을 공유하면 호환이 가능하도록 한다.

또한 사용자의 입력을 기능 중심부분으로 전달하는 메시지 전달 구조는, 우선 메시지 발생 부분에는 Send\_mess()에 해당하는 가상의 함수를 제공하고, 메시지 전달 부분에는 windows의 메시지 전달구조를 이용하며 Send\_mess()의 함수를 받아서 처리할 함수명을 정한다. 메시지를 받아서 처리할 함수에는 메시지의 내용을 파악, 각각의 경우에 맞는 처리를 한다. 이렇게 구성함으로써 기능중심 부분의 S/W 교체시에도 Visual Builder의 구조 및 S/W 변경없이 기능중심부의 메시지를 받을 함수만을 공개된 형식으로 수정함으로써 쉽게 새로운 S/W에 적용이 가능하다.

#### 4. 결 론

본 논문에서는 시스템의 고성능을 중심으로 S/W의 구조를 설계하여, 사용이 불편하고, 새로운 기능 추가나 새로운 H/W에 적용할때, H/W와 무관한 그래픽 처리 부분까지 새로이 작성해야 하는 기존 출입자 통제 시스템용 S/W의 구조적인 문제점을 개선하여 출입자 통제 시스템을 위한 객체지향 그래픽 시스템 구성 편집기인 Visual Builder를 설계하고 구현하였다.

Visual Builder는 출입자 통제 시스템용 H/W와는 상관없는 도면의 작성 및 관리기능과 관계점 설정 기능을 H/W에 의존적인 통신부분과 분리 구현함으로써 H/W 교체시에도 H/W와 관계 깊은 통신 부분만을 수정 또는 교체함으로써 Visual Builder 부분은 전혀 변경없이 기존 구조를 유지한채 쉽게 새로운 H/W에 적용이 가능하다. 이 경우 사용자 인터페이스 부분은 전혀 수정이 필요 없으며 필요시 관계 설정 부분만을 변경함으로써 사용자 인터페이스의 일관성을 유지할 수 있다. 또한 출입자 통제 시스템의 각 구성요소인 Database, 통신, 및 시스템 구성 편집 기능을 개념적으로 분리 구현하므로 각각의 부분을 따로 개발할 수 있다.

또한, Visual Builder 작성에 필요한 각 부분들을 클래스화하고, 공통된 특성을 계층적 구조로 구현하였으므로 향후 새로운 기능이 필요할 때에도 쉽게 추가할 수 있다.

#### 참 고 문 헌

- [1] Deborah Hix, "Generation of User-Interface Management System", *IEEE Software* September 1990
- [2] Joelle Coutaz, Len Bass, "Requirements for UIMSs", *Esprit/Eurographics Workshop on User Interface Management Systems and Environments*, Losbon, 4-6 June 1990
- [3] LG-하니웰 기술사보 통합본 (빌딩부분 제1집) 출입통제시스템(Access Control System)에 관한 소개 P. 87-88
- [4] LG-하니웰 기술사보 통합본, 인텔리전트 출입자 통제 시스템, P. 141-143
- [5] LG-하니웰 기술사보 통합본, 해외기술동향 P. 8
- [6] 서봉원, UIMS를 위한 객체지향 클래스, 서울대학교 전자계산학과 석사학위 논문
- [7] 이정호, 출입자 통제 시스템을 위한 객체지향 그래픽 UIMS 설계 및 구현, 숭실대학교 전자공학과 석사학위논문 96.8
- [8] 인민아, 소프트웨어 재사용을 위한 GUI 객체 라이브러리 구축에 관한 연구, 중앙대학교 전자계산학과 석사학위 논문 92.6