

# 애니메이션 객체를 도입한 HMI의 개발

## Development of HMI Which Took in a Animation Object

\*김민석, \*\*김한승, \*\*\*홍정기  
(Min-Seok Kim, Han-Sung Kim, Jung-Gi Hong)

**Abstract** - Our society experiences a big change recently for ten years. The public of multimedia by development of a PC and Internet diffusion is it. this situation was applied in the field of industry. HMI of an industrial field experienced a big change these days. it comes out in new general ideas one after another, for example Interface of an Internet base, the animated graphic object, etc.. A study of a book is contents about animation function development of Human Interface Station(HIS) which is HMI of existing DCS system.

**Key Words** : DCS, HMI, ANIMATION

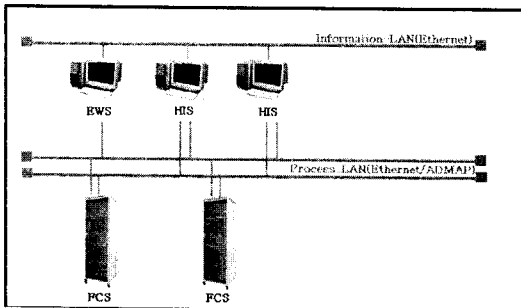
### 1. 서론

#### 1.1 개발 배경

최근 10년간 우리사회는 큰 변화를 겪고 있다. PC의 발전과 인터넷 확산으로 인한 멀티미디어의 대중화가 그것이다. 산업분야에서도 이 물결은 유효했다. 근래 산업분야의 HMI는 큰 변화를 겪었다. 인터넷 기반의 인터페이스를 보여주고 그래픽 객체에 애니메이션을 도입하는 등 새로운 개념들을 속속 선보이고 있다. 본 논문은 기존의 효성 DCS 시스템의 HMI인 Human Interface Station(HIS)의 애니메이션 기능 개발에 관한 내용이다.

### 2. 본론

#### 2.1 시스템 구성 및 기능



저자 소개

- \*인하대학 컴퓨터공학과 공학사
- \*\*인천대학 컴퓨터공학과 공학사
- \*\*\*명지대학 전기공학과 석사

#### <그림1> HIPAC NT 시스템 구성도

당사에서 개발한 분산제어 시스템인 HIPAC NT의 시스템 구성은 위 구성도와 같다.

#### 2.1.1 Hardware 구성

<그림 1>에서 보는 것과 같이 크게 4부분으로 나눌 수 있다. HMI와 태그그룹과 사용자그래픽 엔지니어링을 담당하는 HIS (Human Interface Station), 컨트롤러와 태그로직 엔지니어링을 담당하는 EWS (Engineering WorkStation), DCS 컨트롤러인 FCS (Field Control Station) 로 구성된다.

#### 2.1.2 Software 구성

본 제품의 S/W 구성은 <표 1> 과 같다

스테이션 명	S/W구성	기능
FCS	-	-
EWS	Launcher	프로젝트설정 사용자관리 FCS다운로드 등
	System Builder	시스템, I/O설정 등
	Control Logic Builder	제어로직작성 제어로직모니터링 등
	TagBrowser	태그조회
HIS	Xrun	현장 감시 제어
	Xbuilder	태그표시 태그그룹설정 레포트등록 등
	Xdraw	사용자그래픽제작 태그속성지정 등

<표 1> HIPAC NT S/W 구성

## 2.2 애니메이션 객체 추가

### 2.2.1 애니메이션 객체

기존 HIS의 그래픽 에디터인 Xdraw 에 GIF 형식의 애니메이션 객체타입을 추가했다.

GIF는 JPEG과 함께 월드와이드웹에서 지원되는 두 가지 그래픽 파일 형식 중의 하나이다. 웹이나 기타 인터넷의 다른 서비스(예를 들면 게시판 서비스 등)에서 GIF는 이미지 형식에 있어, 사실상의 표준이 되어가고 있다.

GIF는 2차원 라스터 데이터 형식으로서, 바이너리로 표현되며, 압축 기술은 LZW를 이용한다. GIF에는 87a와 89a의 두 가지 버전이 있는데, 1989년 7월에 발표된 89a는 하나의 파일내에 짧은 순서를 갖는 일련의 이미지들을 포함시킴으로써 animated GIF를 만들 수 있는 기능과, 인터레이스(interlaced GIF) 표현 기법을 지원한다.

향후 GIF 형식 말고도 플래쉬나 avi형식의 객체타입 추가도 고려하고 있다.

### 2.2.2 GIF 객체 타입 추가 (Xdraw의 경우)

#### 2.2.2.1 Gif89a 지원

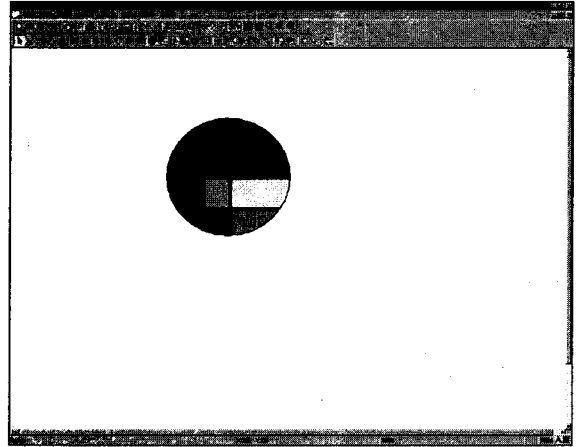
Animated GIF를 지원하게 하기위해 GIF89a 형식의 이미지를 다룰 수 있는 class를 Xdraw과 Xrun에 추가하였다.

#### 2.2.2.2 GIF 타입 추가

아래 코드를 보면 그래픽객체의 타입을 enum 형으로 나열해 놓은 것을 볼 수 있다. 여기에 gif라는 타입을 추가했다.

```
enum DrawShape
{
    selection,
    line,
    hvline,
    rectangle,
    roundRectangle,
    ellipse,
    arc,
    poly,
    polyline,,
    .....
    sell,
    meter,
    htrend,
    extarc,
    extcir,
    cirarc,
    bitmap,
    gif
};
```

### 2.2.2.3 GIF 그래픽 객체 생성

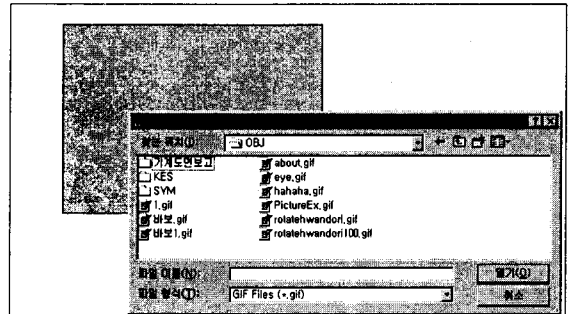


<그림2> Xdraw 에서 GIF 객체를 불러온 모습

그리기>그림>그림(GIF 형식) 메뉴를 클릭하면 메뉴의 메시지 핸들러 함수가 불러진다.

```
void CDrawView::OnDrawGif() {
    CDrawTool::SetCurrentTool(gif);
}
```

현재 Tool이 gif타입으로 결정되고 이제 마우스 드래그로 화면에 그려질 영역을 설정한다. 그리고 영역 설정 후에는 아래처럼 보여줄 파일을 선택하게 된다.



<그림3> Xdraw 에서 GIF 객체 선택 화면

view에 표시 : CDrawView class 의 'OnDraw' 에서 CDrawDoc class의 'Draw' 함수를 호출하게 된다. 여기서 함수의 인자로 화면의 DC를 넘겨준다.

```
void CDrawView::OnDraw(CDC* pDC)
{
    .....
    pDoc->Draw(pDrawDC, this);
    .....
}
```

view에 표시 : CDrawDoc class의 Draw 함수는 현재

페이지의 모든 객체를 순차적으로 검색해서 그 객체타입에 맞는 그리기 함수를 호출한다.

```
void CDrawDoc::Draw(CDC* pDC, CDrawView* pView)
{
    POSITION pos = m_objects.GetHeadPosition();
    while (pos != NULL)
    {
        CDrawObj* pObj = (CDrawObj*)m_objects.GetNext(pos);
        if (pObj->Intersects(gClipRect))
            pObj->Draw(pDC);
    }
}
```

view에 표시 : Gif 파일을 xdraw 의 view 영역에 표시한다.

```
void CDrawRect::Draw(CDC* pDC)
{
    .....
    // 해당 도형의 드로잉 함수를 실행.
    switch (m_nShape)
    {
        .....
        case gif:
            .....
            AfxBeginThread(PlayGif, (LPVOID)gif);
            .....
    }
}
```

## 2.2.3 애니메이션 동작 (Xrun)

### 2.2.3.1 동작

Xrun 의 Custom Graphic 화면을 선택하면 그래픽 파일을 호출하고, GIF 객체가 존재할 경우 각 GIF 객체의 Delay Time을 적용해서 화면에 보여주게 된다

기본적으로 타입추가나 GIF 지원 클래스 등은 Xdraw 에서의 경우와 같다.

### 2.2.3.2 구현

Custom Graphic 화면이 선택되고 GIF객체가 존재한다고 가정. CDrawView class 의 OnDraw 함수에 아래와 같은 코드가 추가된다.

```
void CDrawView::OnDraw(CDC* pDC) {
    .....
    if (gif_list.size() != 0) {
        .....
        while (pos != NULL) {
            .....
        }
        std::list<GIFLIST>::iterator where = gif_list.begin();
        for (int i = 0 ; i < gif_list.size(); i++) {
            int delay = ((GIFLIST)*where).nDelayTime;
            SetTimer(((GIFLIST)*where).keyNum, delay, NULL);
            where++;
        }
        .....
    }
}
```

현재 화면의 GIF객체의 정보를 별도의 리스트로 가지고

있게 된다. 그 정보를 가지고 Animated GIF를 구동하게 된다.

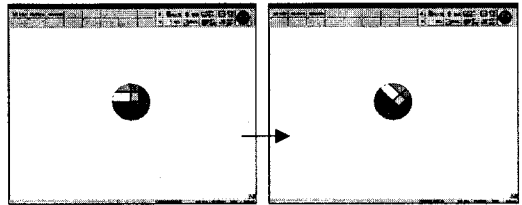
그리고 각 GIF 객체의 frame 당 지연시간 (Delay Time) 정보를 가지고 타이머를 적용한다.

정해진 Delay Time이 지난 후에는 다음 frame 의 새로운 Delay Time 을 적용해서 다시 타이머를 생성한다. 이 과정이 현재 Custom Graphic 화면이 다른 화면으로 바뀔 때 까지 계속 반복된다.

```
void CDrawView::OnTimer(UINT nIDEvent)
{
    .....
    ObjAnimationGIF(&(*where), GetDC() );
    .....
}
```

그리고 인자로 넘어간 GIF객체의 리스트와 View 클래스 DC를 가지고 GIF 그래픽을 한 frame씩 보여주고 Delay Time을 갱신한다. (ObjAnimationGIF 함수)

CGif89a class 의 PlaySeq 함수에서 실제로 GIF의 각 Frame을 Custom Graphic 화면에 표시해 준다. 아래는 그 화면이다.



## 3. 결론

본 논문에서는 HMI 에서 GIF 포맷을 이용한 애니메이션 구현 기술을 적용한 제품에 대해서 소개하고 있다.

애니메이션 객체가 적용됨으로써 유저는 감시 대상의 상태를 좀더 쉽게 알 수 있게 된다. 이러한 개념은 이미 산업현장에서 적용이 되고 있다.

최근 분산제어시스템과 IT기술의 접목이 급격하게 진행되고 있다. 이러한 대내환경의 변화에 대처하기 위해서는 IT 기술이 접목된 솔루션의 개발이 필요하다. 특히 멀티미디어 기술과의 접목은 주목할 만하다.

당사는 현재 이러한 요구에 대해 빠르게 적용하고 있으며 앞으로 환경 변화에 적극적으로 대응하기 위해 노력 하고 있다.

## 참 고 문 헌

- [1] (주)효성, "HIPAC Series NT Ver2.0 기술매뉴얼", 2004년
- [3] <http://www.bril.pe.kr/>
- [4] <http://www.imadang.co.kr/>