

Drag&Drop 메커니즘을 이용한 웹 이미지 수집기의 구현

An Implementation of Web Image Collector using Drag&Drop Mechanism

이 선 응*, 문 일 영*

Seon-Ung Lee*, Il-Young Moon*

요 약

Drag&Drop 메커니즘의 전신은 마이크로소프트 윈도우즈의 클립보드이다. 클립보드를 사용하는 복사, 붙여넣기 기능이 마우스 이벤트를 통해 처리되는 것이 바로 Drag&Drop인 것이다. 요즘 PC, 노트북, 모바일 폰을 막론하고 터치 인터페이스가 각광을 받고 있다. 마우스와 터치 인터페이스는 눈에 보이는 상호작용을 통해 더욱 쉽고 직관적으로 작업할 수 있는 환경을 만들어 준다. 본 논문에서는 Drag&Drop을 활용할 수 있는 웹 이미지 수집기를 구현하였다. 이를 통해 Drag&Drop의 적용 방법 및 활용 방안을 제시한다.

Key Words : Drag&Drop, Clipboard, Intuitive Interface

ABSTRACT

Drag&Drop mechanism was formerly the clipboard of Microsoft Windows. Drag&Drop means that copy and paste functions using the clipboard are processed by a mouse event. The touch interface come into the spotlight not to speak of PCs, laptops and mobile phones. Mouse and touch interfaces make an environment to work easier and intuitive through visible interactions. In this paper, we implemented a web image collector to utilize Drag&Drop. And we proposed the how to apply and a utilizable plan from it.

I. 서론

요즘 컴퓨터에 설치된 운영체제들은 대부분이 GUI(Graphic User Interface)를 제공하고 있다. 그래픽 환경에서 주된 작업은 마우스를 통하여 입력되는 신호를 통하여 시작되고 상황에 맞게 처리된다. 이렇게 마우스로 처리되는 작업 중에 유용한 것은 바로 Drag&Drop이다. Drag&Drop은 사용자들의 작업을 쉽고 직관적으로 진행할 수 있게 한다.

현재 우리 주변 환경에는 이미 많은 수의 기기 및 서비스가 배치되어 있다. 또한 미래형 컴퓨팅 환경에서는 환경에 편재하는 서비스의 종류와 수가 다양해지기 때문에, 이들 서비스와 사용자가 어떻게 상호작용을 할 것인지에 대한 연구가 필요하다. 기존 연구에서는 사용자에게 상황과 용도에 따른 개별적으로 분리된 다른 인터페이스를 제공하는 방식이 사용되었다. 이로 인해 사용자는 선택의 여지가 없이 다 수의 인터페이스를 사용해야 하는 불편함을 감수

* 한국기술교육대학교 인터넷미디어공학부(kogangdon@kut.ac.kr, iymoon@kut.ac.kr)

제1저자 (First Author) : 이선응

교신저자 : 문일영

접수일자 : 2009년 7월 1일

수정일자 : 2009년 8월 5일

해야 하고 그 수가 늘어남에 따라 각기 다른 인터페이스 학습에 필요한 시간 또한 증가하게 된다. 그래서 서비스를 사용하는 사용자 입장에서 좀 더 편리하고 직관적으로 사용할 수 있는 방안이 요구되고 있다⁽¹⁾⁽²⁾.

최근 직관적인 사용자 인터페이스가 다양한 방면에서 중요하게 여겨지고 도입하려는 시도가 생겼다. 이러한 흐름은 특히 애플의 아이폰(iPhone)의 등장하면서 스마트폰 시장에서 크게 형성이 되었다. 아이폰의 이슈로 인하여 여러 기업에서 이를 따라잡기 위해 화려한 사용자 인터페이스를 가지고 나왔다. 이들의 인터페이스는 화려하기는 했지만 아이폰의 심플하고 직관적인 사용자 인터페이스에 비해 사용자들의 마음을 잡지 못했다는 평이다⁽²⁾. 이러한 흐름이 점차 확대되어 아이폰에 탑재되었던 멀티 터치 인터페이스가 일반 컴퓨터와 노트북에도 탑재되고 있는 추세이다⁽⁴⁾. 멀티 터치 인터페이스든지 일반 터치 인터페이스든지 사실 두 가지 모두 마우스보다 더 직관적인 인터페이스를 지향한다. GUI에서 직관적인 작업을 제공하는 Drag&Drop의 효율성은 더욱 커질 것으로 보고 있다.

따라서 본 논문에서는 위와 같은 중요성을 바탕으로 웹 이미지 수집기라는 프로그램을 구현하는 방법을 제시하여 Drag&Drop의 메커니즘과 활용 방안을 알아본다. 제 II장에서는 Drag&Drop 메커니즘에 대하여 알아본다. 제 III장에서는 다양한 웹 브라우저의 이미지 포맷을 알아보고 적합한 데이터 표현 방법을 기술한다. 제 IV장에서는 구현 알고리즘에 대하여 설명한다. 마지막으로 제 V장에서는 Drag&Drop의 활용 방안에 대해서 정리하면서 결론을 맺는다.

II. Drag&Drop 메커니즘

Drag&Drop 메커니즘은 크게 두 가지로 나눌 수 있다. 하나는 파일 관리자의 Drag&Drop 메커니즘이고, 다른 하나는 OLE Drag&Drop 메커니즘이다. 파일 관리자의 Drag&Drop 메커니즘은 파일명을 수정하거나 파일명을 응용 프로그램에 전달하는 역할만 수행한다. OLE Drag&Drop 메커니즘은 더 일반적이다. 이는 클립보드에 올라갈 수 있는 데이터는 무엇이든 Drag&Drop 작업이 가능하기 때문이다⁽⁵⁾. 따라서 앞으로 나오는 Drag&Drop은 파일 관리자를 제외한 OLE Drag&Drop임을 명시한다.

앞에서 언급하였듯이 Drag&Drop은 클립보드에

올라갈 수 있는 데이터라면 모두 가능하다. 하지만 몇 가지 조건들이 충족되어야만 가능하다. 첫째, 복사할 데이터가 들어있는 소스 윈도우와 데이터를 붙여 넣을 목적지 윈도우가 현재 창이 열린 상태로 실행되고 있어야 한다. 둘째, 소스 윈도우에서는 Drag가 시작될 때 데이터를 클립보드에 복사하는 기능과 목적지 윈도우에서는 Drop될 때 클립보드에서 데이터를 가져오는 기능이 구현되어 있어야 한다. 셋째, 소스 윈도우와 목적지 윈도우 간에는 정해진 포맷이 있어야 클립보드를 통하여 전달할 수 있다. 먼저 Drag&Drop의 전신인 클립보드에 대해서 자세히 설명한다.

1. 클립보드

프로그램 간에 데이터를 공유하는 여러 가지 방법 중 가장 기초적이고 쉬운 방법이 바로 클립보드를 사용하는 것이다. 클립보드는 프로그램 간에 또는 프로그램 내부적으로 교환할 데이터를 잠시 저장해 두는 곳이다. 소스 윈도우가 데이터를 넣어 놓으면 이 데이터를 사용하는 목적지 윈도우는 필요할 때 데이터를 꺼내 간다. 클립보드는 시스템이 관리하고 시스템 전체에 유일한 자원이므로 개별 프로그램들은 꼭 필요할 때에 한해 정해진 절차를 따라 클립보드를 사용해야 한다⁽⁶⁾.

목적지 윈도우에서 클립보드에 저장된 데이터를 꺼내 가기 위해서는 저장된 데이터의 형태를 알아야 한다. 클립보드에 저장된 데이터의 형태를 식별하는 정수 값을 클립보드 포맷이라 한다. 클립보드 포맷은 윈도우가 미리 지정해 놓은 표준 클립보드 포맷과 특정 프로그램에서 등록하는 사용자정의 포맷이 있다. 표 1에서 윈도우즈가 지원하는 표준 클립보드 포맷들을 나타내고 있다.

표 1. 윈도우즈 지원하는 표준 클립보드 포맷

CF_BITMAP	CF_DIB
CF_DIBV5	CF_DIF
CF_DSPBITMAP	CF_DSPENHMETAFILE
CF_DSPMETAFILEPICT	CF_DISPTXT
CF_ENHMETAFILE	CF_GIOBJFIRST
CF_GIOBJLAST	CF_HDROP
CF_LOCALE	CF_METAFILEPICT
CF_OEMTEXT	CF_OWNERDISPLAY
CF_PALETTE	CF_PENDATA
CF_PRIVATEFIRST	CF_PRIVATELAST
CF_RIFF	CF_SYLK
CF_TEXT	CF_WAVE
CF_TIFF	CF_UNICODE

각 프로그램에서는 각기 다른 포맷으로 데이터를 처리한다. 따라서 메모리나 파일에서나 데이터의 표현 방식이 달라지기 마련이다. 이를 위해 윈도우즈에서는 사용자정의 포맷을 등록하여 사용할 수 있도록 지원하고 있다. 사용자정의 포맷은 레지스트리에 등록이 되었다가 윈도우가 종료되는 순간 레지스트리에서 제거된다.

클립보드 시스템은 공동의 저장 장소를 제공하고 정해진 약속에 따라 데이터를 입출력하므로 불특정 다수의 프로그램이 데이터를 공유할 수 있다. 불특정 다수의 프로그램은 각기 다른 포맷으로 데이터를 구성하기 때문에 윈도우즈에서는 클립보드에 복수의 포맷으로 데이터를 넣을 수 있도록 지원한다. 소스 윈도우에서는 복사 명령을 받았을 때, 자신이 알고 있는 모든 포맷으로 변환하여 클립보드에 데이터를 넣어 둔다. 단, 가장 상세한 서식을 가지는 포맷을 우선적으로 넣고 점차 단순한 포맷을 넣는다.

클립보드에서 붙여넣기를 할 때는 자신이 원하는 포맷이 있는지 먼저 살펴보는데 이때 가장 상세한 포맷부터 점검한다. 만약 클립보드에 자신이 이해하는 포맷 중 아무 것도 없으면 붙여넣기를 할 수 없다. 워드 프로세서의 경우를 예를 들어 pseudo code로 표현하면 표 2와 같다.

표 2. 워드 프로세서에서 붙여넣기 과정

```

if (워드 포맷이 있는가?) {
    워드 포맷으로 붙임
} else if (RTF 포맷이 있는가?) {
    RTF 포맷으로 붙임
} else if (텍스트 포맷이 있는가?) {
    텍스트 붙임
} else if () {
    .....
} else {
    붙여넣기를 할 수 없음
}
    
```

2. Drag&Drop의 구현

Drag&Drop이 구현되기 위해서는 소스 윈도우에서 Drag가 시작될 때의 클립보드에 데이터를 복사하는 기능과 Drop된 목적지 윈도우에서 데이터를 클립보드에서 붙여 넣는 기능을 구현하여야 한다. 소스 윈도우에서는 Drop Source(COLEDropSource)라고 불리는 클래스를 상속받아 원하는 동작을 하도록 구현해야 한다. 목적지 윈도우에서는 Drop Target(COLEDropTarget)이라고 불리는 클래스를 직접 사용하거나 상속받아 원하는 동작을 하도록 수정한

후 사용가능하다.

Drop Source는 소스 윈도우에서 Drag가 시작될 때, Drag 작업 중의 피드백, Drag가 종료될 때의 작업을 책임진다. Drag Source는 기본적으로 COleDataSource라는 클래스의 DoDragDrop이란 함수를 부름으로써 클립보드에 데이터를 저장한다. 기본 동작을 바꾸기 위해서는 Drag Source를 상속받아 동작을 재정의해야 한다.

목적지 윈도우에서는 Drop Target을 Drop을 받아서 작업할 뷰에서 인스턴스를 하나 만들어 두어야 한다. 그리고 프로그램이 실행될 때 레지스트리에 등록해 주어야만 소스 윈도우에서 일어난 Drag 작업의 이벤트를 받아서 Drop이 가능해진다. 또한 Drag중의 목적지 윈도우에 마우스 포인터가 들어왔을 때의 기능의 구현과 마우스 모양을 변경해주는 함수들을 아래 목록에서 정리하였다.

- virtual BOOL OnDrop(COLEDataObject* pDataObject, DROPEFFECT dropEffect, CPoint point);
- virtual DROPEFFECT OnDragEnter(COLEDataObject* pDataObject, DWORD dwKeyState, CPoint point);
- virtual DROPEFFECT OnDragOver(COLEDataObject* pDataObject, DWORD dwKeyState, CPoint point);

OnDrop은 Drag 중에 목적지 윈도우에서 마우스 버튼이 떨어져 Drop될 때의 동작을 처리한다. OnDragEnter와 OnDragOver는 Drag 중 목적지 윈도우에서의 마우스 포인터의 모양을 결정한다. 표 3과 표 4는 실제 소스 코드에서 사용된 부분의 일부를 정리하였다. 아래에서 DROPEFFECT_COPY는 마우스 포인터 우측 하단에 +기호가 추가된 모양을 나타내는 상수이다. 마우스 포인터의 효과를 나타내는 상수는 DROPEFFECT_NONE, DROPEFFECT_COPY, DROPEFFECT_MOVE, DROPEFFECT_LINK와 같이 4개가 등록되어 있다.

표 3. 뷰 헤더파일

```

class CImageCollectorView : public CView{
// Drag&Drop 부분
public:
    COleDropTarget m_OleDropTarget;
    virtual BOOL OnDrop(COLEDataObject* pDataObject, DROPEFFECT dropEffect, CPoint point);
    virtual DROPEFFECT OnDragEnter(COLEDataObject* pDataObject, DWORD dwKeyState, CPoint point);
    virtual DROPEFFECT OnDragOver(COLEDataObject* pDataObject, DWORD dwKeyState, CPoint point);
}
    
```

표 4. 뷰 소스파일

```

void CImageCollectorView::OnInitialUpdate()
{
    CView::OnInitialUpdate();
    m_OleDropTarget.Register(this);
}

BOOL CImageCollectorView::OnDrop(ColeData
Object* pDataObject, DROPEFFECT dropEffect,
CPoint point)
{
    // Drop 될 때의 작업 기술
}

DROPEFFECT CImageCollectorView::OnDragEnter(ColeDataObject* pDataObject, DWORD dwKeyState, CPoint point)
{
    return DROPEFFECT_COPY;
}

DROPEFFECT CImageCollectorView::OnDragOver(ColeDataObject* pDataObject, DWORD dwKeyState, CPoint point)
{
    return DROPEFFECT_COPY;
}
    
```

3. Drag & Drop에서 데이터의 붙여 넣기

Drag&Drop에서 데이터의 붙여 넣기는 목적지 윈도우의 OnDrop 함수에서 실행된다. 즉 Drop될 때의 작업을 데이터 붙여 넣기로 작성하는 것이다. Drop될 때 클립보드에 있는 데이터는 COleDataSource 클래스의 포인터로 넘어온 pDataObject를 통해 확인 및 데이터를 가져올 수 있다.

먼저 데이터를 가져오기 전에 가져올 수 있는 포맷이 있는지 확인해야 된다. 가져올 수 있는 포맷의 확인은 두 가지 방법을 통해 알 수 있다. 하나는 현재 클립보드에 저장된 데이터의 포맷들을 탐색하여 처음부터 끝까지 확인하는 방법과 프로그램에서 처리 가능한 포맷들을 물어보고 있는지 없는지 확인하는 방법이 있다. 일반적으로 대부분의 프로그램에서는 후자의 방법을 통해 전체 처리과정을 줄임으로서 프로그램의 속도를 향상시킨다. 전자의 경우는 BeginEnumFormats 함수와 GetNextFormat 함수를 통하여 모든 포맷을 확인한다. 후자의 경우는 IsDataAvailable 함수를 통하여 해당 포맷의 데이터가 있는지 확인한다.

Ⅲ. 각 웹 브라우저의 이미지 포맷

웹 이미지 수집기를 구현하기 위해 먼저 각 웹 브라우저들이 이미지를 클립보드에 저장하는 포맷과 해당 이미지의 URL(Uniform Resource Locator)을 저장하는 포맷을 알아내야 한다. 이를 위해 앞에서 살펴보았던 포맷 확인 방법 중 BeginEnumFormats 함수와 GetNextFormat 함수를 이용하여 각 웹 브라우저의 지원 포맷을 확인한다.

1. Internet Explorer 7

Internet Explorer 7에서는 링크가 걸린 이미지의 경우 이미지 데이터가 담긴 CF_DIB 포맷으로 데이터가 저장되지 않는다. 따라서 링크가 없는 이미지를 Drag하여 테스트하였다. 이미지는 5가지 포맷으로 클립보드에 저장되었다.

표 5. Internet Explorer 7에서의 저장된 이미지 포맷

CF_DIB
CF_HDROP
HTML Format
UniformResourceLocatorW
UntrustedDragDrop

CF_DIB 포맷은 이미지 데이터가 화면에 출력된 그대로의 데이터를 DIB 포맷으로 클립보드에 저장한 것이다. CF_HDROP 포맷은 인터넷 임시 폴더에 저장된 이미지의 경로를 나타낸다. HTML Format은 이미지 태그가 위치한 HTML 문서 조각을 저장한다. UniformResourceLocatorW 포맷은 실제 이미지의 URL을 유니코드로 저장하는 포맷이다. UntrustedDragDrop 포맷은 Internet Explorer에서 시작된 Drag 라는 것을 알려주는 포맷이다.

2. Firefox 3.5

Firefox는 다양한 포맷을 지원할 뿐만 아니라 링크가 걸려 있는지 아닌지 상관없이 이미지 데이터 포맷을 클립보드에 저장하여 준다. 특히 같은 내용의 텍스트 데이터도 시스템의 환경에 따라 쓸 수 있도록 일반 ASCII 코드와 유니코드 두 가지 포맷으로 구분하여 제공하여 주는 부분도 프로그래머들을 위해 배려된 부분이라 볼 수 있다.

표 6. Firefox 3.5에서의 저장된 이미지 포맷

CF_TEXT
CF_DIB
CF_UNICODETEXT
CF_HDROP
FileGroupDescriptor
FileGroupDescriptorW
HTML Format
Preferred DropEffect
UniformResourceLocator
UniformResourceLocatorW
DragContext
text/html
DragImageBits
text/x-moz-url
text/x-moz-url-data
text/x-moz-url-desc
text/uri-list
text/_moz_htmlcontext
text/_moz_htmlinfo
application/x-moz-file-promise-url
application/x-moz-file-promise-dest-filename

3. Chrome

Chrome은 구글에서 만든 웹 브라우저이다. 프로그램이 상당히 가볍고 빠르면서 웹 브라우저에 필요한 요소들은 다 포함되어 있다. 하지만 Internet Explorer와 Firefox와 달리 이미지를 클립보드에 저장하는 포맷이 좀 다르다. Chrome은 FileContents라는 포맷을 사용하여 이미지 데이터를 저장한다. Internet Explorer는 CF_DIB 포맷을 사용하는데, 이는 화면에 출력된 이미지를 장치 독립적 비트맵으로 저장하는 포맷이다. 실제로 장치 독립적 비트맵 파일로 저장된 포맷과 일치하지만 Chrome의 FileContents 포맷은 파일의 원본 파일의 내용 그대로를 담고 있다. 따라서 JPG나 PNG 혹은 또 다른 압축된 이미지 파일이라면 이를 화면에 바로 출력할 수 없다. 표 7은 Chrome에서 이미지 데이터를 클립보드에 복사한 데이터 포맷들을 나열한 것이다. 세계의 브라우저 모두 UniformResourceLocatorW을 통하여 이미지의 실제 URL을 클립보드를 통하여 데이터를 전달하고 있다.

표 7. Chrome에서의 저장된 이미지 포맷

CF_TEXT
CF_UNICODETEXT
FileContents
FileGroupDescriptorW
HTML Format
UniformResourceLocator
UniformResourceLocatorW
text/html
text/x-moz-url

IV. 웹 이미지 수집기 구현 알고리즘

이미지가 Drop 될 때에 우리는 출력할 수 있는 이미지 포맷을 가져와야 한다. 세 가지 웹 브라우저에서 통일된 이미지 포맷을 제공하고 있지 않지만 Internet Explorer와 Firefox에서는 우리가 바로 출력할 수 있는 CF_DIB 포맷을 제공한다. 또한 만약 우리가 JPG와 PNG 파일과 같은 파일을 처리할 수 있는 라이브러리가 있다면 Chrome 브라우저일 때 FileContents 포맷을 통해 원래 이미지 파일을 처리할 수 있다. 또한 이미지의 URL을 저장하기 위하여 우리는 세 브라우저에서 지원되는 UniformResourceLocatorW 포맷을 통해 리스트를 작성한다. 이 흐름의 중요 부분을 pseudo code로 나타내면 표 8과 같다. GetData 함수 부분은 실제로 아래와 같이 동작하지 않고 인자로 데이터를 받을 버퍼를 넘기고 해당 버퍼에 입력된 데이터를 리스트에 추가하는 일련의 작업이 필요하다. 또한 GetData, GetFileData, GetGlobalData 함수마다 조금씩 다른 동작을 하기 때문에 해당 부분은 MSDN(MicroSoft Developer Network)을 참고하길 바란다.

표 8. Drop될 때 이미지 리스트 추가 pseudo code

```

BOOL CImageCollectorView::OnDrop(ColeDataObject*
pDataObject, DROPEFFECT dropEffect, CPoint point){
    if (pDataObject->IsDataAvailable(CF_DIB))
    {
        이미지리스트.Add(pDataObject->GetData(CF_DIB));
        URL리스트.Add(pDataObject->GetData(UniformResourceLocatorW 포맷번호));
    }
    else if (pDataObject->IsDataAvailable(FileContents 포맷번호))
    {
        이미지리스트.Add(pDataObject->GetData(FileContents 포맷번호));
        URL리스트.Add(pDataObject->GetData(UniformResourceLocatorW 포맷번호));
    }
}
    
```

이렇게 Drop으로 추가된 이미지들을 출력하기 위해서는 OnDraw 함수에서 리스트에 있는 이미지들을 다시 그려주어야 한다. 해당 부분은 윈도우 프로그래밍 관련 서적들을 보면 자세히 설명되어 있다.

V. 결론

웹 이미지 수집기를 구현하면서 Drag&Drop 메커니즘에 대하여 자세하게 알아보았다. 이러한 Drag&Drop 메커니즘에 대하여 자세하게 알게 된다면 직관적인 GUI 구현뿐만 아니라 사용자 인터페이스를 설계할 때에도 도움을 준다. 이러한 사용자 인터페이스를 설계할 때에는 보다 사용자 중심에서 설계를 하여야 한다. 사용자 입장에서 고려하지 않은 인터페이스들은 전혀 활용되지 않는다^{[2][7]}.

앞으로 마우스를 포함한 터치 인터페이스나 멀티 터치 인터페이스는 사용자 경험 자체를 바꾸게 될 것이다. 기존의 키보드의 사용이 완전히 사라지진 않겠지만, 대부분의 데이터 처리가 터치로 이뤄질 시대가 다가왔다. Drag&Drop으로 문서들이 출력되고, 파일들은 실제 책상처럼 바탕화면 한쪽에 쌓여 있고, 인터넷 뉴스를 Drag해서 내 블로그에 인용할 수 있을 것이다. 이처럼 GUI 환경에 Drag&Drop은 앞으로 큰 역할을 할 것으로 기대된다. 따라서 Drag&Drop에 대한 연구와 도전이 계속적으로 필요하다고 생각된다.

참 고 문 헌

[1] 정상훈, 이진표, "PC 사용경험이 PDA 사용에 대한 직관적 인지에 미치는 영향에 관한 연구", *디자인학연구*, 통권 제52호(Vol.16 No.2), pp.301-310, 2003. 05.

[2] 진영범, 김만수, 오일석, "스테레오 입체영상과 3차원 입력 디바이스를 이용한 직관적 사용자 인터페이스 설계", *한국정보과학회 학술발표논문집*, 제24권 제2호(II), pp. 371~374, 1997. 10.

[3] 타도 아이폰에 필요한 것은 UI가 화려한 것이 아닌 사용하기 편하게 만들어주는 것 Available at: <http://poem23.com/entry/타도-아이폰에-필요한-것은-UI가-화려한-것이-아닌-사용하기-편하게-만들어주는-것>.

[4] 이정호, 이운형, "개인용 컴퓨터를 위한 멀티 터치 입력장치에 관한 연구", *한국디자인포럼*,

Vol. 21, pp.233~242, 2008. 11.

[5] Drag and Drop (OLE), [http://msdn.microsoft.com/en-us/library/aa294159\(VS.60\).aspx](http://msdn.microsoft.com/en-us/library/aa294159(VS.60).aspx).

[6] 김상형, "윈도우즈 API 정복 volume 2 개정판", 한빛미디어, 2006. 06.

[7] 유재욱, "사용자 중심적 인터페이스 설계에 관한 연구", *한국비블리아학회지*, 제 12권 1호, 2001. 06.

이 선 응 (Seon-Ung Lee)



2004년 3월~현재 : 한국기술교육대학교 인터넷미디어공학부 재학
관심분야 : 모바일 플랫폼, RIA, UI, UX

문 일 영 (Il-Young Moon)



2000년 2월 : 한국항공대학교 항공통신정보공학과 (공학사)
2002년 2월 : 한국항공대학교 대학원 항공통신정보공학과 (공학석사)
2005년 2월 : 한국항공대학교 대학원 정보통신공학과 졸업 (공학박사)

2004년 ~2005년 : 한국정보문화진흥원 선임연구원
2005년 3월~현재 : 한국기술교육대학교 인터넷미디어공학부 조교수
관심분야:무선 인터넷 응용, 무선 인터넷, 모바일 IP