

# 가상환경에서의 사용자 인터랙션을 지원하기 위한 가상 계측 단말기 스크립팅 툴킷

김석환\*, 정다영\*, 조용주\*\*  
\*상명대학교 소프트웨어학부  
\*\*상명대학교 미디어학부  
e-mail : baseball@smu.ac.kr

## A Toolkit for Constructing Virtual Instruments to Augment User Interactions in Virtual Environments

Seokhwan Kim\*, Dayoung Chung\*, Yongjoo Cho\*\*  
\*Division of Computer Software, Sangmyung University  
\*\*Division of Media Technology, Sangmyung University

### 요 약

본 논문은 가상 계측 단말기 (Virtual Instrument) 제작을 위해 만들어진 가상 계기 스크립팅 툴킷 (VIST)의 구현과 설계를 보여준다. 가상 계기는 가상현실에서 휴대용 컴퓨터에 탑재되어 실행되는 응용 프로그램들을 일컫는다. 가상 계기는 가상환경의 특성에 대해 조사하고 보여주는 데 사용되고 가상환경에서 사용자와 가상현실 시스템의 상호 작용을 개선시키는 데에도 사용할 수 있다. VIST는 가상 계기를 구성하는 컴포넌트를 쉽게 생성하도록 해주고, 그 컴포넌트들을 사용하여 프로토타입을 빠르게 제작할 수 있게 한다. 또한 가상환경 안에서 가상환경의 특성과 가상객체의 속성을 보여주고 그것들을 쉽게 파악하도록 하는 메커니즘을 제공한다. 본 논문은 VIST의 설계 원리와 구조를 기술하고, 실제로 어떻게 가상 계측 단말기를 제작하였는지 보여준다.

### 1. 서론

현실 세계의 계측기는 아날로그 혹은 디지털 디스플레이를 이용한다. 그리고 임베디드 프로세서와 소프트웨어로 세부 장치들을 제어하여 사용자가 원하는 정보를 보여준다. 가상 계측기는 일반 컴퓨터를 사용하여 기존의 계측기를 대체 가능하도록 만들어진 시스템들을 의미한다. 이 논문에서 의미하는 가상 계측 단말기는 현실 세계에서 사용하기 위해 만들어진 계측기와는 달리 가상현실 환경에서 사용자와 가상 환경의 상호작용을 개선하기 위해 만들어진 응용 프로그램을 일컫는다.

지난 몇 년간 초등학생들을 위한 가상 학습 환경을 만들면서 사용했던 가상 계측 단말기(이하 가상 계기)는 주로 가상 온도계, 가상 위치 추적기 (Simulated Global Positioning System) 또는 가상 기압계와 같이 가상환경의 특정 속성들을 보여주는 간단한 정보 뷰어로서의 역할을 하거나 가상환경에서 사용자가 자료를

쉽게 수집할 수 있도록 도와주는 기능을 한다[1,4]. 특히 가상환경에서 가상 단말기를 사용한 학생들은 그렇지 않은 학생들에 비해 자료의 조사량이나 관찰력에서 더 나은 결과를 보인 것으로 나타났다. 또한 가상현실 환경의 취약점으로 분류되는 자신의 위치나 방향을 파악하는 문제[2]에 대해서도 가상 계기를 사용하면 향상된 결과가 나오는 것으로 나타났다.

그 밖에도 가상 계기를 사용하면 데이터가 풍부한 가상 환경을 만들 수 있다. 가령 가상환경을 만드는 개발자는 토양의 습도 또는 바위의 강도나 구성 성분 등과 같은 사용자들이 측정해야 하는 속성들을 포함한 가상환경을 구축할 수 있다. 그리고 사용자들은 가상 계기를 이용해서 이런 가상 환경의 가상 개체와 그것들의 속성을 자세히 분석하는데 도움을 받는다.

이렇게 가상 계기는 가상현실 환경을 이용해 학습 환경을 꾸밀 때 유용한 도구임이 분명하다. 하지만 가상 계기를 개발하는 것은 그다지 쉽지 않다. 휴대용

컴퓨터 기반의 개발 경험이 있어야 하고 가상 환경과의 연결을 위해 네트워크 통신에 대한 지식도 필요하다. 이를 개선하고 해결하기 위해 VIST (Virtual Instrument Scripting Toolkit)라는 가상 계기를 만들 수 있는 도구를 만들었다. 이 논문에서는 그동안 가상 계기들을 만들면서 부각된 요구사항들과 스크립팅 툴킷에 대해서 설명을 할 것이다. 또 VIST 를 이용해 가상 계기를 제작한 예와 앞으로의 연구 방향에 대해서도 설명하겠다.

## 2. 가상 계측 단말기에 대한 요구 사항

가상 계기를 사용한 몇 번의 실험을 통해서 계기는 다음과 같은 4 가지 설계적인 요구사항을 만족시켜야 한다는 것을 확인할 수 있었다.

- **유연함 (Flexibility).** 가상 계기는 다양한 사용자의 요구에 쉽게 부응할 수 있도록 만들어야 한다. 간단한 값을 표현해주는 것도 사용자에게 따라 그 사용자에게 가장 적합한 방법으로 만들어져야 한다. 예를 들면 가상환경에서 사용자의 위치나 방향 등을 보여주는 것도 x, y 좌표, 혹은 위도나 경도, 아니면 지도상에서 사용자의 위치를 점으로 표현해 보여줄 수도 있을 것이다. 그 외에도 사용자에게 따라 정보를 표현할 수 있는 정밀도 등을 조절할 수도 있을 것이다. 가령 일반적으로 어떤 사물의 길이를 표현할 때 1600cm 라는 표현보다는 16m 라는 표현이 더자연스러울 것이다. 하지만 특정 사람들에게는 1600cm 라는 것이 더 적합하게 느껴질 수도 있다.
- **투명성 (Transparency).** 일반 현실 세계에서 온도계는 극한 상황을 제외한 세계 어느 곳에서도 온도를 측정할 수 있어야 한다. 한 곳에서 작동하던 온도계가 다른 곳에 가서 작동을 하지 않는 것은 아니다. 계측기로 사용할 수 있는 가상 계기도 마찬가지로 같은 속성을 가지고 있는 모든 환경에서 그 속성을 측정할 수 있어야 한다. 예를 들어, 들판을 재현한 가상환경에서 온도를 측정해주는 가상 계기가 있다고 가정해보자. 그렇다면 이 계기는 온도라는 속성을 지닌 다른 가상환경에서도 특별한 장치를 부착한다거나 조정을 해주지 않아도 자연스럽게 온도라는 속성을 측정해줄 것이라고 사용자는 생각하기 마련이고 이런 인식을 지원해줄 수 있도록 만들어져야 한다.
- **인터랙션 (Interaction).** 가상 계기는 사용자가 가상환경에서 좀 더 자유롭고 편하게 활동할 수 있도록 도와주는 인터페이스 장치로 사용할 수도 있어야 한다. 처음 가상현실을 접하는 사람은 가상환경에 적응하는데 어려움을 많이 느낀다. 특히 자신의 위치를 파악하고 자신이 가고자 하는 곳으로 제대로 가지 못하는 way-finding [2] 문제에 쉽게 노출되어 있고, 가상환경에 있는 개체를 조작하고 선택하는 것에도 많은 어려움을 느낀다. 이런 문제를 해결하기

위해 가상 계기를 사용할 수 있다. 가령 개체를 선택하는 것은 사용자의 주변에 있는 개체들의 목록을 만들어 계기에 보여주고 선택하도록 한다면 사용자는 개체를 훨씬 쉽게 선택할 수 있을 것이다.

- **가벼움 (Lightweight).** 휴대용 컴퓨터는 데스크탑 컴퓨터에 비해 한정된 자원(CPU 성능, 메모리, 저장매체 크기)을 가지고 있다. 따라서 가상 계기는 전체적인 프로그램의 구성을 단순화시키고 가볍게 만들어야 한다. 이는 가상 계기에서 다양성을 지원하기 위해 너무 많은 기능들을 한 개의 프로그램에 넣기 보다는 최소한의 기능만을 가진 프로그램을 작성하는 것이 더 바람직하다는 것을 의미하기도 한다.

이렇게 가상 계기에서 필요하다고 요구되는 사항들 중에는 사용자를 염두에 두고서 제기된 것들도 있고 사용하고자 하는 컴퓨터 기기의 제한성 때문에 발생된 것들도 있다. 어떤 이유에서 만들어졌든 가상 계기를 만들 때에는 이런 요구사항들을 염두에 두고 만들어야 할 것이다.

## 3. VIST

우리는 위에 언급한 4 가지 요구사항을 만족하면서도 쉽게 가상 계기를 만들 수 있도록 도와줄 수 있는 도구인 VIST 를 제작하였다. VIST 를 이용하면 초보 프로그래머도 쉽게 가상 계기의 프로토타입을 제작할 수 있다. 특히 VIST 제작의 주된 목적은 가상 계기 제작의 어려움을 줄임과 동시에 GUI 및 계측기 컴포넌트의 재사용을 높이는 것이었다.

VIST 는 스크립트와 메시지 전달 메커니즘을 사용하여 유연성을 높이려 했다. 우선 계기의 응용 프로그램들은 사용자가 정의한 스크립트 기반의 설정 파일을 통해서 동적으로 생성된다. 다시 말하면 가상 계기의 응용 프로그램에서 사용되는 모든 사용자 인터페이스, 계측, 디스플레이, 혹은 통신 모듈 등은 사용자가 만들어놓은 설정 파일에 의해 레이어아웃을 잡고 생성된다. 뿐만 아니라 보이지 않는 로직에 관련되거나 하드웨어를 담당하는 모듈도 역시 설정 파일에 의해서 제어된다. 이렇게 VIST 에서는 프로그램이 처음 실행될 때 설정 파일을 이용해 프로그램을 동적으로 재설정하고 배치하므로 설정 파일의 스크립트 코드 몇 줄만 수정하면 새로운 모듈의 삽입 및 삭제를 쉽게 할 수 있다.

VIST 에서 사용자와의 인터랙션 혹은 각 모듈간의 인터랙션은 이벤트와 메시지를 통해서 일어난다. VIST 에서 이벤트란 사용자와의 인터랙션을 담당하는 부분, 즉 버튼을 누른다던가 아니면 목록을 보여주는 리스트에서 아이템 한 개를 선택한다던가 하는 부분과 네트워크에서 데이터를 받았다던가 내장되어 있는 카메라에서 화상을 입력 받는 등 주로 프로그램 외부에서 전달되는 모든 자극(stimulus)을 의미한다. 그리고 이런 입력 자극 외에도 각 모듈에서 만들어 전달해주는 이벤트가 있을 수도 있다. 그림 1 의 코드 10-14 를 보

```

1 <VISUAL_COMPONENT>
2 <NAME>Plant6</NAME>
3 <FILE>ImageButton.dll</FILE>
4 <TYPE>VAIImageButtonClass</TYPE>
5 <TEXT>plant6.gif</TEXT>
6 <X>185</X>
7 <Y>110</Y>
8 <WIDTH>50</WIDTH>
9 <HEIGHT>50</HEIGHT>
10 <ACTION event="OnClick">
11 SelectedList.addItem("plant6.20.gif");
12 SelectedList.addSubItem(0, 1, $XPos);
13 SelectedList.addSubItem(0, 2, $YPos);
14 </ACTION>
15 </VISUAL_COMPONENT>

```

그림 1 가상 단말기의 이벤트 처리, 메시지 패싱

면 “OnClick” 이벤트를 처리하는 부분을 보여주고 있다. “OnClick” 이벤트는 버튼 모듈에서 사용자가 버튼을 마우스의 버튼으로 클릭했을 때 발생해 개발자가 버튼이 눌러졌을 때 처리해야 하는 일을 지정할 수 있도록 해준다.

VIST의 개발자는 각 모듈에서 일어날 수 있는 이벤트를 지정하고 그런 이벤트들을 처리할 수 있는 핸들러(Handler)를 스크립트로 작성해 설정 파일에 넣을 수 있다. 그리고 가상 계기 프로그램들이 처음 실행될 때 이벤트 핸들러 역시 동적으로 메모리에 올리고 특정 이벤트가 일어날 때마다 동적으로 실행시키게 된다. 그래서 개발자들이 이벤트를 처리하는 방법을 쉽게 바꿀 수 있도록 만들었다. 그림 1의 “OnClick” 이벤트 핸들러를 보면 버튼이 눌러질 때마다 SelectedList 라는 이름의 리스트뷰 인터페이스에 적절한 아이콘 이미지와 그 버튼이 눌러졌을 당시의 사용자의 X, Y 위치를 추가하도록 되어 있는 것을 확인할 수 있다 (그림 1의 11-13 참조).

VIST에서는 각 모듈에 정의된 특정 코드를 실행시키기 위해 메시지를 사용한다. 우선 가상 계기의 개발자들은 각 모듈에 정의되어 있는 메시지를 전달해 모듈의 특정 코드를 실행시킬 수 있고 각 모듈의 메시지 핸들러에서는 주어진 메시지를 단순히 처리하고 끝내거나 실행 결과를 반환할 수도 있다. 예를 들어 그림 1의 11에 있는

SelectedList.addItem(“plant.6.20.gif”); 라는 코드를 보면 SelectedList 라고 이름 붙은 모듈에 addItem 라는 메시지를 전달하면서 “plant6.20.gif”라는 문자열을 메시지의 인자로 전달하라는 내용이 된다. 마찬가지로 12 줄 째에 있는 코드는 역시 같은 모듈에 addSubItem 이라는 메시지를 전달하며 0, 1, 그리고 전역변수인 \$XPos 의 값을 인자로 전달하라는 의미가 된다.

VIST는 앞에서 말한 것처럼 휴대용 컴퓨터의 제한된 시스템 리소스를 활용하며 부담을 최소화시키고 개발을 쉽게 할 수 있도록 가벼운 컴포넌트 구조를 사용했다. 즉 VIST에서는 CORBA 나 Java Beans 와 같이 잘 알려졌지만 복잡한 컴포넌트 구조를 사용하기 보다는 자체 제작한 구조를 가진 아주 간단한 컴포넌

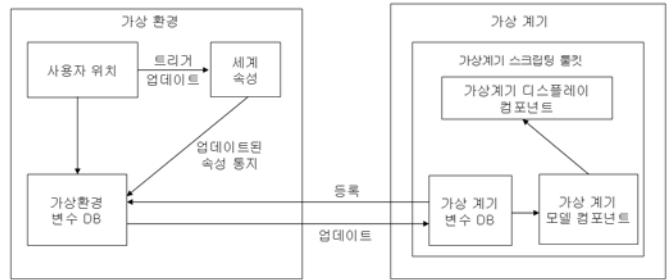


그림 2 가상 단말기와 가상환경의 결합성을 최소화시킨 디자인인 VIST의 구조.

트를 사용하였다. VIST의 현재 버전에서는 단순히 CreateComponent, Destroy, Initialize, Message 등 4개의 추상 인터페이스만을 사용하여 제작할 수 있다. 또한 VIST는 이미 Message를 제외한 인터페이스의 내용을 제공하기 때문에 모듈을 만드는 개발자는 실질적으로 각 모듈에서 정의하는 메시지를 처리해주는 Message 인터페이스만 구현하면 된다. 그림 1에서 보이는 SelectedList 모듈의 경우에는 addItem 이나 addSubItem 등과 같은 메시지를 Message 인터페이스에서 처리하게 될 것이다.

또한 VIST에서는 새로운 컴포넌트 구조를 디자인하고 그것들을 처리하는 부분을 프로그램에 추가해 시스템에 부담을 주지 않기 위해 운영체제의 동적 로딩 기능 (dynamic loading mechanism)을 최대한 활용했다. 즉 VIST에서는 마이크로소프트 윈도우즈의 Dynamic Linking Library 와 유닉스 계열의 Shared Library 같은 최신 운영체제의 동적 모듈 로딩 메커니즘을 이용해 구현되었다. 이를 통해 컴포넌트 관리를 위한 새로운 코드가 추가되지 않아 제한된 리소스를 가지고 있는 휴대용 컴퓨터에서도 좀 더 쾌적한 실행을 보장받도록 했다.

요구사항 중 투명성을 지원하기 위해서는 한 개의 가상 계기 프로그램을 만들어 여러 가상환경에서 사용할 수 있어야 한다. 이를 위해 VIST는 그림 2에서 보이는 것처럼 가상 환경과의 결합성을 최소화시켰다. 그림 2에서 보면, 가상 계기가 시작되면 그 계기가 받아서 디스플레이할 수 있는 속성의 이름들을 가상 환경 속성 데이터베이스에 등록을 하게 된다. 즉 가상 온도계를 사용한다면 “온도”라는 이름으로 등록을 하게 된다. 그리고 온도는 속성을 지닌 가상환경에서 가상 온도계를 사용하면 사용자의 위치에 따라 데이터베이스에는 온도가 지속적으로 업데이트 된다. 궁극적으로 업데이트 된 온도는 가상 계기에 전해져 항상 새로 업데이트된 값을 보여주게 되는 것이다. 만약 가상환경에 계기에서 등록한 속성이 따로 존재하지 않는다면, 아무런 일도 일어나지 않게 된다. 이런 방법은 결국 가상 계기의 투명성을 높여주어 특정 속성을 가지고 있는 모든 가상환경에서 가상 계기를 활용할 수 있도록 해준다. 사용자의 입장에서는 특별한 지시 없이 계속 장비를 사용할 수 있게 되는 것이다.



그림 3 왼쪽 이미지는 가상필드에서 사용한 가상단말기를 보여주고 있고 오른쪽에는 그 단말기에서 어떻게 이벤트와 메시지가 처리되는지 보여준다.

#### 4. 가상 계기 제작을 위한 VIST의 활용 예

지난 몇 년간 가상환경에서 어린이들이 과학적 탐구와 자료 수집하는 것을 돕기 위해 다수의 가상 계기를 개발했다 [1]. 초기에는 마이크로소프트사의 포켓 피씨 API와 C++언어를 이용해 구현했지만, 나중에는 유연성 (flexibility), 사용성 (usability), 그리고 유지보완성 (maintainability)을 위하여 VIST를 사용했고 많은 효과를 보았다. 예를 들어, 가상필드 [3]를 위한 계기는 C++와 포켓 피씨 API, 그리고 직접적인 네트워크 프로그래밍 등 때문에 초기 개발에 수개월이 소요되었다. 그러나 VIST를 이용한 뒤로는 개발자는 큰 어려움 없이 같은 기능을 가진 프로그램을 몇 주만에 개발할 수 있었다. 특히 이는 윈도우즈나 휴대용 컴퓨터 프로그래밍을 많이 해보지 않은 프로그래머들에게 더욱더 두드러지게 나타나는 현상이었다.

그림 3은 가상필드에서 사용한 계기를 보여주고 있다. 이 계기에는 GPS, 이미지 버튼, 리스트뷰 등의 사용자 인터페이스 모듈과 네트워크 통신을 담당하는 TCP 모듈 등으로 구성되어 있다. 우선 GPS 모듈은 가상환경에 있는 사용자의 위치와 방향을 보여준다. TCP 모듈은 가상환경으로부터 데이터를 받으면 "OnReceived"라는 이벤트를 만들어내게 되고 이는 결국 TCP 모듈의 "OnReceived" 이벤트 핸들러에 의해 처리되게 된다. 또 사용자가 이미지 버튼을 누르면 그림 1에서 이미 설명한 바와 같이 가상 계기의 버튼 모듈은 "OnClick" 이벤트를 발생시키고 그 버튼의 이벤트 핸들러는 GPS 모듈을 통해 사용자의 방향을 추적하고 위치에 따라 사용자가 수집할 수 있는 자료 목록을 만들어 리스트뷰 모듈을 업데이트 한다.

우리가 실험했던 또 다른 가상 환경 중에 가상 대양(ocean)이 있었다. 이를 위한 가상 계기는 휴대용 컴퓨터 프로그래밍 경험이 전무한 프로그래머가 개발했다. 계기에는 수심에 따른 압력, 온도, 염도, 빛 투과력과 같은 바다 속의 환경적 속성을 표시할 수 있게 개발 되었다. 얼마 후 이 계기는 실험을 위해 수심, 빛 투과력, 염도를 표시하는 부분과 수심, 압력, 온도를 보이는 부분 두 가지로 분리되어야 했다. 만약 C++와 APIs를 사용하여 만든 가상 환경의 경우에는 계측기의 속성을 수정하기 위해선 운영체제, API의 기본적인 사항과 네트워크 통신에 관한 지식이 필요하지만 VIST를 이용해 만들 경우에는 필드에 대한

지식이 필요하다. 뿐만 아니라 이런 지식이 있다고 하더라도 그 프로그래밍의 복잡성은 이루 말할 수 없다. 하지만 VIST를 이용해 만든 경우에는 그 개발자는 이를 설정 파일을 약간 고치는 것만으로 계측기의 속성을 아주 간단히 분리 할 수 있었다.

#### 5. 결론과 향후 연구 방향

이 논문은 컴포넌트 기반 가상 계기 제작 도구인 VIST의 구현과 설계에 대해 설명하고 있다. 가상 계측기는 가상환경에서 사용자의 인터랙션을 쉽게 할 수 있도록 휴대용 컴퓨터에서 실행되는 응용 프로그램이다. 가상환경에서 가상 계기는 유용하기는 하지만 이를 개발하기 위해선 프로그래밍 언어와 휴대용 컴퓨터에 대한 이해가 수반되어야 했다. 우리는 그동안 개발해온 경험에 기초하여 4 가지 설계적 요구사항을 정리하였고 이를 해결하기 위해 가벼운 컴포넌트 기반 스크립팅 툴킷을 계획하고 개발하였다.

전에 만들어진 휴대용 컴퓨터 응용 프로그램들은 주로 사용자와 가상 세계간의 인터랙션에 주안점을 두었지만 VIST는 여기에 추가로 앞서 설명되었던 가이드라인을 충족시키기 위해 개발되었다. 또 VIST는 가상 세계의 속성을 표시하기 위한 가상 계기를 쉽게 제작할 수 있게 만든다는 것도 확인할 수 있었다. 우리는 지속적으로 VIST를 이용하여 가상 계기를 제작하려 하고 또 필요한 모듈들을 개발해 업데이트 할 예정이다. 앞으로의 연구 방향은 초보 프로그래머에게 VIST를 교육시킨 후 얼마나 빠르게 가상 계기를 제작할 수 있는가에 대해서 실험해보는 것과 좀 더 사용자들이 쉽게 가상환경과 상호작용할 수 있도록 도와줄 수 있는 모듈을 제작하는 것에 대해서 연구할 것이다.

#### 참고문헌

- [1] 조용주, 박경신, "가상 필드에서 초등학생들의 과학적 탐구를 도와주는 멀티미디어 보조 도구", 한국정보처리학회논문지 제 12(B)권 2호 게재예정, 2005.4.
- [2] Vinson, N., Design Guidelines for Landmarks to Support Navigation in Virtual Environments, Proc. of ACM Human Factors in Computing Systems, 278-285, 1999.
- [3] Johnson A., Moher T, Cho Y, Edelson, D, Reiser B.: Sixth Grader doing science Collecting Data and Looking for Patterns in a Virtual Field. Proc.of IEEE Virtual Reality, 281-283, 2000.
- [4] Park K, Leigh J, Johnson A, Carter B, Brody J, Sosnoski J.:Distance Learning Classroom using Virtual Harlem. In Proc. of IEEE International Conference on Virtual System and Multimedia, 489-498, 2001.