사용자 인터페이스 마크업 언어 표준 분석 및 설계

*김성한,*이승윤
*한국전자통신연구원 표준연구센터 서비스표준연구실,

Analysis and Design for User Interface Markup Language Standardization

* Sunghan Kim, *Seung-yun Lee
*Electronics and Telecommunications Research Institute
E-mail: {sh-kim, syl}@etri.re.kr

요 약

본 논문은 사용자와 시스템간의 인터페이스 관련 표준화를 추진 중인 ISO/IEC/JTC1의 SC35 UI의 제스처 기반 UI 요소 및 요구사항 및 제스처 기반 UI 표준 프레임워크를 분석하고, 이를 기반으로 GIML(Gesture-based Markup language) 마크업 언어 설계에 대해 제안 한다.

ABSTRACT

This paper analysis gesture-based UI requirements and framework in ISO/IEC/JTC1 SC35 standardization activity, and propose design of GIML(Gesture-based Markup language).

키워드

사용자 인터페이스 마크업언어, 표준화, 제스쳐 응용

Ⅰ. 서 론

정보통신 기술의 발전으로 사용자가 언제 어디서나 환경에 적합한 스마트 디바이스를 이용하며, 따라서 사용자가 다양한 스마트 디바이스와 자연스럽고 편리한 상호작용을 할 수 있도록 사용자의 움직임을 이용한 제스처 기반 사용자 인터페이스에 대한 요구가 증가하고 있다. 제스처 기반 사용자 인터페이스는 다른 사용자 인터페이스에 비해서 비교적 직관적이고 간단하기 때문에 스마트 디바이스와 사용자와의 상호작용에 효과적이다.[1]

최근, 새로운 사용자 인터페이스 입력장치가 등장함에 따라 어플리케이션들도 새로운 제스처를 정의하여 사용자에게 새로운 사용자 경험을 제공하고 있다. 컴퓨터 시스템이 다양한 사용자 인터페이스 입력장치를 수용하고 입력장치에 종속적이지 않은 제스처를 정의하고 지원하기 위해서는 컴퓨터 시스템이 새롭게 정의된 제스처를 인식하여야 한다. 어플리케이션 개발자가 정의한 새로운 제스처를 컴퓨터 시스템에서 인지될 수 있도록 하기 위해서는 제스처를 XML 형식의 구조화된 제스처 표현 형식의 표준이 요구된다.

본 논문은 사용자와 시스템간의 인터페이스 관련 표준화를 추진 중인 ISO/IEC/JTC1의 SC35 UI의 제스처 기반 UI 요소 및 요구사항 및 제스처 기반 UI 표준 프레임워크를 분석하고, 이를 기반으로 GIML(Gesture-based Markup language) 마크업 언어 설계에 대해 제안한다.

Ⅱ. 기능 분석

2.1 GIML 기능 요구사항

싱글 포인트(single point) 제스처는 마우스, 스타일러스(stylus), 핑커 팁(fingertip), 손 등에 의한 싱글 포인트 장치에 의해 만들어진다. 제스처는 플랫폼, 시스템, ICT(Information and Communications Technology)를 실행하는 어플리케이션 등에 의해 명령(command)으로 해석된다. 만일 사용자가 제스처를 자연스럽게 여기고 편리

하게 입력할 수 있다면, 사용자는 어떻게 사용하는지를 배우지 않고 빨리 제스처를 ICT 시스템 어플리케이션에 활용할 수 있다. 어플리케이션에서 싱글 포인트 제스처를 사용하는 몇몇 상용 ICT 시스템들이 있다. 이러한 ICT 장치를 포함하는 시스템은 개인용 컴퓨터, 스마트 TV, 스마트 폰, 비디오 게임 콘솔 등이 있다. 반면에 싱글 포인트 제스처를 정의하는 국제 표준은 제정되지 않았다. 제스처가 서로 다른 명령어들로 대응이 될 때에는 사용자들이 혼란스러워했다. 이러한 싱글 포인트 제스처의 다양성과 불일치성은 ICT 시스템의 어플리케이션에서 사용의 어려움을 겪었다. 본 장에서는 싱글 포인트 제스처의 기술과 관련된 제스처 명령어들과 ICT 시스템 어플리케이션의 공통적인 기능들의 요구 사항들을 살펴본다. 시스템 레벨 기능들은 운영체제 또는 플랫폼에서 수행되는 선택 또는 초기화 기능들을 포함하고 있다. 어플리케이션의 공통적인 기능들을 살펴보며, 이러한 기능들에는 메뉴 선택(menu navigation), “도움말(help)”, “되돌아가기(undo)”, “재실행하기(redo)” 등이 있다. 표준 싱글 포인트 제스처는 멀티 포인트(multi-point) 제스처와 조화를 이루어야 한다. 사용자는 표준 싱글 포인트 제스처들을 사용하여 쉽고 혼란없이 어플리케이션을 실행할 수 있어야 한다. 제스처는 사용자에 의해 특정한 인식 방법이나 상호작용 방법 또는 장치에 관계없이 수행될 수 있어야 한다.

본 장은 GIML 기능 요구와 관련된 시스템 레벨 기능과 ICT 시스템의 어플리케이션과 관련된 공통 기능과 관련된 싱글 포인트 제스처를 살펴본다[8][9]. 싱글 포인트 제스처는 마우스, 스타일러스, 핑커 팁, 손 등을 이용하여 수행된다. 싱글 포인트 제스처는 시스템, 플랫폼, 또는 장치와 관계없이 실행될 수 있어야 한다. 제스처의 정의를 하는데 사용자와 사용자의 인식 관점에 초점을 맞추었다. 시스템 레벨 기능은 시스템 또는 플랫폼 레벨에서 수행되는 기능을 의미한다. 이것은 어플리케이션의 초기화(initiation), 재복귀(resume), 재시작(restart), 종료(termination) 기능들을 포함한다. 어플리케이션 사이의 공동적인 기능은 시스템 또는 플랫폼의 어플리케이션에서 공통적으로 수행되는 기능들을 의미한다. 이것은 메뉴 찾기(navigation of menus), 열기(open) 동작, 닫기(close) 동작, 다시 불러오기(reload) 등의 기능들을 포함한다.[2]

2.1.1 제스처 방식

마우스 제스처는 마우스, 조이스틱, 트랙볼, 스타일러스 등의 single pointing 장치들의 연속적인 행동들을 포함한다. 마우스 포인팅과 같은 행동들과 다른 행동들의 제스처를 구분하기 위해서는 사용자가 제스처 입력을 어떻게 활성화하는가를 알아야 한다. 예로 두 버튼이 있는 마우스의 경우, 오른쪽 버튼이 제스처를 활성화시킨다면, 제스처는 초기 상태로 진입한다. 제스처 입력의 잘못된 인식을 최소화하려면, 시스템은 제스처 생성 시에 포인팅 장치의 작은 또는 아주 큰 움직임은 무시해야 한다. 마우스 제스처의 경우 가로, 세로, 또는 대각선 방향으로 30 픽셀 보다 움직임이 커야 한다. 제스처 입력이 활성화된 후로는 시스템은 중간 상태로 진입한다. 이 때, 사용자에게 피드백(feedback)과 피드-포워드(feed-forward)를 제공하기 위하여 움직임의 상태(마우스의 움직임과 가이드를 표시하는)가 화면에 표시되는 것이 권장된다. 이러한 것들은 시스템과 플랫폼의 활용도를 높여준다.

터치 제스처는 터치패드 또는 터치 스크린에서 집기(pinching), 움직이기(swiping) 같은 일련의 손가락 행동들을 의미한다. 터치 패드와 터치 스크린은 터치 제스처의 전형적인 입력 장치이다. 사용자가 스크린 또는 패드를 건드리는 순간 제스처의 초기 상태가 시작된다. 그리고, 접촉 행동이 진행됨에 따라(예로, 집기(pinching), 움직이기(swiping), 유지하기(holding) 중간 상태가 시작된다. 하지만, 다음과 같은 싱글 포인트 터치 제스처들은 유일한 제스처들로 고려된다.

- 터치(Touch): 터치 제스처는 패드 또는 스크린을 손가락으로 행동을 취할 때 수행된다. 입력 장치는 손가락의 압력을 감지하여 실행된다. 제스처는 물체를 선택하거나 터치 인터페이스를 활성화시키는데 사용된다.
- 홀드/유지하기(Hold): 홀드 제스처는 패드 또는 스크린에서 손가락을 일정한 시간 동안 움직이지 않고 유지할 때 실행되는 행동이다. 홀드 제스처와 태핑(tapping) 제스처의 차이는 패드 또는 스크린을 건드리는 시간의 차이이다.
- 누르기(Press): 누르기 제스처는 패드 또는 스크린에 홀드 행동과 터치 행동의 조합으로 수행된다. 입력 장치는 손가락의 압력과 손가락으로 건드리는 영역의 변화로 감지한다.
- 스와이프(Swipe): 스와이프 제스처는 4 방향 움직임 제스처다. 스와이프 제스처는 속도가 있는 행동이다. 스와이프 제스처의 특정한 속도는 해당하는 입력 장치에 따라 정의된다.

손 제스처는 손 제스처 명령을 실행하는 손 행동의 연속이다. 이 제스처는 3차원 인식 공간에서 인식되고 실행된다. 손 제스처는 손의 모양과 손가락의 움직임에 의해 생성된다. 이 제스처는 손 모양을 유지하기 또는 모양을 바꾸는 등의 다양한 방법으로 실행된다 (잡기 또는 튀기기 등).

손의 모양은 입력 장치에 따라 바뀔 수 있다. 일단 손 제스처가 실행되면, 손 모양이 초기 상태와 같이 중간 상태에서도 유지될 수 있다. 손의 “펴기” 모양은 초기 상태를 의미한다. 사용자가 편 손을 움직이면, 중간 상태로 진입한다. 제스처의 행동을 완료하고 나서, 사용자는 손을 쥠으로써 마지막 상태로 진입 또는 제스처를 종료할 수 있다. 만일 손 제스츄얼르 실행 중 손 모양이 바뀌면, 멀티 포인트 제스처로 간주한다.[3][4]

2.2 GIML 분석

GIML은 ISO/IEC JTC SC35 DIS 30113-11에 정의된 제스처 표준을 기술하는 GIML 문서 구조, 엘리먼트, 속성을 정의하는 것이다.

이 국제 표준은 특정한 인식 기술 또는 특정한 HCI 상호작용 방법 또는 장치와 관련없이 ISO/IEC 30113-11 에 정의된 표준 제스처들을 기술하는 확장된 마크업 언어를 정의한다. 이 것은 GIML 기능 모델, GIML 구조, GIML 문법, GIML 의미 문법, 그리고 예제들을 기술한다. 이 범위는 제스처에 바탕을 둔 응용, 제스처 처리(매칭) 장치, 사용자 제스처 인터페이스 하드웨어 장치, 응용과 제스처 처리 장치의 인터페이스를 포함한다. 이 범위는 제스처 처리 장치 그 자체를 포함하지 않는다.

2.2.1 기능 모델

기능 모델을 그림 1의 GIML을 위한 다이어그램에 보였다. GIML은 JTC1 SC35 제스처를 사용하여 제스처 결과를 제스처 응용에 제공한다. GIML은 XML 언어에 바탕을 두고 있으며, JTC1 SC35 제스처의 특징과 문법을 기술하는데 사용된다. TV는 제스처에 바탕을 둔 양방향 서비스를 제공할 수 있으며, 마우스(mice), 터치 스크린(touch screens), 터치 패트(touch pads), 3D 마우스(mice), 조이스틱(joysticks), 게임 제어기(game controllers), 유선 장갑(wired gloves), 깊이 카메라(depth-aware cameras), 스테레오 카메라(stereo cameras), 웹 카메라(Web cameras),립 모션(Leap Motion) 등을 지원할 수 있다.

기존의 방법들은 제스처 자료를 다루는 표준 형식이 부족하다. 이러한 이유로 제스처로 제어하는 양방향 멀티미디어 시스템과 말을 사용하지 않는 제스처에 바탕을 둔 내용 처리를 지원하기 위해 쉽게 확장이 가능한 제스처 기술 언어가 필요하다. 만일 이것을 사용할 경우, 우리는 응용에 일관적인 플랫폼에 독립적인 사용자 인터페이스를 제공할 수 있다.

GIML은 제스처 이름과 자료를 사용자 제스처 인터페이스 하드웨어 장치로부터 받아 응용으로 전달한다. 제스처 매핑과 장치 제스처로부터 응용 제스처로 변환하는 것이 필요하다. 이것은 XML을 사용하여 GIML로 기술한다.

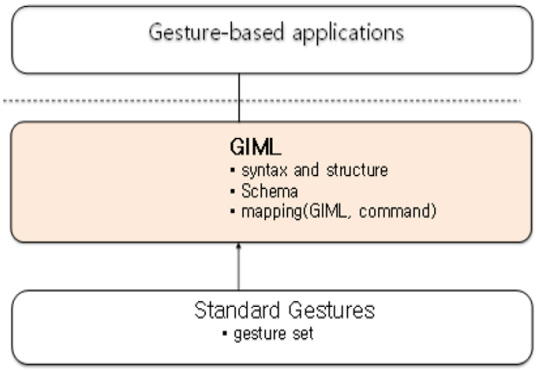

그림 1. GIML 개념적인 기능 다이어그램

2.2.2 GIML 문법과 구조

GIML 스키마는 그림 2에 보인다. <gesture> 엘리먼트는 제스처 문서의 제일 위 root 엘리먼트이다.

- <environment> 엘리먼트는 기본 environment attribute 변수들을 제스처에 할당한다. <environment> 엘리먼트는 제스처 인식 블록의 입력을 정의한다.
- <result> 엘리먼트는응용으로 전달할 제스처 결과를 정의한다. <result> 엘리먼트는 제스처 인식 블록의 출력을 정의한다.
- <initial> 엘리먼트는 초기 environment attribute를 할당한다.
- <point> 엘리먼트는 point 객체를 위한 attribute 값을 설정한다.
- <cluster> 엘리먼트는 cluster 객체를 위한 attribute 값을 설정한다.
- <event> 엘리먼트는 제스처가 종료되었을 때의 조건을 가르키는 gesture_event_status attribute 값을 설정한다.
- <property> 엘리먼트는 ref attribute variable의 값을 설정한다.
- <gesture_event> 엘리먼트는 하나 또는 하나 이상의 제스처 이벤트 어노테이션을 의미한다. 이것은 객체의 인벤트를 기술한다.

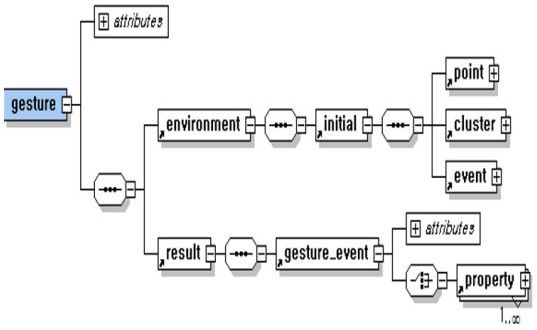

그림 2. GIML 스키마

III. 설 계

3.1 프로토타입 시스템 설계

본 장에서는 GIML에서 정의한 제스처 관련 API 프로토타입 시스템에 대해서 설명한다. 프로토타입 시스템 개발은 Hammer.js를 기반으로 한다. Hammer.js는 터치, 마우스, 포인터 등과 같은 입력장치를 통해 만들어진 제스처를 입력장치에 관계없이 어플리케이션을 개발할 수 있도록 하는 오픈소스 라이브러리이다.

본 연구에서 기반한Hammer 라이브러리는 마우스, 포인터 이벤트, 터치를 포함하여 새로운 카메라, 3D 제스처와 같은 새로운 제스처의 인식에도 확장이 가능하다. 또한 Hammer라이브러리는 현재 v2.0.4가 출시되었으며 라이브러리가minified gzipped 형태일 때 3.96 kB이다. Hammer v2.0.4는 동시에 여러 개의 Manager 객체를 지원하기 때문에 여러 사람이 동시에 사용하는 것이 가능하며 제스처 인식기를 재사용할 수 있으며 touch-action css를 사용할 수 있도록 하여 최근의 모바일 브라우저를 지원한다. 본 연구에서 개발한 프로토타입은 Hammer 라이브러리를 기반한다.

또한 jQuery와 Angular.js를 지원할 수 있는 확장 기능도 제공되고 있으며 싱글 터치 뿐만아니라 멀티 터치의 제스처도 지원이 가능하도록 구조가 설계되어 있으며 멀티 터치를 에뮬레이션하는 에뮬레이터도 함께 제공되고 있다.

Hammer라이브러리는 다음 그림 3에 나타낸 제스처 기반 어플리케이션 실행 환경의 제스처 인식 소프트웨어 플랫폼에 해당한다. Hammer 라이브러리는 마우스, 포인터, 터치와 같은 제스처 입력장치에서 발생한 이벤트 또는 수집 데이터를 처리할 수 있는 이벤트 핸들러 기능을 가진다. 이는 Input 객체로 추상화된다. Input객체는 Manager객체와 관련 제스처 인식을 담당하는 제스처 인식기(gesture recognizer) 객체와 메시지 전달을 통해 제스처 입력장치에서 수집한 움직임 정보를 이용하여 제스처를 인식할 수 있도록 한다. 웹 어플리케이션은 제스처 입력의 대상이 되는 객체와 객체와 관련하여 인식되어야 할 제스처를 Manager 객체에 등록한다. Manager객체는 컴퓨팅 시스템의 제스처 입력장치를 감지한 후 해당 입력장치의 이벤트 핸들러를 구동시킨다. 이처럼 제스처 입력장치에서 전달된 정보들은 Input 객체와 Recognizer 객체간 인터페이스를 통해서 전달되어 처리된다. Hammer 라이브러리는 3차원 제스처 입력장치 등 새로운 제스처 입력장치를를 수용하고 새로운 제스처 인식 알고리즘을 적용을 쉽게 할 수 있는 구조를 가지고 있다.

향후 Hammer 라이브러리를 이용하여 kinect 또는 Leap Motion과 같은 3차원 제스처 입력장치를 지원하고 ISO/IEC/JTC1 SC35에서 표준화를 진행중인 3차원 제스처 인식 알고리즘의 개발 시 활용할 수 있다.

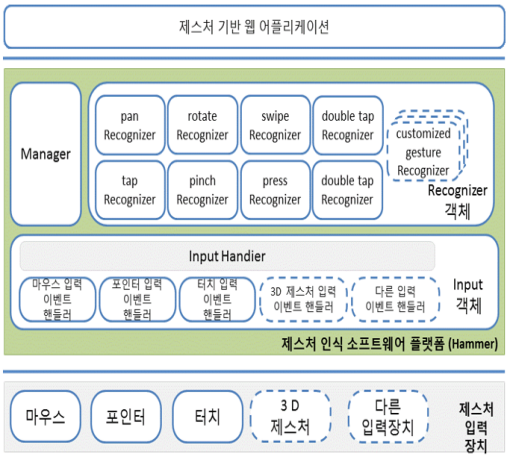

그림 3. GIML 어플리케이션 실행환경

IV. 결언

본 논문에서는 SO/IEC/JTC1의 SC35 UI의 제스처 기반 UI에서 진행되는 GIML의 설계에 대해 상위레벨에서 언급하였다. 향후 관련 표준 기술 및 구현 작업이 요구된다.

본 연구는 미래창조과학부 및 정보통신기술연구진흥센터의 정보통신·방송 연구개발사업의 일환으로 수행하였음. [R0166-15-1002, 융합기반 웹 표준 개발]

참고문헌

[1] 정혁, “제스처 기반 UI 표준화 현황”
[2] GIML Markup Language, NWI proposal, ISO/IEC JTC1 SC 35
[3] Information technology - User Interface, ISO/IEC JTC1 SC 35 30113
[4] Information technology: Framework for describing user interface objects, actions, and attributes, ISO/IEC TR 11580:2007