

상황인지 지원을 위한 규칙엔진 기반 안드로이드 인터랙티브 사이니지 설계 및 구현

차준엽*, 정선태**
*승실대학교 정보통신공학과
junyup44@gmail.com

Design and Implementation of Rule Engine based Android Interactive Signage for Context Awareness

Jun-Yeob Cha*, Sun-Tae Chung**
*Dept. of Information and Telecommunication Eng, Soongsil University

요 약

현재 전개되고 있는 대부분의 인터랙티브 사이니지는 SI 작업으로, 해당 응용 및 적용 현장에 맞게 하드코딩을 하여 고정된 방식으로 구현되고 있다. 따라서, 또 다른 현장의 응용 목적 사이니지 개발을 위해 다시 엔지니어가 재프로그래밍하여야 하는 불편 및 낭비가 존재한다. 본 논문에서는 안드로이드 사이니지 플레이어 환경에서의 더욱 유연하게 확장되고 개선된, 상황인지 지원 안드로이드 인터랙티브 사이니지 시스템 설계와 구현에 대한 개발 결과를 보고한다.

1. 서론

디지털 사이니지는 원하는 다양한 광고, 디지털 콘텐츠를 원하는 장소, 원하는 시간에 표출할 수 있어, 불특정 다수에게 광고하는 TV 매체 광고와는 다르게 고객 타겟팅 광고의 가능성으로 매우 주목받고 있다[1]. 그런데 보다 높은 광고 효과를 위해서는 외부 이벤트 및 센서 이벤트에 연동하는 인터랙티브 사이니지의 지원 요구가 증대되고 있으나, 현재 전개되고 있는 대부분의 인터랙티브 사이니지는 SI 성(性)으로, 해당 응용 및 해당 적용 현장에 맞게 하드코딩을 하여 고정된 방식으로 구현되고 있다. 따라서 또 다른 현장의 또 다른 응용 목적의 사이니지의 개발을 위해 다시 엔지니어가 재프로그래밍하여야 하는 불편 및 낭비가 존재한다.

인터랙션은 외부 이벤트 또는 센서 입력 이벤트가 발생했을 때 적절한 조치(액션)를 제공하는 것인데, 다양한 경우의 이벤트 발생과 이에 따른 해당 조치를 'if(등록 지정 센서/외부사건 이벤트 발생), then(상응하는 조치(액션; 함수)수행)'이라는 규칙으로 기술하고 이러한 규칙들이 수행하도록 지원하는 규칙

엔진[2]을 활용하면, 기존 인터랙티브 디지털 사이니지 시스템의 고정된 방식을 벗어나 사이니지 시스템이 설치된 장소 및 여러 고려 사항에 맞는 인터랙션을 지원할 수 있다.

본 논문에서는 윈도우즈 사이니지 플레이어 환경에서의 규칙엔진 기반 센서 인터랙티브 사이니지 서비스에 대한 이전의 본 연구실 논문에 이어, 안드로이드 사이니지 플레이어 환경에서의 더욱 유연하게 확장되고 개선된, 상황인지 지원 안드로이드 인터랙티브 사이니지 시스템 설계와 구현에 대한 개발 결과를 보고한다.

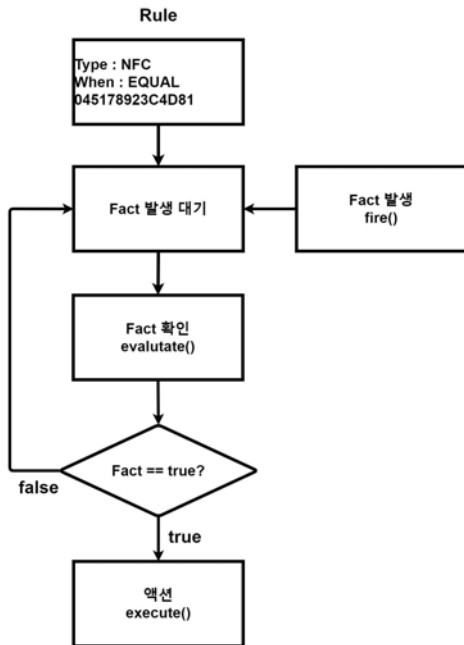
2. 기술적 배경; Easy Rules

사이니지 플레이어는 전송되어 온 다양한 광고 콘텐츠를 재생하는 S/W로 현재 많은 안드로이드 응용 프로그램으로 구현되고 있다.

Drools[3]와 같은 매우 강력한 규칙엔진과의 통합은 전체적으로 임베디드 환경에서의 사이니지 플레이어 S/W의 원활한 동작에 부담을 가중시킨다.

반면 Easy Rules[4]은 경량 자바 규칙엔진 라이브러리이다. Easy Rules에서는 'when'에 Fact 라는

객체를 이용하여 Fact 상태를 점검하는 것으로 조건 점검을 지원하며, 조건이 만족하는 경우에 then에 등록된 액션을 수행하게 된다. <그림 1>은 본 논문의 인터랙티브 사이니지 플레이어에 통합된 Easy Rules 규칙 엔진 이벤트 구동 동작을 보여준다.



<그림 1> Easy Rules 규칙엔진 기반 본 논문 인터랙티브 사이니지 이벤트 구동 동작 과정

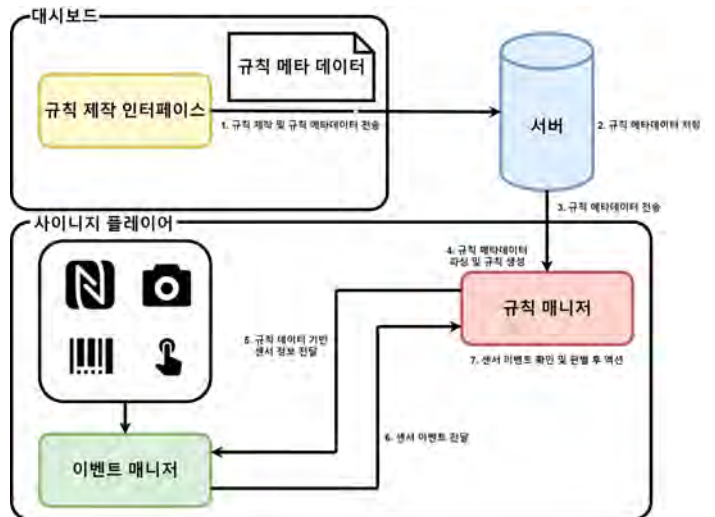
3. 제안 설계

3.1. 규칙 적용 동작과정

본 논문에서 제안하는 규칙엔진 기반 인터랙티브 시스템 동작 구조는 <그림 2>와 같다. 규칙 메타데이터를 제작할 수 있는 규칙 제작 인터페이스를 포함하며, 해당 규칙을 사이니지 플레이어로 전송하는 대시보드, 규칙 관련 메타 데이터를 저장하고 대시보드에서 선택한 사이니지 플레이어로 규칙 메타데이터를 전달해주는 서버, 받은 규칙 메타데이터를 이용하여 규칙 매니저에서 규칙을 생성하고, 이벤트 매니저에서 외부 이벤트 또는 센서 이벤트 연동을 지원한다. 사이니지 플레이어는 이벤트 연동 특정 콘텐츠를 표출한다.

이벤트 구동 동작 흐름은 다음과 같다. (1) 우선 규칙 제작 인터페이스에서 사용자가 원하는 상황 및 상황에 따른 액션을 설정하게 되면 대시보드는 이에 대한 규칙 메타 데이터를 생성하고 생성된 규칙을 사이니지 플레이어에 적용 시키기 위해 특정 플레이

어를 선택하여 해당 규칙을 전송한다. (2) 전송된 규칙 메타데이터는 바로 사이니지 플레이어로 가지 않고 서버를 거치게 되며, 이때 서버는 해당 규칙 메타데이터를 서버에 저장한 후 (3) 선택된 사이니지 플레이어로 전송한다. (4) 전송받은 규칙 메타데이터는 사이니지 플레이어 내부의 규칙 매니저에서 구문 분석한 후 Easy Rules 규칙 엔진의 규칙 적용에 쉬운 형태의 Fact 객체, 규칙 객체들을 생성하고 규칙 엔진이 작동할 수 있도록 구성한다. (5) 이후, 외부 이벤트/센서 이벤트가 발생하게 되면 해당 이벤트 매니저는 해당 이벤트에 연관된 Facts 개체의 상태 값을 변경한다. (6) 규칙엔진은 이벤트 Fact 상태 정보를 기반으로 조건이 부합하는지를 판별한 후 모든 조건이 만족 되면 규칙에 맞는 액션을 취한다.



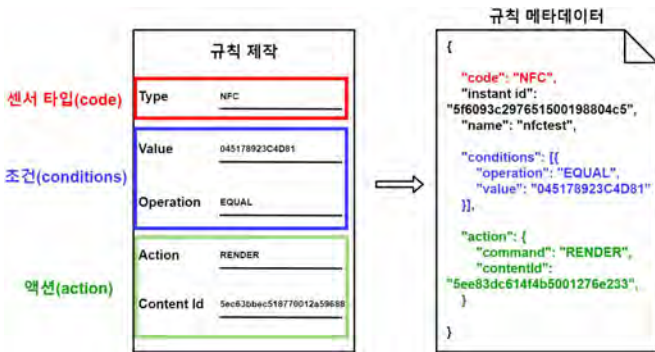
<그림 2> 인터랙티브 디지털 사이니지 규칙 적용 동작 구조

3.2. 규칙 제작 인터페이스 구현 및 메타데이터 생성

규칙 제작을 위해서는 사용자가 직접 규칙을 구성할 수 있는 인터페이스가 필요하며 규칙을 위하여 본 논문에서는 규칙 설정을 위한 UI를 대시보드 상에 구현하였다.

규칙 적용을 위해 생성되는 메타데이터의 핵심은 이벤트 타입(code), 조건(condition), 액션(action) 세 가지로 표현할 수 있다. <그림 3>은 본 논문에서 설계·구현한 대시보드 UI 상에서 NFC 센서 연동 규칙 구성 UI 및 생성되는 규칙 메타 데이터 형식의 예시이다. 이에 대한 설명은 다음과 같다. 사용자가 사용할 이벤트 타입(code)인 NFC를 설정한 후, 특정 NFC의 값이 들어 오면 기입한 Value와 읽어

진 NFC의 값이 같은지를 확인하는 조건(condition)을 기입한다. 마지막으로 해당 조건이 사용자가 이용할 센서 타입을 선택한 후, 액션을 하기 위해 부합하여야 하는 조건들을 기술한다. 이후 해당 조건들이 맞아떨어지게 되었을 때 사이니지 플레이어가 수행하여야 할 액션(action)을 지정하여 이를 기반으로 규칙 메타데이터를 제작한다.

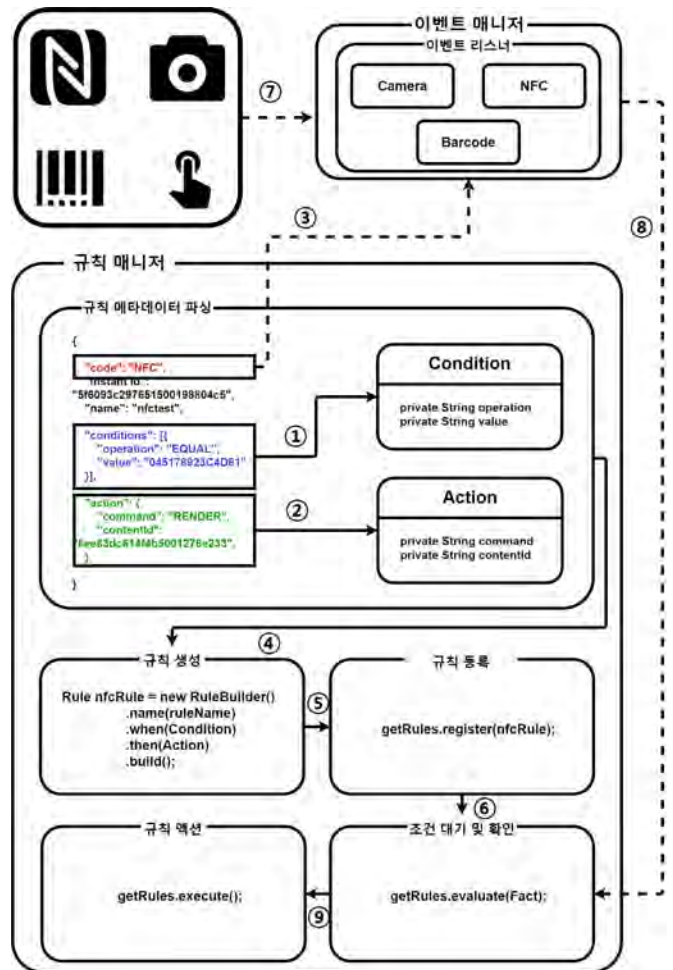


<그림 3> NFC 센서 규칙 구성 UI 인터페이스 및 메타 데이터

3.3. 규칙 적용 및 센서 연동

규칙 제작 인터페이스에서 특정 규칙을 제작하였으면 해당 규칙을 사용하기 위한 작업이 필요하다. 따라서 규칙 메타데이터를 파싱 및 생성할 수 있고, 규칙에 대한 조건이 부합하였을 경우, 액션에 대한 명령을 수행하는 규칙 매니저를 제작하였다. 또한, 해당 규칙에 대한 조건은 센서로부터 받은 값과 비교하여 확인하기 때문에 외부/센서 이벤트와 그에 대한 값을 받아들 수 있는 이벤트 매니저를 제작하였다. 규칙 적용 및 이벤트 연동과 관련된 과정은 NFC 센서 연동 규칙 적용의 예시로 <그림 4>와 같다. (1, 2) 우선 전달받은 규칙 메타데이터에 대해 사용하기 쉬운 형태의 객체로 파싱을 진행한다. 조건(Fact)에 대한 객체인 Condition과 Action에 대한 객체를 제작하고, (3) 사용할 센서에 대한 정보를 기반으로 이벤트 매니저에서 해당하는 센서의 이벤트 리스너를 생성한다. (4) 객체로 만들어진 Condition과 Action을 기반으로 규칙을 생성하며, 규칙 생성 시 조건에 해당하는 'when()'을 이용하여 조건을 기입하고, 액션에 해당하는 'then()'을 이용하여 해당 액션을 정의한다. (5) 규칙의 생성이 완료될 경우 'register()' 함수를 이용하여 규칙을 등록하며, 해당 함수를 통해 등록된 규칙은 백그라운드 상에서 동작하며 조건에 대한 이벤트의 값이 들어올 때 까지 (6) 대기상태가 된다. (7) 센서에서 특정 이벤트가 일어난 후 이벤트에 해당하는 값을 이벤트 매니저

내부의 이벤트 리스너로 가져온다. (8) 이벤트에 대한 값을 받은 후 이벤트 매니저는 해당 값을 대기 중인 규칙에 전송하게 되며, 규칙에서 조건으로 정한 값과 이벤트 매니저로부터 받은 이벤트에 대한 값이 올바른지 확인하는 작업을 진행하며 해당 작업은 'evaluate()' 함수에서 진행한다. (9) 'evaluate()' 함수 내부에서 해당 값을 비교한 후 조건이 부합할 경우 해당 true, false 값을 'execute()' 함수로 넘겨주게 되며 사용자가 지정한 액션을 하게 된다. 예시에서는 특정 프레젠테이션을 호출하는 액션을 정의하였으며 해당 프레젠테이션에 대한 데이터는 Intent[5]를 통해 전달 후 호출하였다.



<그림 4> 규칙엔진 및 이벤트 연동 구조 (NFC 센서 이벤트 경우)

4. 구현 결과

해당 실험은 Android 9 및 24인치 디스플레이를 이용하여 진행되었다. 결과는 <그림 5>와 같으며, 대시보드에서 규칙 제작이 가능한 인터페이스를 구현하였으며, 해당 인터페이스를 이용해 특정 센서에 대한 여러 가지 상황의 규칙을 제작할 수 있다. 또

한, 해당 규칙을 디지털 사인지에 적용하여 사용자가 정의한 규칙에 따라 조건이 부합될 경우 특정 액션을 할 수 있다.



(규칙 제작 인터페이스)



(사인지 플레이어)

<그림 5> 규칙 적용 인터랙티브 디지털 사인지 동작 결과

5. 결론 및 향후 연구

본 논문에서는 기존 인터랙티브 디지털 사인지에서 여러 가지 상황에 맞게 다채로운 정보를 제공하지 못하는 문제점을 해결하기 위해 사용자가 규칙을 구성할 수 있는 규칙 제작 인터페이스를 제공하여 여러 상황에 따라 다양하게 반응할 수 있는 규칙을 제작할 수 있게 하였다. 또한 여러 상황에 반응을 하기 위한 도구로 센서를 이용하여 만들어진 규칙을 센서에 대입해 구성한 규칙에 대한 행동을 수행할 수 있는 인터랙티브 디지털 사인지 플레이어를 구현하여 사용자의 흥미 유도 및 송출 콘텐츠에 대한 구성을 다양하게 만들었다.

향후 연구에서는 본 논문의 내용을 포함하며, 여러 개의 규칙을 전부 적용할 수 있는 다중 규칙 적용과 관련된 연구를 할 것이며, 이를 통해 더욱 유연하게 제작할 수 있는 인터랙티브 사인지를 제작할 것이다. 또한, 빅데이터 분석 결과 이벤트를 연결하여 사용자에게 맞는 상품을 추천하여 좀 더 다양한 정보를 제공할 수 있게 제작할 것이다.

참고문헌

[1] 김성원, “2020 디지털 사인지 전망과 기대,” 2020. 3.
 [2] M. Fowler, “Should I use a Rules Engine?,” <https://martinfowler.com/bliki/RulesEngine.html>
 [3] Drools, <https://www.drools.org/>
 [4] EasyRules, <https://github.com/j-easy/easy-rules>
 [5] Android Developers, Intent, <https://developer.android.com/reference/android/content/Intent>