

# 상황 인지 기반의 자동 태스크 설정을 위한 안드로이드 어플리케이션 구현

## Implementation of Context-Aware Android Application for Automatic Task Setting

박나연\*, 정다해\*, 창병모\*, 최광훈\*\*

숙명여자대학교 컴퓨터과학부\*, 전남대학교 전자컴퓨터공학부\*\*

Na-Yeon Bak(b\_newyork@naver.com)\*, Da-hae Chung(dahaech@gmail.com)\*  
Byeong-Mo Chang(chang@sm.ac.kr)\*, Kwanghoon Choi(kwanghoon@jnu.ac.kr)\*\*

### 요약

안드로이드 스마트폰 사용자는 수동적으로 특정 태스크를 수행하도록 설정할 수 있으나 이러한 수동적 태스크의 설정은 불편할 뿐만 아니라 사용자의 현재 상황을 고려하지 않기 때문에 사용자가 원하지 않는 불필요한 서비스를 제공하여 사용자를 곤란하게 할 수 있다. 본 연구에서는 이러한 단점을 보완하는 상황-인지 기반의 자동 태스크 설정 어플리케이션을 설계, 구현하였다. 이 어플리케이션을 사용하면 사용자는 자신이 원하는 상황에 원하는 태스크만을 수행하도록 설정하여 상황에 맞는 서비스를 제공 받을 수 있다. 상황에 따라 원하는 태스크를 수행하도록 기술하기 위해 정형화된 구문을 설계 하였으며 안전한 태스크 설정을 위해 구문의 유효성을 자동으로 검사하는 유효성 검사기를 구현하였다.

■ 중심어 : | 상황 | 행동 | 상황-행동 규칙 | 규칙 검사 | 태스크 | 안드로이드 |

### Abstract

Users can set manually Android phone to do particular tasks on some situations. This type of setting is inconvenient, and also provide with unnecessary services, which don't consider users' situation. In this research, we design and implement a context-aware automatic task setting application. Users can get context-aware service by setting desired tasks based on contexts using this system. We design a language for describing context-action rules, and statically check validity of context-action rules by performing syntax and semantic check.

■ keyword : | Context | Action | Context-Action Rule | Rule Checking | Task | Android |

## 1. 서론

스마트폰은 전화, 문자, SNS를 통해 의사소통할 수 있는 기능뿐만 아니라 컴퓨터와 같이 인터넷을 통해 다양한 정보와 서비스를 제공한다. 사용자는 자신이 원하

는 상황에서 원하는 서비스와 정보를 제공받기를 원하며 이를 위해 수동적으로 특정 태스크를 수행하도록 설정할 수 있다. 그러나 이러한 수동적 태스크의 설정은 불편할 뿐만 아니라 사용자의 현재 상황을 고려하지 않기 때문에 사용자가 원하지 않는 상황에서 불필요한 서

\* 이 논문은 2014년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. NRF-2014R1A1A2053446)

접수일자 : 2016년 08월 03일

수정일자 : 2016년 11월 10일

심사완료일 : 2016년 11월 10일

교신저자 : 창병모, e-mail : chang@sm.ac.kr

비스를 제공하여 사용자를 곤란하게 할 수 있다.

이러한 문제점을 해결하기 위해 태스크 자동 설정을 위해 Tasker, AutomateIt 등의 어플리케이션이 개발되었다[1-4]. 이들은 사용자가 설정한 대로 해당 상황에서 원하는 태스크를 수행하지만 Tasker는 사용이 어렵다는 단점을 갖고 있으며[1][2] AutomateIt은 기능 혹은 UI의 구현이 부족하다는 단점을 갖고 있다[3][4].

본 연구에서는 이러한 단점을 보완하기 위해 상황-인지를 기반으로 하여 사용자가 설정한 상황-행동 규칙에 따라 원하는 태스크를 자동으로 수행하는 안드로이드 어플리케이션을 설계, 구현하였다. 이 어플리케이션 사용자는 사용자 맞춤형 태스크 수행 계획을 상황-행동 규칙(context-action rule) 형태로 작성하여 자신이 원하는 상황에 원하는 태스크만을 수행하도록 설정하여 상황에 맞는 서비스를 제공 받을 수 있다. 예를 들어 사용자는 다음과 같은 상황에서 아래와 같은 서비스를 받도록 설정할 수 있다.

- 집에 도착하면 WiFi를 켜다.
- 수업시간(특정 시간이 되었을 때)에 학교(특정 장소에 위치할 때)에 있으면 무음모드로 변경된다.
- 운전 중 문자가 오면 문자 내용을 읽어준다.

본 연구에서는 사용자가 직관적인 단계에 따라 최소화된 터치로 자신이 원하는 태스크 수행을 설정할 수 있도록 한다. 또한 사용자가 작성한 태스크 수행 계획에 문법적, 의미적 오류가 있는지 규칙검사를 통해 미리 확인한다. 이 검사는 크게 2가지로 구문 검사와 의미 검사를 통해 이루어지며 오류 발생 가능성이 있을시 오류 해결에 대한 도움말 서비스를 제공한다. 이는 사용자가 적합한 구문을 작성할 수 있도록 지원하고 안정적인 어플리케이션 사용을 돕는 것이다.

본 논문의 구성은 다음과 같다. 2절에서는 이 논문의 배경과 관련연구에 대해서 기술한다. 3절에서는 상황에 따라 원하는 태스크를 기술할 수 있는 상황-행동 규칙에 대해서 기술한다. 4절에서는 시스템의 구조와 시스템 구현에 대해 자세히 기술한다. 5절에서는 구현 결과에 대해서 기술하고 6절에서 결론을 맺는다.

## II. 관련 연구

상황-인지 서비스(context-aware service)는 사용자의 현재 상황을 인식하고 상황에 따라 필요로 하는 서비스를 자동적으로 제공하는 것이다. 상황 정보는 사용자가 상호 작용을 하는 시점에 이용할 수 있는 모든 정보로서 사람, 객체의 위치, 식별, 활동, 상태 등을 포함한다. 이러한 상황 정보의 수집 및 처리 과정을 거쳐, 사용자가 필요로 하는 적절한 서비스를 제공하는 것이 상황-인지 서비스의 목적이다. 이러한 서비스는 의료, 교육, 재난, 구호, 쇼핑 등 사회 전 분야에 걸쳐 사용될 수 있다[5-10].

그 중에서도 스마트폰과 같은 모바일 기기는 개인이 사용하는 특징을 가지고 있다. 이러한 특징은 사용자 정보를 보다 쉽게 구할 수 있으며 이를 이용한 개인화된 상황-인지 서비스를 적용하기 적합한 환경을 제공한다. 특히 안드로이드 폰은 넓은 범위의 앱 권한 승인을 어플리케이션에게 허용함으로써 데이터네트워크, 와이파이, 블루투스 등의 설정, 전화 혹은 SNS 수신, GPS 위치 수신, 케이블 혹은 이어폰 연결 등과 같은 다양한 상황을 처리할 수 있다[6-13].

상황-인지 기반의 자동 태스크 수행을 제공해주는 어플리케이션으로는 Tasker, AutomateIt 등이 있다. Tasker의 경우, 매우 풍부한 기능을 제공하지만 직관적이지 않은 설정 등록 방법과 키 조작법으로 사용자의 혼란을 야기할 뿐만 아니라 논리 및 산술 연산자와 함께 구체적 수치를 이용해 세부사항을 설정해야 하는 등 일반 사용자가 사용하기에 어려움이 크다[1][2].

AutomateIt의 경우, Tasker에 비해 직관적이고 명료한 UI/UX 디자인으로 사용자의 접근이 용이하며 순차적인 명령문의 작성이 가능하지만 복잡조건 설정을 위해서는 여러 화면을 오고 가면서 보조적인 몇 개의 키를 반복적으로 눌러야 하고 복잡한 조건을 확인할시 여러 개의 조건 설정을 한눈에 볼 수 없다[3][4].

또한 지금까지의 자동 설정에 관한 연구들은 설정된 플랜(혹은 규칙)의 유효성을 효과적으로 검사하는 과정이 부족하여 설정된 플랜이 정상적으로 작동할 수 있는지 미리 확인하기 힘들었다.

본 연구에서는 사용자가 원하는 상황에 원하는 태스크 수행을 쉽게 기술할 수 있도록 규칙 형태의 정형화된 언어를 제시하였으며 안전한 태스크 설정을 위해 작성된 규칙의 유효성을 자동으로 검사하는 유효성 검사를 설계하고 구현하였다.

### III. 상황-행동 규칙

본 시스템의 주요 기능은 사용자가 계획한 상황에 따라 원하는 행동 즉, 태스크를 수행하도록 미리 설정하는 것이다. 이를 위해서 사용자의 태스크 수행 계획을 효과적으로 기술할 수 있어야 하며 본 연구에서는 이를 상황-행동 규칙(context-action rule) 형태로 정의한다. 이 규칙은 상황에 따라 수행할 행동을 기술하기 위한 것이다. 상황-행동 규칙 R은 “상황 ⇒ 행동”과 같은 형식을 가지며, 그 의미는 상황이 만족될 때, 해당 행동을 수행하라는 것이다.

- 상황-행동 규칙

$$R \rightarrow \text{상황}(C) \Rightarrow \text{행동}(A)$$

- 상황(Context)

$$C \rightarrow CK$$

$$| \text{AND } C \text{ } C \text{ DONE}$$

$$| \text{OR } C \text{ } C \text{ DONE}$$

$$CK \in \text{Context Keyword List}$$

- 행동(Action)

$$A \rightarrow AK^+$$

$$AK \in \text{Action Keyword List}$$

상황은 행동이 실행되는 조건으로 시스템이 인지하는 상황이 사용자가 기술해 놓은 상황과 일치할 때, 해당 행동이 수행된다. 상황은 [표 1]에 있는 한 개의 상황 키워드만으로 단순하게 구성하거나 AND, OR와 같은 상황연산자와 상황키워드들로 보다 복잡하게 구성할 수 있다. DONE은 AND 혹은 OR의 끝을 나타낸다. 상황키워드(CK)는 [표 1]과 같이 사용자의 스마트폰이 처해있는 시스템 상태, 시간, 위치정보 등의 상황을 나타낸다. 3개 이상의 상황도 다음과 같이 이항 연산을 사용하여 표현할 수 있다. 따라서 이 논문에서는 이항 연산

을 중심으로 상황을 설명한다.

$$\text{AND CK CK CK DONE}$$

$$= \text{AND AND CK CK DONE CK DONE}$$

$$\text{OR CK CK CK DONE}$$

$$= \text{OR OR CK CK DONE CK DONE}$$

표 1. 상황 키워드 목록

상황 키워드(CK)	
WiFi 켜짐	WiFi 꺼짐
데이터네트워크 켜짐	데이터네트워크 꺼짐
블루투스 켜짐	블루투스 꺼짐
비행기모드 켜짐	비행기모드 꺼짐
화면 켜짐	화면 꺼짐
소리모드일 때	SMS 수신 시
진동모드일 때	전화 수신 시
무음모드일 때	통화 종료 시
충전기 연결 시	충전기 해제 시
이어폰 연결 시	이어폰 해제 시
배터리 n% 이상일 때	배터리 n% 이하일 때
위아래 흔들기	양쪽 흔들기
특정 시간이 되었을 때	특정 장소에 위치할 때

행동은 사용자가 시스템으로 하여금 수행하길 바라는 태스크를 의미한다. 시스템이 인지하는 상황이 사용자가 기술해 놓은 상황과 일치할 때 행동이 수행될 것이다. 행동은 1개 이상의 행동 키워드(AK)로 구성된다. 행동키워드로는 [표 2]와 같이 본 어플리케이션에서 수행 가능한 행동을 나타낸다.

표 2. 행동 키워드 목록

행동키워드(AK)	
WiFi 켜기	WiFi 끄기
데이터네트워크 켜기	데이터네트워크 끄기
블루투스 켜기	블루투스 끄기
음악 볼륨 변경	벨 볼륨 변경
소리모드로 전환	진동모드로 전환
수신번호 읽어주기	무음모드로 전환
SMS 보내기	전화 걸기
원하는 텍스트 읽어주기	문자 읽어주기
어플리케이션 실행(즐거찾기)	홈 화면으로 가기
알림(Notification) 띄우기	플래시 켜기
카메라	녹음
상황-행동 규칙 활성화	상황-행동 규칙 비활성화

예를 들어 사용자가 문자 메시지가 도착했을 때 수신번호를 자동으로 읽어주는 서비스를 원한다면 다음과 같이 상황으로 ‘SMS 수신 시’ 키워드를 선택하고 행동으로 ‘수신번호 읽어주기’ 키워드를 선택해서 상황-행동 규칙을 작성하면 된다.

'SMS 수신시' ⇒ '수신번호 읽어주기'

또 다른 예로 사용자가 화면을 켜올 때 WiFi가 꺼져 있다면 자동적으로 WiFi를 켜도록 설정하는 상황-행동 규칙은 다음과 같이 작성할 수 있다.

AND '화면켜짐' WiFi꺼짐 DONE  
⇒ 'WiFi켜기'

이러한 상황-행동 규칙의 작성은 상황, 행동 작성 페이지의 상황연산자 AND, OR, DONE 버튼들과 리스트에 있는 상황 키워드, 행동 키워드들을 선택하면서 순차적으로 진행되며 마지막 페이지에서 상황-행동 규칙의 이름을 작성한 후 등록한다. 자세한 사항은 5절의 구현 결과 부분을 참고하기 바란다.

#### IV. 시스템 구현

##### 1. 시스템 주요 구조

본 시스템의 주요 구조는 [그림 1]과 같이 크게 상황 파악 모듈, 상황-행동 규칙 모듈, 행동 실행 모듈로 구성된다. 이 모듈은 사용자가 작성한 상황-행동 규칙을 기반으로 하여 상황에 따라 적절한 행동을 실행하는 데 사용된다. 상황 파악 모듈은 기기의 현재 상황을 파악하여 이를 상황-행동 규칙 모듈로 보낸다. 상황-행동 규칙 모듈은 상황이 충족될 수 있는 상황-행동 규칙을 데이터베이스에서 찾아 상황-행동 규칙과 기기의 현재 상황을 대조 평가하여 해당 상황에서 해야 할 행동을 파악하여 이를 행동 실행 모듈로 알려 실행하도록 한다.

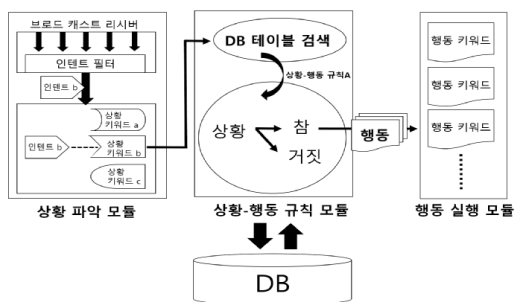


그림 1. 시스템 주요 구조

본 시스템은 다음과 같은 환경에서 구현하였다.

- 개발환경 : 안드로이드 스튜디오
- 운영체제 : 안드로이드 5.1.1 롤리팝
- SDK : 22 버전
- 개발 언어 : Android Java
- DB : SQLite

본 시스템의 각 모듈은 안드로이드 서비스 형태로 구현된다. 상황을 알리기 위해 브로드캐스트되는 인텐트(Intent)들은 상황 파악 모듈에 도착하기 전, 인텐트 필터를 거치는데 이는 수많은 인텐트들 중 본 시스템에서 필요한 인텐트만을 수신하기 위함이다[9].

상황 파악 모듈은 각 인텐트에 해당하는 상황 키워드를 정의하였으며 정의되어 있는 인텐트가 수신되면, 해당하는 상황 키워드를 인텐트를 통해 상황-행동 규칙 모듈로 보내준다. 이에 상황-행동 규칙 모듈은 상황 파악 모듈로부터 받은 상황 키워드로 작성된 상황-행동 규칙이 데이터베이스 테이블에 저장되어 있는지 확인한다. 매칭되는 상황-행동 규칙이 존재하면 상황-행동 규칙의 상황을 기기의 상황과 대조하여 평가한 후 결과 값이 '참'이면 상황-행동 규칙에 작성되어있는 행동을 읽어 인텐트를 통해 행동 실행 모듈로 보내준다. 행동 실행 모듈은 상황-행동 규칙 모듈로부터 행동을 받아, 작성 돼 있는 행동키워드(들을)를 순차적으로 수행시킨다.

##### 2. 상황-행동 규칙의 처리 방법

상황-행동 규칙을 실행하는데 있어서 시스템은 상황 결과 값에 따라 행동 수행 여부를 결정한다. 상황은 트리 형태로 나타낼 수 있으며 루트 노드의 상황 값이 이 상황의 결과 값이다. 부모노드의 상황 값은 자식노드들의 상황 값을 이용하여 결정한다. 그림 2는 상황 키워드 하나로 구성된 단순한 상황-행동 규칙을 나타내고 [그림 3]은 상황연산자를 사용한 복잡한 상황을 트리 형태로 표현한 예이다.



그림 2. 단순한 상황-행동 규칙(Simple Plan)

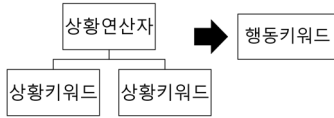


그림 3. 복잡한 상황-행동 규칙(Complex Plan) 예시

상황은 트리 형태로 표현할 수 있으므로 그 결과 값 계산은 깊이 우선 트리 탐색(depth-first search) 형태로 다음과 같이 각 노드를 처리한다.

- 상황키워드: 시스템의 현재 상황이 상황키워드의 조건 혹은 상태와 부합한지를 체크하여 ‘참’ 혹은 ‘거짓’인 상황 값을 계산한다.
- 상황연산자: 지식 노드의 상황 값들을 AND 혹은 OR 연산자에 따라 연산하여 부모노드의 상황 값을 계산한다.

사용자가 만든 상황-행동 규칙은 한 개의 데이터베이스 테이블에 저장되며 이는 어플리케이션이 어떤 상황에서 어떠한 행동을 할지 결정한다. [표 3]의 테이블은 4가지 필드(index, keyword, level\_id, keyword\_info)로 구성된다. keyword 필드는 사용자가 작성한 상황-행동 규칙의 상황 부분과 행동 부분의 키워드와 연산자들이 저장된다. ‘END’는 상황 결과 값으로 행동의 실행 여부를 결정한다. level\_id 필드는 keyword 필드의 ‘레벨 값’을 저장한다. ‘레벨 값’이란 상황을 트리로 표현할 때 keyword의 깊이를 나타내는 값이다. keyword\_info 필드는 특정 상황 키워드의 세부 조건 부합 여부와 특정 행동 키워드의 태스크 수행을 위한 정보를 저장한다.

표 3. 상황-행동 규칙의 테이블 저장형태

index	keyword	level_id	keyword_info
1	상황연산자	0	
2	상황키워드1	1	상황정보
3	상황키워드2	1	상황정보
4	DONE	1	
5	END	0	
6	행동키워드1		행동정보
7	행동키워드2		행동정보

### 3. 상황-행동 규칙의 유효성 검사

규칙의 유효성 검사는 새로 작성된 상황-행동 규칙이 정상적으로 실행 가능한지 그 유효성을 미리 확인하는 검사이다. 정상적인 실행이 가능하지 않는 상황-행동 규칙이 등록된다면 그 규칙은 어플리케이션의 강제 종료 혹은 사용자가 예상하지 못한 작업을 수행할 수 있기 때문이다. 규칙 검사는 크게 2가지로 하나는 구문 검사이며 다른 하나는 의미 검사이다.

상황-행동 규칙의 구문 검사(syntax check)는 상황-행동 규칙의 상황과 행동이 문법 형식에 맞게 만들어졌는지를 검사한다. 의미 검사(semantic check)는 상황-행동 규칙의 의미를 검사하여 사용자가 예상한 정상적 실행이 가능한지 검사한다. 새로 등록된 상황-행동 규칙이 문법적으로 문제가 없으나 실제로는 절대 실행되지 않거나 반복적으로 실행되는 등 비정상적인 수행을 일으킬 수 있다. 이 의미 검사는 3가지 검사를 중심으로 진행되는데 상황 키워드 사이의 유효성 검사, 상황과 행동 사이의 유효성 검사, 상황-행동 규칙 사이의 유효성 검사이다. 모든 검사를 통과하면 사용자가 만든 상황-행동 규칙은 정상 수행 가능하므로 시스템의 데이터베이스에 저장된다.

#### (1) 상황-행동 규칙의 구문 검사

구문 검사는 사용자가 만든 상황-행동 규칙이 문법적으로 올바른지 확인하는 검사이다. 상황 부분은 종류에 따라 검사기준이 달라지는데 단순한 상황의 경우, 한 개의 상황키워드만으로 작성되어야 하며 복잡한 상황의 경우, 상황연산자 AND 혹은 OR가 문법에 맞게 작성되었는지 검사한다. 행동은 종류에 상관없이 행동 키워드 개수가 1개 이상이어야 한다.

또한 모든 상황과 행동은 특정 키워드의 정보유무를 검사해야 한다. [표 4]에 작성된 키워드의 경우, 키워드가 요구하는 정보가 사전에 저장되어 있어야만 정상수행이 가능하다. ‘음악 볼륨 변경’ 키워드의 경우, 설정될 볼륨 정보가 필요한데 그 정보가 저장되어 있지 않으면 해당 키워드를 수행할 수 없다.

표 4. 특정정보를 필요로 하는 키워드

키워드 명		키워드의 특정 정보
전화 수신시	SMS 수신시	전화번호
배터리 n% 이상일 때	배터리 n% 이하일 때	배터리 용량
특정 장소에 위치할 때		설정 위치
특정 시간이 되었을 때		설정 시간
전화 걸기		전화번호
SMS 보내기		전화번호, 문자내용
원하는 텍스트 읽어주기		텍스트 내용
벨 볼륨 변경		벨 볼륨 용량
음악 볼륨 변경		음악 볼륨 용량
어플리케이션 실행(줄여찾기)		어플리케이션
알림(Notification) 띄우기		알림 내용
상황-행동 규칙 활성화	상황-행동 규칙 비활성화	상황-행동규칙

(2) 상황 키워드 사이의 유효성 검사

상황 키워드 사이의 유효성 검사는 상황을 구성하는 키워드 간 발생할 수 있는 오류를 사전에 방지하기 위한 검사로 잘못된 상황 키워드의 조합은 상황의 결과값을 항상 거짓이나 참으로 만들어 전혀 실행되지 않게 하거나 비정상적인 실행을 유발할 수 있다.

표 5. 키워드 타입표

키워드		타입
WiFi 꺼짐	WiFi 꺼짐	WiFi
화면 꺼짐	화면 꺼짐	Screen
소리 모드	진동 모드	무음 모드
데이터네트워크 켜짐	데이터네트워크 꺼짐	Data
블루투스 켜짐	블루투스 꺼짐	Blue
비행기모드 켜짐	비행기모드 꺼짐	AirMode
SMS 수신 시		SMS
전화 수신 시	통화 종료 시	Call
충전기 연결 시	충전기 해제 시	Power
이어폰 연결 시	이어폰 해제 시	Earphone
배터리 이상	배터리 이하	Battery
양쪽 흔들기	위-아래 흔들기	Shake
폰 뒤집기		Upside
특정 장소에 위치할 때		Location
특정 시간이 되었을 때		Time
수신 번호 읽어주기	문자 읽어주기	Read

상황 키워드 사이의 유효성 검사는 [표 5]에 정의된 각 상황 키워드의 타입을 서로 비교한다. AND 혹은 OR 연산자로 작성된 상황 키워드의 타입이 같다면 연산 결과가 항상 거짓이거나 항상 참일 수 있으므로 상황 키워드 타입은 서로 달라야 한다. 예를 들어 AND 'WiFi 꺼짐' 'WiFi 꺼짐' DONE과 같은 상황은 동시에 성립할 수 없다. 이는 다음과 같은 논리 규칙으로

정리할 수 있는데  $K1:T1$ 은 상황 키워드 K1의 타입이 T1임을 나타내고  $\wedge$ 는 논리곱  $\vee$ 는 논리합을 나타낸다.

- $AND K1 K2 DONE$  혹은  $OR K1 K2 DONE$  경우  
If  $K1: T1 \wedge K2: T2$ , then  $T1 \neq T2$

(3) 상황과 행동 사이의 유효성 검사

상황-행동 규칙의 정상적인 실행을 위해 상황 부분과 행동 부분 간의 함께 작성될 수 없는 키워드 조합 혹은 함께 작성되어야만 하는 키워드 조합이 있다. 이러한 조합들이 지켜지지 않을 경우 상황-행동 규칙의 비정상적인 실행을 유발할 수 있으며 이러한 오류를 사전에 방지하기 위해 상황-행동 규칙에 대해서 아래의 2가지 검사를 실시한다.

- 양립 불가 판별 검사

이 검사는 함께 성립될 수 없는 상황키워드와 행동키워드가 함께 사용되고 있는지 검사한다. 아래 표 6은 상황키워드 '화면 꺼짐', '비행기 모드 꺼짐'과 함께 성립될 수 없는 행동키워드들의 목록을 보여주고 있다. 예를 들어 '화면 꺼짐' 상황에서 '카메라' 행동은 성립될 수 없다. 또한 '비행기모드 꺼짐' 상황에서 'WiFi 켜기' 행동은 성립될 수 없다.

표 6. 양립 불가 상황, 행동 키워드 조합표

상황 키워드	행동 키워드
화면 꺼짐	카메라
	어플리케이션 실행(줄여찾기)
	홈 화면으로 가기
비행기모드 꺼짐	WiFi 켜기
	데이터네트워크 켜기
	블루투스 켜기
	전화하기
	메시지 보내기

- 의존성 확인 검사

본 검사는 함께 존재되어야만 하는 키워드들의 작성 여부를 확인하는 검사이다. 특정 행동키워드는 상황키워드로부터 수행에 필요한 정보를 제공받아야 하며 그렇지 않으면 해당 행동키워드는 수행이 불가능하다. 예를 들어 Read 타입의 행동키워드로 작성된 행동은 반

드시 SMS 혹은 Call 타입인 상황키워드와 함께 정의되어야 한다. Read 타입의 행동키워드는 ‘수신번호 읽어주기’, ‘문자 읽어주기’이며 이들은 수신번호와 문자 내용을 제공해 줄 수 있는 SMS 혹은 Call 타입의 상황키워드인 ‘SMS 수신 시’ 혹은 ‘전화 수신 시’와 함께 정의되어야 한다.

- $CK \Rightarrow AK$   
 If  $AK: AT = Read \wedge CK: CT,$   
 then  $CT \in \{SMS, CALL\}$

AND 연산자로 상황이 작성된 경우에는 하나 이상의 상황 키워드 타입이 SMS 혹은 CALL이어야 하며 OR 연산자의 경우에는 상황의 키워드 타입은 모두 SMS 혹은 CALL이어야 한다.

- $(AND CK1 CK2 DONE) \Rightarrow AK$   
 If  $AK: T = Read \wedge CK1: T1 \wedge CK2: T2,$  then  
 $SMS \in T1 \cup T2 \vee CALL \in T1 \cup T2$
- $(OR CK1 CK2 DONE) \Rightarrow AK$   
 If  $AK: T = Read \wedge CK1: T1 \wedge CK2: T2,$  then  
 $T1 \cup T2 \subseteq \{SMS, CALL\}$

(4) 상황-행동 규칙 사이의 유효성 검사

상황-행동 규칙 사이의 유효성 검사는 상황-행동 규칙들 간에 발생할 수 있는 오류들을 사전에 방지하는 검사로써 무한히 반복될 수 있는 태스크 수행을 방지하는 검사이다. 이러한 오류를 사전에 방지하기 위해 새로운 상황-행동 규칙이 등록될 때마다 상황-행동 규칙 사이에 루프가 형성되는지 확인한다.

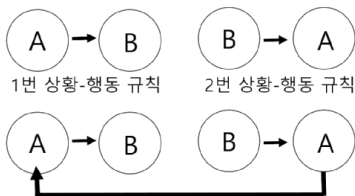


그림 4. 상황-행동 규칙의 루프생성

[그림 4]와 같이  $A \Rightarrow B$  규칙과  $B \Rightarrow A$  규칙이 함께 있으면 무한 루프가 발생할 수 있다. 첫 번째 상황-행동 규칙의 상황 A가 참이 되어 행동 B가 실행되면 두 번째 규칙의 상황 B가 참이 되고 행동 A가 수행됨으로써 루프가 생성되고 결국엔 무한 루프가 발생한다. 예를 들어 상황-행동 규칙 ‘소리모드 일 때’  $\Rightarrow$  ‘진동모드로 전환’이 사전에 저장돼 있는 상태에서 새로운 상황-행동 규칙 ‘진동모드 일 때’  $\Rightarrow$  ‘소리모드로 전환’이 작성된다면 ‘소리모드 일 때’ 상황이 참이면 ‘진동모드로 전환’ 행동이 수행된다. 그 영향으로 ‘진동모드 일 때’ 상황이 참이 되어 다시 ‘소리모드로 전환’ 행동이 수행됨으로써 무한루프가 발생한다.

V. 구현결과

1. 주요 기능 구현 결과

어플리케이션을 최초 실행되었을 때 그림 5와 같은 계획 추천 페이지를 볼 수 있다. 이 페이지에서는 사용자의 기호를 조사하여 그에 알맞은 상황-행동 규칙을 추천한다. 이 서비스는 한번만 수행되며 자동으로 저장된 상황-행동 규칙을 통해 더욱 쉽게 상황-행동 규칙을 작성하고 이용할 수 있다.



그림 5. 계획 추천 페이지

본 시스템에서 새로운 상황-행동 규칙을 만들기 위해선 [그림 6]의 메인 메뉴에서 ‘새 부탁하기’를 선택한 후 상황과 행동을 차례대로 작성해야 한다. 그 후 시스템에서 규칙검사를 실시한 다음 상황-행동 규칙을 저장한다. 상황-행동 규칙의 등록은 상황 작성, 행동 작

성, 상황-행동 규칙의 이름 작성으로 진행된다. 단순한 상황은 리스트에 있는 상황키워드들 중에서 한 개의 키워드를 선택한 후 완료 버튼을 누르면 된다. 복잡한 상황은 연산자 AND, OR, DONE들과 리스트에 있는 상황키워드들로 사용자가 원하는 상황을 만든 후 상황 완료 버튼을 누르면 된다. 행동은 특별한 연산자 없이 리스트에 있는 행동키워드들을 원하는 만큼 선택한 후 행동완료 버튼을 누르면 된다. 마지막으로 규칙의 이름을 작성한 후 저장 버튼을 눌러 상황-행동 규칙의 등록한다.



그림 6. 메인 메뉴

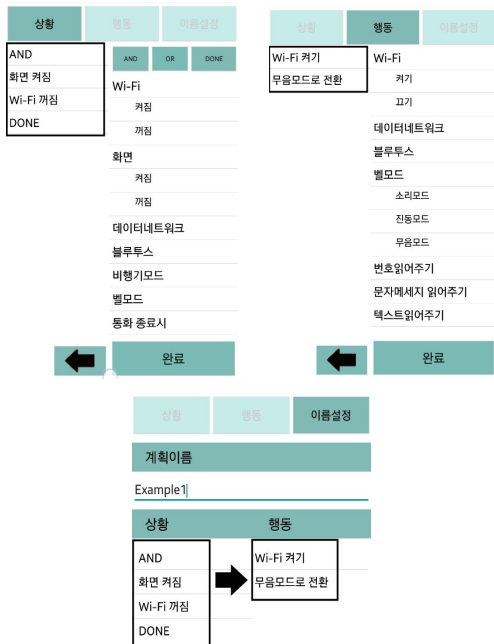


그림 7. 상황-행동 규칙 등록과정

[그림 7]은 '스마트폰의 화면이 켜지고 그때 WiFi가 꺼져 있다면 WiFi를 켜고 무음모드를 설정하겠다.'라는 상황-행동 규칙을 작성하는 예이다. 상황 작성에서는 AND 연산자와 '화면 켜짐'과 'WiFi 꺼짐' 상황키워드를 선택하여 복잡한 상황을 작성하였고 행동 작성에서는 'WiFi 켜기'와 '무음모드로 전환' 행동키워드로 행동을 작성하였다. 그리고 상황-행동 규칙의 이름을 'Example1'로 작성하였으며 이로써 상황-행동 규칙의 작성은 완료되고 규칙검사가 실시된다.

[그림 8]에서는 위의 예에서 작성한 상황-행동 규칙의 실행 결과를 시현한 것으로 실행전 모습과 실행후 WiFi가 켜지고 무음모드가 설정된 것을 보여준다.



그림 8. 실행 전-후

## 2. 규칙검사 오류처리

본 시스템은 상황-행동 규칙이 규칙검사에서 실패하였을 때 사용자가 더욱 쉽게 상황-행동 규칙의 오류를 수정할 수 있는 도움말 서비스를 제공한다. 도움말 서비스는 해당 상황-행동 규칙을 분석하여 규칙검사가 실패한 원인을 찾아내고 그에 대한 해결방안을 제공하는 서비스이다.

아래 [그림 9]에서는 작성한 상황-행동 규칙의 상황이 규칙검사에서 실패하여 나타나는 오류 메시지를 보여 주고 있다. 오류 메시지는 상황과 행동의 작성이 끝난 후 실시된 규칙검사가 실패하였다면 팝업 메시지로 나타나며 오류메세지의 도움말 버튼을 클릭하면 도움말 서비스가 제공되는 페이지로 이동한다. [그림 10]의 도움말 서비스를 보면 오류가 있는 상황-행동 규칙에 대한 해결책을 제공하고 있다.

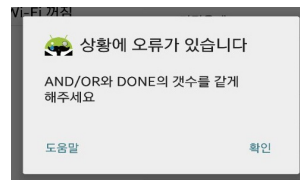


그림 9. 오류 메시지



도움말 서비스



해결책  
AND/OR와 DONE의 갯수를 같게 해주세요  
행동키워드를 한개이상 선택해주세요

그림 10. 도움말 서비스

VI. 결론

본 연구는 상황-인지를 기반으로 하여 사용자 맞춤형 태스크 수행 계획을 상황-행동 규칙 형태로 작성하고 이를 바탕으로 상황에 따라 원하는 행동을 자동으로 수행하는 안드로이드 어플리케이션을 구현했다는 점에서 그 의미가 있다. 또한 사용자가 상황-행동 규칙 형태로 작성한 수행 계획의 유효성을 미리 검사하여 규칙의 오류들을 사전에 예방하도록 하였다.

본 연구에서 구현한 태스크 수행 계획의 유효성 검사이 외에도 다른 형태의 유효성 검사를 생각해 볼 수 있으며 이는 보다 완전한 유효성 검사를 위해 필요하며 향후 좋은 연구주제가 될 것이다. 또한 상황의 동적인 변화를 고려하기 위해 더 많은 변수들을 고려하는 방법이나 제안한 방법의 다양성을 검증하기 위해 다른 스마트폰 OS에 적용하면 발생할 수 있는 문제점 분석 등에 관한 향후 연구가 필요하다.

참고 문헌

[1] Tasker, <http://tasker.dinglisch.net/>  
 [2] Tasker, <http://www.androidauthority.com/how-to-use-tasker-199872/>  
 [3] AutomateIt, <http://automateitapp.com/>  
 [4] AutomateIt, <http://www.androidauthority.com/automateit-app-review-126143/>  
 [5] G. W. Musumba and H. O. Nyongesa, "Context

awareness in mobile computing: A review," Int. Journal of Machine Learning & Applications, Vol.2, No.1, 2013.  
 [6] J. Pauty, D. Preuveneers, P. Rigole, and Y. Berbers, "Research challenges in mobile and context-aware service development," in Proceedings of the Future Research Challenges for Software and Services Conference, 2006.  
 [7] A. K. Dey, D. Salber, and G. D. Abowd, "A Conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications," Human-Computer Interaction, Vol.16, pp.97-166, 2001.  
 [8] H. Fagrell, K. Forsberg, and J. Sanneblad, "FieldWise: A Mobile Knowledge Management Architecture," ACM Conf. on Computer Supported Cooperative Work, pp.211-220, 2000.  
 [9] J. Herstad, D. Thanh, and J. Audestad, "Human Centered Mobile Communication Using Contextual Information," Int. Workshop on Collaboration and Mobile Computing, 1999.  
 [10] J. Pascoe, "Adding Generic Contextual Capabilities to Wearable Computers," IEEE Int. Conf. on Wearable Computers, 1998.  
 [11] D. P. Siewiorek, A. Smailagic, L. Bass, J. Siegel, and R. Martin, "Adtranz: A Mobile Computing System for Maintenance and Collaboration," IEEE Int. Conf. on Wearable Computers, pp.25-32, 1998.  
 [12] 류윤지, 김바울, 김상욱, "사용자 상황인지 서비스를 위한 안드로이드 기반 모바일 플랫폼," 한국멀티미디어학회 추계학술발표대회 논문집, 제12권, 제2호, 2009(11).  
 [13] 박연진, 송경아, 황재원, 창병모, "은톨로지 기반의 개인화된 여행 추천 시스템의 구현," 한국콘텐츠학회논문지, 제15권, 제9호, 2015(9).

저 자 소 개

박 나 연(Na-Yeon Bak)

준회원



- 2016년 현재 : 숙명여대 컴퓨터 과학부 재학

<관심분야> : 모바일 소프트웨어

정 다 해(Da-hae Chung)

준회원



- 2016년 2월 : 숙명여대 컴퓨터과 학부(이학사)
- 2016년 3월 ~ 현재 : 숙명여대 대학원 석사과정

<관심분야> : 모바일 소프트웨어

창 병 모(Byeong-Mo Chang)

정회원



- 1988년 2월 : 서울대학교 컴퓨터 공학과(공학사)
- 1990년 2월 : KAIST 전산학과 (공학석사)
- 1994년 2월 : KAIST 전산학과 (공학박사)

- 1995년 3월 ~ 현재 : 숙명여대 컴퓨터과학부 교수

<관심분야> : 소프트웨어 분석, 검증, 보안

최 광 훈(Kwanghoon Choi)

정회원



- 1994년 2월 : KAIST 전산학과 (공학사)
- 1996년 2월 : KAIST 전산학과 (공학석사)
- 2003년 8월 : KAIST 전산학과 (공학박사)

- 2016년 9월 ~ 현재 : 전남대학교 전자컴퓨터공학부 부교수

<관심분야> : 프로그래밍언어, 컴파일러, 소프트웨어 공학, 모바일소프트웨어