

Android 모바일 응용을 위한 계획 실행 모델의 설계 및 구현

오휘경*, 정종근*, 박찬영**, 김인철[†]
경기대학교 컴퓨터과학과 *학부과정, **석사과정, [†]교수
e-mail:{ohkv770, ttigem, cyboys, kic}@kyonggi.ac.kr

Design and Implementation of a Plan Execution Model for Android Mobile Applications

Hui-Kyung Oh*, Jong-Geun Jung*, Chan-Young Park**, In-Cheol Kim[†]
^{*}Under-Graduate Course, ^{**}Master Course, [†]Faculty
Department of Computer Science, Kyonggi University

요 약

본 논문에서는 스마트폰 환경에서 개인 사용자 편의 서비스 프로그램 개발의 기초가 되는 작업 계획 모델 및 실행 모델을 제시하고, 이 모델을 활용하여 Android 플랫폼에서 개발된 자동 작업 실행 체계인 Smart Script 시스템의 설계와 구현에 대해 소개한다.

1. 서론

최근 들어 iPhone, Android 폰 등 다양한 스마트폰들이 급속히 보급되면서, 스마트폰을 이용한 개인 사용자 편의 서비스 개발에 대한 연구가 활발히 진행되고 있다. 기존의 데스크톱 컴퓨팅 환경과는 달리 스마트폰 환경은 휴대성과 이동 통신성과 같은 장점들뿐만 아니라 화면과 키보드의 소형화, 배터리의 제한성과 같은 단점들도 동시에 가지고 있다. 따라서 이러한 스마트폰의 특성을 고려하여 키보드 입력이나 화면 터치 등 스마트폰 사용자의 직접 조작은 가능한 줄이고, 사용자를 대신해 보다 쉽고 편리하게 사용자가 요구하는 작업들을 수행해줄 수 있는 자동 계획, 기계 학습, 지능형 에이전트 등의 인공지능 기술들에 대한 관심이 높아지고 있다. Siri[1], CALO[2] 등은 이러한 인공지능 기술을 바탕으로 개발된 대표적인 개인 사용자 편의 서비스 프로그램들이다. 본 논문에서는 스마트폰 환경에서 개인 사용자 편의 서비스 프로그램 개발의 기초가 되는 작업 계획 모델 및 실행 모델을 제시하고, 이 모델을 활용하여 Android 플랫폼에서 개발된 자동 작업 실행 체계인 Smart Script 시스템의 설계와 구현에 대해 소개한다.

2. 계획 및 실행 모델

자동 계획 수립(automated planning)을 위해서는 계획을 구성하는 기본 동작(primitive action)들에 대한 모델이 필요하며, 이러한 동작 모델에 관한 연구는 STRIPS의 초기 동작 모델에서 시작하여 최근에는 다양한 실세계 제

약들을 표현할 수 있는 PDDL과 같은 표준 모델로 발전하고 있다. 동작 모델과는 달리 계획 자체를 표현하는 계획 모델에 관한 연구는 아직 표준화 단계에 이르지 못하고 있다. 하지만 최근 들어 자동 계획 수립이외에도 기계 학습 또는 사용자에게 의한 모델링 등 다양한 방법으로 계획 지식을 직접 확보할 수 있는 방법들이 개발되고 계획의 응용 영역이 더욱 확대됨에 따라, 보다 더 표현력이 뛰어난 계획 모델에 관한 요구가 높아지게 되었다[3]. 본 연구에서는 스마트폰 환경의 응용 목적에 부합하는 계획 모델 p와 실행 모델 PEM을 아래와 같이 제안한다.

계획 모델 p = (ID, Name, Type, Body, Comment),

여기서 ID는 식별번호(identification number), Name은 계획 이름(plan name), Type은 사건-주도 계획(event-driven plan), 목표-주도 계획(goal-driven plan), 혹은 명령-주도 계획(command-driven plan) 중 하나, Body는 <a1, ..., an>와 같은 일련의 기본 동작들, Comment는 계획에 관한 설명문이다.

계획 모델 p는 서로 다른 3 가지 유형의 계획들을 표현할 수 있다. 먼저, 사건-주도 계획(event-driven plan) p_e은 특정 사건이 발생하면 자동으로 실행되는 계획을 의미하며, 아래와 같이 계획이 실행되기 전에 만족해야 하는 전-조건(precondition)을 반드시 포함한다. 이 전-조건에는 계획 실행을 요구하는 특정한 사건(event) 발생에 관한 조건들을 포함해 다양한 상태 조건들을 포함할 수 있다.

사건-주도 계획 p_e =

(ID, Name, Type, Precondition, Body, Comment)

※ 본 연구는 경기도의 경기도지역협력연구센터사업의 일환으로 수행하였음

목표-주도 계획(goal-driven plan) p_g 은 특정 목표(goal)를 달성하기 위해 실행될 수 있는 계획을 의미하며, 아래와 같이 이 계획의 실행으로 달성 가능한 목표(goal)를 반드시 포함해야 하며, 필요하면 전-조건(precondition)도 선택적으로 포함할 수 있다.

목표-주도 계획 $p_g =$
(ID, Name, Type, [Precondition], Goal, Body, Comment)

명령-주도 계획(command-driven plan) p_c 은 언제든지 사용자의 필요에 따라 선택하여 즉시 실행시킬 수 있는 계획을 의미하며, 따라서 명령-주도 계획은 아래와 같이 어떤 전-조건도 목표도 포함할 필요가 없다.

명령-주도 계획 $p_c =$ (ID, Name, Type, Body, Comment)

주어진 작업 목표와 환경의 실시간 상태를 고려하여 미리 정의된 계획들을 동적으로 실행하는 체계를 하나의 계획 실행 모델(plan execution model)로 표현할 수 있다. 본 연구에서는 앞서 정의한 계획 모델 p 에 부합되는 계획 실행 모델 PEM을 다음과 같이 정의한다.

계획 실행 모델 PEM = <P, S, W, U, I, N, T>,

- P : 계획(plan)들의 집합, 즉 계획 라이브러리(plan library), $P = P_e \cup P_g \cup P_c$, 여기서 P_e 는 사건-주도 계획들(event-driven plans), P_g 는 목표-주도 계획들(goal-driven plans), P_c 는 명령-주도 계획들(command-driven plans)을 각각 나타냄.
- S : 외부 환경(environment)에서 발생하는 사건(event)들을 감지하는 감지기(sensor),
 $S : E \rightarrow 2^F$, 이때
E는 사건들(events), F는 사실들(facts)의 집합.
- W : 환경의 현재 상태를 나타내는 사실(fact)들의 집합, 즉 월드 모델(world model),
 $F \subseteq W$
- U : 계획 실행과 연관된 사용자의 제어 명령들(control commands),
{execute_plan, stop_plan, activate_plan, deactivate_plan, insert_plan, update_plan, delete_plan, ...} $\subset U$
- I : 사용자의 제어 명령이나 월드 모델의 변화에 따라 계획 라이브러리에서 다음에 실행할 하나의 계획을 결정하는 해석기(interpreter),
 $I(W, u, P) = p$,
이때, 계획 $p \in P$ 는 $W \models \text{PRECOND}(p)$ 이거나 사용자 명령 $u \in U$ 와 연관된 것임.
- N : 실행이 결정된 계획의 실제 수행과정을 관리하는 의도 구조(intention structure),
 $N(p) = \text{SUCCESS}(\text{DO}(a_1))?$ $N(p') : \text{STOP}(p)$,
이때 $\text{BODY_SEQ}(p) = \langle a_1, a_2, \dots, a_n \rangle$,
 $\text{BODY_SEQ}(p') = \langle a_2, a_3, \dots, a_n \rangle$, 동작 $a_k \in A$
- T : 계획을 구성하는 각 동작(action)을 실제 수행하여 환경을 변화시키는 실행기(effector),
 $T : A \rightarrow E$

계획 모델 p 와 실행 모델 PEM은 다음과 같은 특성을 가지고 있다. 첫째, 3 가지 서로 다른 유형의 계획들을 지원함으로써, 응용영역 고유의 다양한 절차적 지식(procedural knowledge) 즉, 작업 계획 지식을 매우 효과적으로 표현하고 실행할 수 있다. 둘째, 계획 실행에 관한 사용자의 실시간 제어(realtime control)를 허용함으로써, 시스템의 일방적 실행이 아닌 사용자와의 상호작용에 의한 계획 실행(interactive plan execution)이 가능하다. 셋째, 사건-주도 계획(event-driven plan)들을 통해 환경의 변화에 매우 빠르게 반응할 수 있다. 넷째, 목표-주도 계획(goal-driven plan)들을 통해 목표 지향적인 추론과 계획 실행이 가능하다. 다섯째, 각 기본 동작의 실행 성공 여부에 대한 실시간 모니터링(realtime monitoring)을 통해 해당 계획의 실행 계속 여부를 결정할 수 있으므로 작업 성공률(task completion)을 높일 수 있다. 여섯째, 자동 계획 합성(plan synthesis)과 계획 학습(plan learning)을 위한 공통의 효과적인 기초 토대를 제공한다. 일곱째, 다양한 모바일 응용에 적합하도록, 작지만(lightweight) 매우 강력한 핵심 기능들을 갖도록 설계된 계획 및 실행 모델이다.

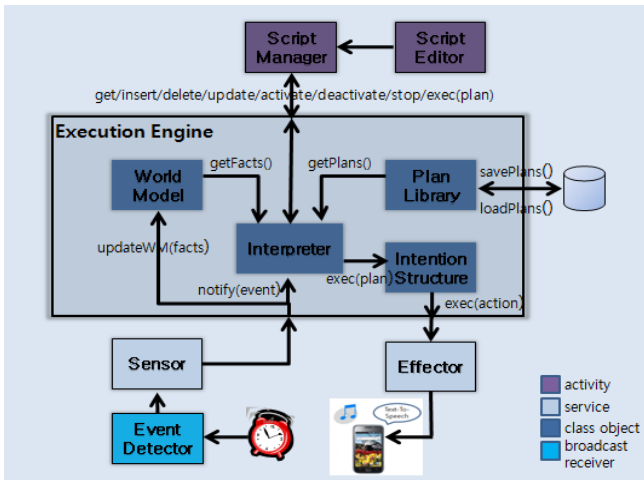
<표 1> 스크립트 형식

```
(script ScriptID
 :name Name
 :type event | goal | command
 [:goal Goal ]
 [:precondition (and Condition1 Condition2 ...)]
 :body (Action1 Action2 ...)
 [:failure (and Condition1 Condition2 ...)]
 [:comment Description])
```

3. 구현 및 응용

앞서 정의한 계획 및 실행 모델의 유용성을 확인하기 위해, 이 모델들을 이용하여 Android 폰 사용자를 위한 편의 서비스의 하나인 Smart Script 시스템을 설계하고 구현하여 보았다. Smart Script 시스템은 스마트폰 사용자가 일상생활에서 자주 반복 수행해야 하는 작업이나 예약 실행하고 싶은 작업들을 하나의 스크립트(script)로 정의해 두었다가 자동으로 실행할 수 있는 모바일 응용 프로그램을 말한다. 일반적으로 하나의 스크립트는 미리 정의되어 있는 기본 작업(primitive task)들의 순차적 결합으로 구성되며, 스크립트에 연관된 사건(event)의 발생이나 사용자에게 의해 제시되는 작업 목표(task goal) 혹은 제어 명령(control command) 등이 해당 스크립트의 실행을 촉발하는 트리거(trigger) 역할을 한다. 이러한 개념의 스크립트는 기본 동작(action)들로 구성된 하나의 계획(plan)으로 볼 수 있으며, 따라서 Smart Script 시스템은 본 논문에서 제시한 계획 및 실행 모델을 적용하기에 적합한 응용 프로그램으로 판단된다.

<표 1>은 계획 모델 p 에 기초한 스크립트의 형식을 나타낸다. failure 부분은 스크립트 실행 중 오류 상태를 검사하기 위한 조건들로 구성된다.

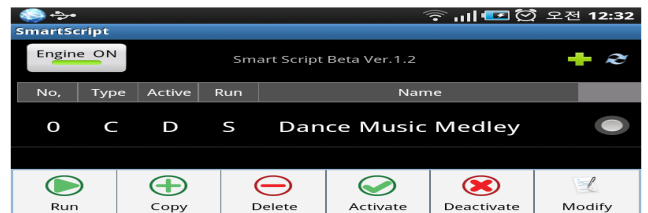


(그림 1) Smart Script 시스템 구조

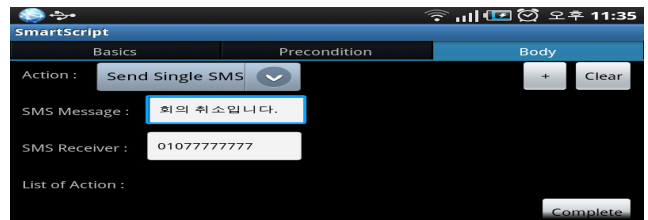
Smart Script 시스템은 <그림 1>과 같이 크게 스크립트 자동 실행을 총괄하는 실행 엔진(Execution Engine), 환경으로부터 다양한 사건과 상태 정보를 받아들이는 센서(Sensor), 각 기본 작업/동작의 실행을 담당하는 실행기(Effector), 그리고 사용자가 스크립트를 편집하고 수정할 수 있는 스크립트 편집기(Script Editor), 사용자가 스크립트의 실행을 실시간으로 감시하고 제어할 수 있는 스크립트 관리기(Script Manager) 등으로 구성된다. 센서는 문자 메시지 도착(SMS arrival), 전화 벨 울림(call ring), WiFi 검출(WiFi detect), 예약 시간 통보(timer notification) 등 스마트폰 환경에서 발생 가능한 다양한 사건들을 감지하고 이들과 연관된 새로운 사실(fact)들로 실행 엔진 내부의 월드 모델(World Model)을 갱신한다. 이러한 센서는 Android 브로드캐스트 리시버(broadcast receiver)와 서비스(service)의 결합으로 구현되었다.

계획 실행 모델 PEM에 기초해 스크립트 실행을 총괄하는 실행 엔진은 내부에 각기 다른 역할을 담당하는 월드 모델(World Model), 계획 라이브러리(Plan Library), 해석기(interpreter), 의도 구조(Intention Structure) 등의 구성요소들을 포함하고 있다. 월드 모델은 센서를 통해 감지된 환경의 상태나 사용자 입력 데이터, 혹은 각 기본 작업/동작의 실행 결과 데이터 등을 나타내는 사실(fact)들을 저장, 갱신함으로써, 언제나 스크립트가 실행되고 있는 현재 시스템의 상황을 표현한다. 반면에 계획 라이브러리에는 현재까지 사용자가 정의한 스크립트들이 실행 대기 상태로 저장되어 있으며, 필요할 때 DB에 스크립트들을 저장하거나 DB에 저장된 스크립트들을 가져오기도 한다. 해석기는 월드 모델의 사실(fact)들을 기초로 계획 라이브러리 내의 각 사건-주도 스크립트의 전-조건(precondition)이 만족되는지 검사하거나, 또는 사용자가 제시한 작업 목표(task goal)에 각 목표-주도 스크립트가 매치되는지 검사하는 등의 방법으로, 다음 실행 할 하나의 스크립트를 선택하는 역할을 수행한다. 의도 구조는 해석기에 의해 선택된 스크립트를 전달받아 해당 스크립트 몸체(body)에 기술된 기본 작업/동작들을 차례대로 실행함으로써 각 스크

립트가 의도하는 작업을 수행하는 역할을 담당한다. 스크립트 몸체에 새로운 목표 동작(goal action)을 포함하고 있는 경우, 이것을 하나의 부속목표(subgoal)로 간주하고 해석기를 통해 이 부속목표를 달성할 수 있는 새로운 목표-주도 스크립트를 선택하여 실행한다. 따라서 하나의 작업 스크립트는 실행 도중에 다수의 다른 부속작업 스크립트들의 실행을 유발할 수 있다. 이것은 스크립트 실행과정 중에 목표와 스크립트 간의 관계에 따라 자동적으로 계층적 작업 분해(hierarchical task decomposition)가 이루어지는 것을 의미한다. 실행 엔진은 항상 백그라운드에서 수행 상태에 있으면서 스크립트와 연관된 사건 발생과 사용자 입력에 따라 빠른 스크립트 실행을 보장하기 위해 하나의 Android 서비스로 구현되었다. 실행 엔진이 Smart Script 시스템의 논리적 의사결정(decision-making) 과정을 구현한 것이라고 보면, 별도의 Android 서비스로 구현된 실행기(Effector)는 웹브라우저, 캘린더, Email 프로그램 등 다른 기존의 전용 프로그램들을 이용하거나 Android API로 구현한 자체 프로그램을 이용하여 실행 엔진에서 요청하는 각각의 기본 작업/동작(primitive task/action)을 실제로 실행해주는 역할을 담당한다.



(그림 2) 스크립트 관리기의 실행 화면



(그림 3) 스크립트 편집기의 실행 화면

스크립트 관리기(Script Manager)는 Smart Script 시스템의 주된 사용자 인터페이스를 제공한다. 사용자는 스크립트 관리자를 통해 명령-주도 스크립트중 하나를 직접 실행(run)시킬 수 있으며, 실행 중인 스크립트를 중지(stop)시킬 수 있고, 사건-주도 스크립트들을 활성화(activate)하거나 비활성화(deactivate)할 수 있는 등 스크립트들에 대한 실시간 제어가 가능하다. 또한, 스크립트 편집기의 도움을 받아 새로운 스크립트를 생성(create), 복사(copy), 변경(modify), 삭제(delete)할 수 있다. 스크립트 편집기(Script Editor)는 스마트폰 화면의 크기와 키보드의 제한성을 고려하여, 대부분의 편집과정에 그래픽 아이콘과 메뉴 선택 방식을 적용함으로써 사용자의 키보드 입력을 최소화하도록 구현하였다. (그림 2)는 실행 중인 Smart Script 시스템의 스크립트 관리기 화면을, (그림 3)은 스크립트 편집기 화면을 각각 나타낸다.

<표 2> 사건-주도 스크립트의 예

```
(:script 001
:name 아침알람
:type event
:precondition (and (EQ event_type timer_notification)
(EQ time "07:00"))
:body ((speak "Please wake up.
It's seven o'clock in the morning.")
(playmusic "Morning.mp3")
(speak "Please wake up.
It's seven o'clock in the morning.")
(playmusic "Morning.mp3"))
:failure (and (EQ event_type playmusic_failure)))
```

<표 3> 목표-주도 스크립트의 예

```
(:script 002
:name 회의취소문자보내기
:type goal
:goal (회의취소알림)
:body ((send_SMS "011-222-3333" "오늘 예정되었던
회의가 취소되었습니다.")
(send_SMS "011-333-4444" "오늘 예정되었던
회의가 취소되었습니다.")
(send_SMS "010-444-5555" "오늘 예정되었던
회의가 취소되었습니다."))
:failure (and (EQ event_type send_SMS_failure)))
```

<표 4> 명령-주도 스크립트의 예

```
(:script 003
:name 걸그룹댄스음악메들리
:type command
:body ((speak "You'll listen to the medley of dance
musics sung by Korean Girl Groups.")
(speak "The first is Oh sung by Girls' Generation.")
(playmusic "Oh.mp3")
(speak "The next is Huh sung by Four Minute.")
(playmusic "Huh.mp3")
(speak "The last is Push Push sung by Sistar.")
(playmusic "Push.mp3"))
```

구현된 Smart Script 시스템을 이용하여 실험용 샘플 스크립트들을 몇 가지 작성하여 테스트해보았다. <표 2>는 사건-주도 스크립트들 중 하나인 “아침 알람” 스크립트를 텍스트 형태로 표현한 것이다. 이 스크립트는 아침 7시가 되면 자동으로 실행되며, 정해진 음성 안내와 음악 플레이를 반복하도록 정의되어 있다. 음성 안내는 Android의 TTS(Text-To-Speech) 기능을 이용하여 구현하였다. <표 3>은 목표-주도 스크립트중 하나인 “회의 취소 문자보내기”를 나타낸다. 이 스크립트는 “회의취소알림” 목표(goal)를 달성할 수 있는 스크립트들 중 하나이며, 3명의 서로 다른 수신자에게 회의취소 사실을 알리는 문자 메시지(SMS)를 자동으로 발송한다. <표 4>는 사용자가 스크립트 관리자를 통해 직접 선택하여 실행할 수 있는 명령-주도 스크립트중 하나인 “걸그룹 댄스음악 메들리”이다. 사용자가 평소 좋아하는 음악들을 모아 음성 안내를 곁들여 자신만의 음악앨범을 구성하여 저장해두었다가, 원하는 때는 언제나 스크립트 실행을 통해 손쉽게 들을 수 있는 스크립트의 예이다. 샘플 스크립트들의 생성과 실행을 통해 Smart Script 시스템과 이것의 기초가 되는 계획 실행 모델의 유용성과 효율성을 확인할 수 있었

다. 특히 Smart Script 시스템은 월드 모델의 최신성을 유지하기 위해 짧은 시간 주기별로 계속해서 많은 상태 정보를 가지고 월드 모델을 갱신하지 않고, 대신 특정 사건이 발생할 때만 관련 상태 정보를 중심으로 월드 모델을 갱신하도록 설계되었다. 이러한 특성은 스마트폰의 제한적인 자원에 매우 효과적 동작하는 것으로 드러났다. 또 Smart Script 시스템에서는 사용자에게 각 스크립트의 실행 상태 정보와 함께 각 스크립트에 대한 동적 실행 제어 기능을 함께 제공함으로써, 시스템이 일방적으로 스크립트들을 자동 실행하는 완전 자율 시스템(fully autonomous system)에 비해 사용자가 훨씬 시스템 이용에 편안함을 느꼈다. 향후 기본 작업/동작들과 감지 가능한 사건들을 추가 확장하면, 보다 더 다양한 목적의 응용 스크립트들의 생성이 가능해 스마트폰 사용자의 편의성을 더 높일 수 있을 것으로 판단하였다.

4. 관련연구

CALO 시스템 개발에 적용된 SPARK[2]은 대표적인 BDI 기반의 계획 실행 구조이나, 실행 모델 PEM과는 달리 사용자의 실행 제어 기능을 직접 지원하지 않는다. 이 동 로봇과 천체 망원경 제어분야에 적용된 IDEA[4]는 내부 및 외부의 자동 계획 수립기와 연동 가능한 계획 실행 기이다. IDEA는 다양한 유형의 계획 모델을 지원하나, 불필요한 제어 기능들이 많고 처리 속도가 느리며 규모가 너무 커서 모바일 응용 프로그램 개발에는 적합하지 않다.

5. 결론

본 논문에서는 모바일 응용에 적합한 계획 모델과 실행 모델을 제시하고, 이들을 기초로 Android 폰 사용자를 위한 서비스 프로그램인 Smart Script를 구현하였다. 몇 가지 샘플 스크립트들을 통해 계획 및 실행 모델의 유용성과 효율성을 확인할 수 있었다. 스크립트 몸체에 조건부 분기, 반복 등의 실행제어구조를 추가 도입하고, 다양한 기본 작업/동작들을 추가 구현하는 등의 시스템 확장을 계획하고 있다.

참고문헌

- [1] T. Gruber, "Siri: A Virtual Personal Assistant", Keynote Presentation at the Semantic Technology Conference(SemTech-09), 2009.
- [2] D. Morley and K. Myers, "The SPARK Agent Framework", Proc. of the AAMAS-04, pp. 712-719, July 2004.
- [3] F. Ingrand and F. Py, Proc. of the 4th Workshop on Planning and Plan Execution for Real-World Systems, ICAPS-09, 2009.
- [4] N. M. Gregory, et al, "IDEA: Planning at the Core of Autonomous Agents," Proc. of AAAI-01, 2001.