

A Policy-Based Meta-Planning for General Task Management for Multi-Domain Services

Byunggi Choi[†] · Insik Yu^{††} · Jaeho Lee^{†††}

ABSTRACT

An intelligent robot should decide its behavior accordingly to the dynamic changes in the environment and user's requirements by evaluating options to choose the best one for the current situation. Many intelligent robot systems that use the Procedural Reasoning System (PRS) accomplishes such task management functions by defining the priority functions in the task model and evaluating the priority functions of the applicable tasks in the current situation. The priority functions, however, are defined locally inside of the plan, which exhibits limitation for the tasks for multi-domain services because global contexts for overall prioritization are hard to be expressed in the local priority functions. Furthermore, since the prioritization functions are not defined as an explicit module, reuse or extension of the them for general context is limited. In order to remove such limitations, we propose a policy-based meta-planning for general task management for multi-domain services, which provides the ability to explicitly define the utility of a task in the meta-planning process and thus the ability to evaluate task priorities for general context combining the modular priority functions. The ontological specification of the model also enhances the scalability of the policy model. In the experiments, adaptive behavior of a robot according to the policy model are confirmed by observing the appropriate tasks are selected in dynamic service environments.

Keywords : Task Management, Intelligent Robot, Meta-Planning, Policy Model, Model-Driven

다중 도메인 서비스를 위한 정책 모델 주도 메타-플래닝 기반 범용적 작업관리

최 병 기[†] · 유 인 식^{††} · 이 재 호^{†††}

요 약

지능형 로봇은 사용자의 요구에 따라 상황에 맞는 작업을 선택하여 서비스를 수행할 수 있어야 하며, 다양한 상황에 대해서 현재 가장 적합한 작업이 무엇인지 비교하고 선택할 수 있어야 한다. 이를 위해 절차적 추론 시스템을 활용하는 기존의 지능형 로봇시스템은 작업 모델 내부에 우선도 함수를 정의하고 이를 활용하여 작업 간 우선도를 비교하는 방법으로 작업관리 기능을 제공하고 있었다. 하지만 이러한 기존의 방법은 다중 도메인 서비스를 수행하는 데에 있어, 우선도 결정 함수가 명시적이지 않아 일반적인 상황에 대한 우선순위를 결정할 수 없고, 유용성을 비교하기 위한 표준적인 기준이 존재하지 않아 각 서비스 간의 우선도를 비교하는 방법이 범용적이지 않으며, 이를 재사용하거나 확장할 수 없다는 한계점이 존재했다. 우리는 이를 개선하기 위해 작업관리를 위한 정책 모델과 이를 활용하여 작업의 우선도를 계산하는 방법인 정책 모델 주도 메타-플래닝을 제안한다. 정책 모델은 메타-플래닝 과정에서 작업의 유틸리티를 명시적으로 계산할 수 있는 기능을 제공하며, 기초 모델을 통해 다양한 서비스 간의 유용성을 비교할 수 있게 해준다. 또한, 모델의 구성을 온톨로지를 활용함으로써 확장성을 가지는 특징이 있다. 실험을 통해 동적 환경에서 정책 모델에 따라 로봇의 행동이 변화하는 것을 관찰할 수 있었고, 이를 통해 서비스가 필요에 따라 우선도의 비교를 통해 선택되는 것을 확인할 수 있었다.

키워드 : 작업관리, 지능형 로봇, 메타-플래닝, 정책 모델, 모델 주도

1. 서 론

지능형 로봇이란 변화하는 환경 내에서 상황을 인지하고

사용자의 요구를 만족할 수 있는 다양한 도메인 서비스를 제공할 수 있는 로봇을 말한다. 이러한 로봇을 구현하기 위한 지능형 로봇 플랫폼은 지식관리, 상황 인지, 작업관리 등의 다양한 기능이 필요하며, 이를 실시간으로 수행하여 현재 환경에 적합한 작업을 선택할 수 있어야 한다.

이러한 지능형 로봇을 위해 필요한 기능 중 작업관리 기능을 구현하기 위하여 국내외로 다양한 접근 방법을 통한 연구가 활발하게 이루어지고 있으며, 절차적 추론 시스템(Procedural Reasoning System)이 전통적으로 활용되고 있다[1, 2].

* 이 연구는 2019년도 산업통상자원부 및 산업기술평가관리원(KEIT) 연구비 지원에 의한 연구임(과제번호:10077553).

† 정 회 원: 서울시립대학교 전자전기컴퓨터공학과 박사과정

†† 비 회 원: 서울시립대학교 전자전기컴퓨터공학과 석사과정

††† 중신회원: 서울시립대학교 전자전기컴퓨터공학부 교수

Manuscript Received: August 5, 2019

Accepted: September 16, 2019

* Corresponding Author: Jaeho Lee(jaeho@uos.ac.kr)

절차적 추론 시스템이란 동적 환경에서 복잡한 작업을 수행할 수 있도록 실시간 추론 기능을 제공하는 시스템으로, 주어진 상황 및 지식정보를 통해 현재 수행 가능한 작업 계획을 판단하고 수행하는 기능을 제공한다. 더 나아가 절차적 추론 시스템에선 하나의 목표를 달성하기 위해 다수의 작업 계획이 존재할 수 있으며, 이들을 비교하여 더 적합한 작업을 판단하기 위하여 작업의 우선도 개념을 적용하고 있다. 이러한 시스템을 활용하여 최근 개발된 로봇의 의사 결정 소프트웨어로는 대표적으로 CRAM과 프론티어 사업을 통해 개발된 작업관리 프레임워크(Task Management Framework)가 있다.

CRAM은 CoTeSys의 지원을 받아 개발된 지능형 로봇을 위한 추론 시스템으로, Lisp과 절차적 추론 시스템 중 하나인 RPL을 기반으로 구현된 자체적인 절차적 추론 시스템을 활용하고 있다. 이 시스템은 인지 능력을 가진 자동화된 로봇들을 설계 및 구현하는 데 필요한 의사 결정을 수행하는 가벼운 추론 메커니즘을 제공하고 있다.

작업관리 프레임워크는 2003년부터 진행된 프론티어 사업을 통해 개발된 지능형 로봇시스템으로, 유틸리티(Utility)를 적용한 절차적 추론 시스템인 JAM[3]을 활용한 것이 특징이다. 유틸리티는 작업의 우선도를 나타내며, 이를 결정하는 함수는 개발자가 직접 작업 계획에 정의하여 활용하는 방식을 취하고 있다.

이러한 절차적 추론 시스템들은 시나리오를 통해 구성된 환경에서 하나의 도메인 서비스를 수행하는 것을 연구의 목적으로 하고 있다. 하지만 실제 지능형 로봇이 서비스를 수행하는 환경은 복수의 도메인 서비스를 동시에 수행해야 하거나, 하나의 목표를 달성하기 위한 다중 도메인 작업 간 우선도 비교를 수행해야 하는 상황이 발생할 수 있다. 이러한 경우에 기존의 절차적 추론 시스템은 다음과 같은 한계점을 가지고 있다.

1) 우선도 함수가 각 작업 계획 내부에 암시적으로 정의된다.

지능형 로봇은 상황에 가장 적합한 작업을 판단하고 이를 수행할 수 있어야 한다. 기존의 방법은 작업 모델 내부에 정의된 우선도 함수를 통해 작업의 유용성 비교를 수행하는 방법을 채택하였다. 지능형 서비스 로봇의 적용 범위가 확장됨에 따라, 다양한 도메인의 작업 계획 간 일관적인 우선도 비교를 수행할 수 있어야 하지만, 기존의 방법에선 각 유틸리티는 작업 계획 추론 과정에서 작업 모델 내의 개별적인 방법을 통해 도출된다. 이러한 일관적이지 않은 유틸리티 값 결정 과정으로 인하여 작업 계획 추론 과정이 복잡해질 뿐만 아니라, 일반적인 상황에 대해 유틸리티를 통한 작업 계획 간의 범용적인 우선순위를 결정할 수 없다는 한계점이 있다.

2) 서비스 간 유틸리티를 비교하기 위한 별도의 기준이 존재하지 않는다.

지능형 로봇은 다양한 도메인의 서비스를 수행할 수 있어야 하며, 이를 위해 지능형 로봇 플랫폼은 서비스 간 우선도 비교 방법을 제공하여야 한다. 기존의 방법은 유틸리티를 통해 서비스 간 유용성 비교 방법을 지원하고 있으나, 함수 작성에 대한 별도의 지침이 존재하지 않아 개발자가 서비스 개

발 시 자신의 기준을 통해 우선도 함수를 정의하게 된다. 이처럼 개발자들이 정의한 작업에 대한 각 우선도 함수의 평가 기준이 다르므로 다양한 도메인 서비스 간 범용적인 유용성의 비교적으로도 활용하기에 부적합하다.

3) 우선도 함수가 작업 모델 및 해당 서비스와 결합되어 있다. 이러한 작업 모델 및 서비스와 유틸리티 함수 간 의존성으로 인해 새로운 작업 계획을 작성할 때마다, 기존의 우선도 함수를 재사용하거나 확장하기 매우 어려워 매번 새로운 함수를 정의해야 한다.

위 세 가지 요구사항을 만족하기 위하여 본 논문에서는 범용적 작업관리를 위한 정책(Policy) 모델 주도 메타-플래닝(Meta-Planning) 방법을 제안한다. 이는 1) 정책 주도 메타-플래닝 과정을 통해 일관적이고 명시적인 과정으로 작업의 유용성을 결정하는 방안을 제공하고, 2) 다양한 도메인 서비스 간의 일관적인 비교를 위한 범용적인 항목을 제공하며, 3) 모델의 구성을 온톨로지를 활용함으로써, 기존의 정책을 재사용하거나 쉽게 확장할 수 있다.

2. 관련 연구

2.1 메타-플래닝

플래닝(Planning)은 어떠한 목표를 달성하기 위한 작업 계획을 추론하는 과정으로, 이를 위해선 주변 환경에 대해 나타내는 환경 정보와 작업 계획, 목표 등에 대한 작업 정보가 필요하다. 환경 정보는 카메라, 마이크와 같은 센서를 통해 정보를 입력받는 일관된 과정을 통해 비교적 단순하게 얻을 수 있으나, 작업 정보는 그 대상에 따라 다른 방법을 통해 얻어야 한다. 목표의 경우, 사용자의 요구사항을 분석하여 이를 달성시키는 상태를 분석해야 하며, 작업 계획 변수의 경우, 환경 정보를 활용하여 적절한 값을 바인딩하는 과정이 필요하다. 이러한 작업 정보를 획득하는 과정들은 일련의 추론으로 대체할 수 있으나, 이를 작업 추론 과정과 동시에 수행할 경우 문제의 복잡도가 증가하게 된다. Wilensky[4]는 이러한 문제를 해결하기 위하여 메타-플래닝 방안을 제시하였다.

메타-플래닝은 작업 계획과 관련된 지식을 추론하는 과정을 의미하며, 이를 통해 하나의 복잡한 문제를 더욱 효율적인 해결이 가능하다. 메타-플래닝 과정을 통해 추구하는 대상인 메타 목표는 현재 추구해야 할 목표, 계획의 변수 바인딩, 수행할 작업 계획의 방향성 등 작업과 관련된 정보 전반이 될 수 있다. 메타 작업 계획은 메타 목표를 달성하기 위한 작업 계획이며, 일반적인 작업 계획과 같이 목표를 달성하기 위한 일련의 과정을 서술하고 있으나, 달성하고자 하는 대상이 일반적인 목표가 아닌 메타 목표라는 차이가 있다.

작업의 유용성은 작업 계획을 선택하기 위한 정보로서 일반적인 추론 과정을 통해 계산할 경우, 환경의 변화에 따라 실시간으로 재계산해야 하므로 추론의 복잡도가 증가하여 효율적인 작업 선택이 어려워진다. 그러므로 이는 메타-플래닝 과정을 통해 결정되는 것이 적합하다.

2.2 커뮤니티 컴퓨팅에서의 정책

커뮤니티 컴퓨팅은 유비쿼터스 환경에서 여러 에이전트 간의 협력을 통해 문제를 해결하는 방법으로, 목표를 효율적으로 달성하기 위해 정책을 통해 에이전트의 행동을 제어하는 방식을 활용하고 있다. 여기서 정책은 시스템의 행위 선택을 관리하고, 변경시키는데 사용되는 정보[5]로 정의된다. 이러한 정책을 서술하기 위한 언어와 이를 활용하여 여러 에이전트를 관리하기 위한 시스템이 다양하게 연구되었다.

1) Ponder

Ponder[6]는 분산 환경에서 보안 및 관리 작업을 위한 정책을 서술하기 위해 개발된 선언적 객체 지향 언어이다. Ponder에선 다음과 같은 두 타입의 정책을 지원한다. 사용자의 활동을 제어하기 위한 접근 제어(Access Control) 정책과 특정 상황에 대해 시스템이 반드시 수행해야 하는 작업을 지정하기 위한 의무(Obligation) 정책이다. 그리고 이를 커뮤니티 내에서 효과적으로 활용할 수 있도록 정책의 적용대상을 그룹과 역할 그리고 역할 간의 관계로 분류하였다.

접근 제어란 단순히 사용자가 어떤 정보에 접근하지 못하게 하는 것뿐만 아니라, 접근 권한이 존재하는 다른 사용자 혹은 시스템에 전달하거나, 접근 권한에 맞는 정보만을 필터링하여 전달하는 등 다양한 의미를 포함한다. Ponder의 접근 제어 정책 또한 저작(Authorisation), 정보 필터링(Information Filtering), 위임(Delegation), 자제(Refrain)의 4가지 기능을 수행한다. 저작 정책은 한 도메인의 구성원이 대상 도메인의 정보에 대해 수행할 수 있는 행동에 대해 정의한다. 이는 접근 권한 설정을 통해 정보를 무단 접근으로부터 보호하는 기능을 제공하며, 긍정 저작 정책을 통해 수행 가능한 행위를 정의하거나, 부정 저작 정책을 통해 수행 불가능한 행위를 정의할 수 있다. 정보 필터링 정책은 다양한 조건을 통해 특정 정보만을 접근할 수 있도록 제어하는 항목이다. 위치 서비스를 통해 정보에 접근할 경우, 별도의 조건이 존재하지 않으면 모든 사용자에게 대한 위치를 얻게 되므로 조건을 통해 시스템이 필요한 정보만을 획득할 수 있도록 제한한다. 위임 정책은 접근 권한을 일시적으로 양도하는 기능을 수행한다. 이는 저장 정책과 마찬가지로 긍정 혹은 부정을 통해 권한의 양도 가능 여부를 지정할 수 있다. 자제 정책은 특정 사용자가 대상 도메인의 정보에 대해 수행이 금지된 행위에 대해 정의한다. 이는 부정 저작 정책과 유사한 기능을 수행하지만, 적용대상이 사용자가 아닌 정보라는 차이점이 있다.

의무 정책은 접근 제어 정책과 다르게 반응 적인 항목으로, 시스템의 안정 및 유지를 위해 관리자가 수행해야 할 행위에 대해 정의한다. 개발자는 특정 이벤트를 지정하여 수행해야 할 행위 및 행위의 대상을 서술할 수 있다. 이벤트는 하나의 상황뿐만 아니라 다양한 상황의 조합을 지정할 수 있다.

Ponder는 정책의 목적과 표현 방법을 명확하게 제시하였다. Ponder 정책의 목적은 시스템의 행동을 제어하는 것으로, 각 정책의 이름을 통해 목적을 명시적으로 표현하는 것이

가능하다. 하지만 행동 제어라는 목적 때문에 정책이 나타낼 수 있는 의미가 한정적이며, 상황에 따라 적합한 작업을 나타낸다는 우리의 목표와 큰 차이점이 존재한다. 또한, 한번 구성된 정책을 재사용하기 위해서는 그룹과 역할 등에 대해 새롭게 정의해야 한다는 점에서 재사용성이 부족하다.

2) 분산 시스템을 위한 정책기반 관리

분산 시스템을 위한 정책기반 관리[7]는 정책을 기반으로 행동을 선택하는 관리자와 정책을 분리하여 관리하는 방안을 제안하였다. 이를 통해 관리자의 행위를 동적으로 제어 가능하며, 기존의 정책을 재사용할 수 있도록 설계되었다.

정책기반 관리에서 제공하는 정책은 두 종류로 수행할 수 있는 행위에 관해 서술하는 저작(Authorisation) 정책과 절대로 수행하여야 하는 혹은 수행하지 말아야 하는 행위에 관해 서술하는 의무(Obligation) 정책이다. 이 두 정책의 차이는 적용되는 대상이 다르다는 것으로, 저작 정책은 행위의 대상, 의무 정책은 행위의 주체를 대상으로 한다. 또한, 다양한 조건을 설정하여 구체적인 행동의 제어가 가능하다.

이 관리 시스템에서 정책은 객체 지향성을 고려하여 설계되었기 때문에, 하나의 정책 클래스에 대해 여러 인스턴스의 생성이 가능하며 각 인스턴스에 다른 조건을 설정할 수 있다. 더 나아가 정책은 다른 정책을 참조하여 구성될 수 있으며, 이를 통해 정책 간의 계층 구조를 표현하는 것도 가능하다. 또한, 정책과 정책의 적용대상을 분리하였기 때문에, 기존의 항목들도 손쉽게 재사용이 가능하다. 이와 같은 점들은 우리가 고려하는 확장성과 재사용성 방면에서 매우 적합한 구조로 되어있다고 판단된다. 하지만 한 도메인에 얽매이지 않는 범용적인 항목을 별도로 제공하고 있지 않기 때문에, 다양한 로봇 서비스 간의 유용성 비교에 활용하기엔 부적합하다.

2.3 유틸리티 기반 시스템

유틸리티란 어떠한 행위의 유용성을 나타낸 수치로써, 이를 통해 현재 수행할 수 있는 작업의 유용성을 비교하고 가장 높은 유용성을 지닌 행위를 선택하여 수행하는 시스템을 유틸리티 기반 시스템이라고 한다[8][9].

사람이 물건을 구매하기 전에 가격, 품질, 기능, 재료 등 다방면으로 고려하듯이, 행위의 유용성이라는 것은 다양한 요소를 반영하여 결정되어야 한다. 이처럼 유틸리티 함수는 유용성을 판단하는데 필요한 여러 항목의 값들을 활용하는 선형 방정식의 형태로 구성되며, 각각의 요소들은 방정식에서 활용되기 위한 상숫값으로 변환되어야 한다. 더 나아가 유틸리티 값이 실질적인 의미를 나타내기 위해선 환경의 변화를 실시간으로 반영할 수 있어야 하며, 각 변수의 갱신이 이루어질 때마다 유틸리티값은 재계산되어야 한다. 이렇게 변화하는 환경에 따라 실시간으로 갱신되는 유틸리티 값은 의사 결정 단계에서 작업을 선택하기 위한 하나의 척도로써 활용할 수 있다.

유틸리티는 다양한 상황에 대해 각 행위의 유용성을 수치화하여 이를 비교할 수 있다는 것이 가장 대표적인 장점이다.

하지만 이 외에 다양한 한계점을 내포하고 있는데, 먼저 유틸리티값의 표현력이 부족하다는 점이다. 다양한 요소를 고려하는 방정식을 통해 결정된 유틸리티 값은 행위의 유용성을 수치화해서 보여주나, 단순히 값만 보아서는 어떠한 기준을 통해 결정된 것인지 판단하기 어렵다. 각 유틸리티 값의 구체적인 의미를 파악하기 위해선 각 함수에서 고려하는 요소들과 각 요소의 값 결정 기준을 확인하여야 한다. 하지만 이러한 유틸리티 값의 기준은 해당 함수의 작성자가 아니면 명확하게 판단하기 어렵다. 이 외에도 각 유틸리티 값과 함수는 개발자 개인의 기준에 따라 정의되기 때문에, 다른 개발자가 작성한 항목과 일괄적인 비교가 불가능하다는 한계점 또한 존재한다.

2.4 CRAM & KnowRob

CRAM과 KnowRob[10]은 EU에서 CoTeSys 지원을 통해 개발된 지능형 로봇 작업관리 시스템이다. CRAM은 일반 가정환경과 같은 일상적인 생활 환경에서의 복잡한 조작 활동 수행하는 것을 중점으로 개발되었으며, 지식 추론기로서 KnowRob을 활용하고 있다.

CRAM은 Lisp과 RPL을 기반으로 구현된 자체적인 절차적 추론 시스템과 작업 서술 언어인 CPL을 통해 의사 결정 기능을 수행하며, COGITO 시스템이라는 메타 추론을 기능을 제공하는 것이 특징이다. 이 메타 추론을 통해 로봇이 작업을 수행하기 전에 해당 작업이 성공적으로 수행될 것인지를 판단하고 실패 위험이 있을 때, 작업을 변경하는 기능을 제공하고 있다.

지능형 로봇이 서비스를 수행하기 위해선 해당 서비스와 관련된 다양한 지식정보가 필수적이다. 더 나아가 로봇의 서비스 범위를 확장하기 위해선 다양한 도메인의 지식을 관리하고 이를 활용할 수 있어야 한다. KnowRob은 이러한 지식 처리를 위해 온톨로지를 활용하여 지식을 관리한다. 이를 통해 로봇이 새로운 도메인의 서비스를 위해 필요한 지식을 자유롭게 정의하고 추가할 수 있다. 더 나아가 작업 계획의 추론을 위해 필요한 지식이 부족하거나 불확실한 경우, 지식 추론을 통해 이를 보완하는 기능을 제공하고 있다.

CRAM은 메타 추론을 통해 작업의 성공 여부를 예측하고 이에 미리 대응할 수 있으며, KnowRob[10]은 온톨로지를 활용하여 다양한 도메인의 지식을 관리하고 이를 확장할 수 있다는 장점이 있다. 하지만 작업 간 우선도를 비교할 방법을 제공하지 않아, 상황에 따른 최적의 작업을 선택할 수 없다는 한계점을 가지고 있다.

2.5 작업관리 프레임워크

국내에서는 03년부터 13년까지 진행된 지식경제 프론티어 사업을 통해 지능형 로봇을 위한 작업관리 체계인 작업관리 프레임워크를 개발하였다. 작업관리 프레임워크는 절차적 추론 시스템인 JAM을 활용한 것이 특징이며, 이를 통해 상황의 변화에 능동적으로 대응하여 가장 적합한 작업을 선택하

는 기능을 제공하고 있다.

하나의 목표를 달성하기 위한 작업 계획은 복수로 존재할 수 있으며, 로봇은 현재 수행 가능한 작업 중 최선의 작업을 선택할 수 있어야 한다. 이러한 수행 가능한 여러 작업 계획 중 하나를 선택하기 위해선 이들을 비교하기 위한 기준이 필요하며, JAM에선 작업 계획에 유틸리티를 적용하여 작업의 유용성을 나타낼 수 있도록 지원하였다. 유틸리티 함수는 단순한 사칙연산뿐만 아니라 변수를 활용한 외부 함수를 지정할 수도 있다. 이를 활용하여 각 작업 계획별 유틸리티 함수를 설정할 수 있으며, 다양한 상황을 고려하여 각 작업에 맞는 함수를 정의한다면, 현재 환경에 맞는 가장 유용한 작업 계획을 선택하여 수행할 수 있다.

작업관리 프레임워크는 유틸리티를 통해 작업 계획 간 우선도 비교 방법을 지원하고 있으나, 몇 가지 한계점을 내포하고 있다. 가장 대표적인 문제는 유틸리티 함수 작성 지침이 별도로 존재하지 않는다는 점이다. 개인에 따라 우선도의 기준이 상이할 수 있으며, 따라서 서로 다른 개발자가 정의한 작업 계획을 혼용해서 사용할 경우 형평성 있는 비교가 어렵다.

3. 배경 연구

3.1 지식기반 로봇 지능체계

지능형 서비스 로봇은 주변 상황을 인지하고 서비스와 관련된 지식을 활용하여 사용자의 요구에 적합한 작업을 선택하고 이를 수행할 수 있어야 한다. 더 나아가 하나의 서비스 수행 과정에서 새로운 사용자의 요구가 발생하여도 이에 대응하여 동시에 복수의 서비스를 수행할 수 있어야 한다. 우리는 이러한 요구사항을 만족하기 위해 지식기반 로봇 지능체계라는 지능형 로봇 플랫폼을 개발하였다.

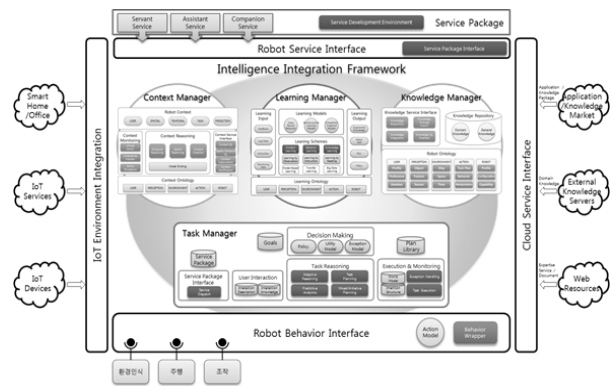


Fig. 1. Architecture of ARBI

지능체계는 지능형 서비스 로봇의 요구사항을 만족하기 위한 상황 정보 관리, 지식 정보 관리, 작업 추론, 외부 서비스용 인터페이스 등 다양한 기능을 지원한다. 지식 정보 관리 기능은 각 서비스에서 활용하기 위한 도메인 지식을 관리하며, 필요한 경우 외부 지식 저장소로부터 새로운 지식을 가져

오는 기능을 제공한다. 상황 정보 관리 기능은 다수의 센서를 활용하여 획득한 환경 정보를 기반으로 시공간 및 사용자에게 대한 추론을 통한 상황 정보 생성 및 관리를 수행한다. 작업 추론 기능은 상황 정보와 지식정보를 활용하여 사용자의 요구를 달성하기 위해 가장 적합한 작업 계획을 선택하고, 로봇 모듈을 통해 이를 수행하는 기능을 제공한다.

우리는 지능형 서비스 로봇이 서비스를 수행하는 데 필요한 정보를 집약하여 하나의 패키지로 구성하고, 이를 서비스 패키지라고 정의하였다. 지능체계는 하나의 서비스 패키지를 통해 하나의 서비스를 수행하며, 두 개 이상의 서비스를 동시에 수행해야 하는 경우, 각 서비스 패키지를 교체하며 수행할 수 있다. 하지만 서비스의 유용성을 결정하는 범용적인 방안을 별도로 제공하지 않아, 단순히 복수의 서비스를 동시에 수행하는 것이 가능할 뿐, 환경에 따라 이를 능동적으로 교체하며 수행하는 것은 불가능하다.

서비스의 유용성은 현재 상황을 다방면으로 고려하여 결정되어야 하며, 이는 단순히 센서 정보를 바탕으로 상황 정보를 생성하는 것이나 환경에 적합한 작업을 추론하는 것과 별개의 과정이 필요하다. 하지만 지능체계는 이러한 기능을 지원하지 않아, 서비스의 유용성을 결정할 수 없다는 한계점을 내포하고 있다.

4. 연구 내용

본 논문은 다중 도메인 서비스의 범용적 작업관리의 요구 사항을 만족하는 방안으로 정책 모델과 이를 활용한 정책 모델 주도 메타-플래닝 방법을 제안한다.

정책 모델은 로봇의 작업관리를 위해 행동 방향성을 명시적으로 표현하기 위한 모델이며, 이를 활용한 메타-플래닝 방법을 통해 일괄적으로 상황에 따른 작업의 유틸리티를 결정한다. 또한, 일반적인 상황에 대한 범용적인 서비스의 유용성을 결정하기 위한 범용 정책 모델을 제안하여 다양한 도메인 서비스 간의 유용성을 비교하는 방법을 제시한다. 더 나아가 범용 모델을 기반으로 특정 도메인의 작업관리를 위한 새로운 정책을 자유롭게 정의하고 확장할 수 있다는 장점이 있다.

4.1 정책 모델 주도 메타-플래닝 방법

정책 모델 주도 메타-플래닝 방법은 상황에 따른 작업의 유틸리티를 실시간으로 계산하는 방법으로 현재 목표를 위한 작업을 선택하는 일반적인 작업 추론 과정과 별도로 수행된다.

메타-플래닝을 수행하는 작업 추론기는 새로운 서비스를 시작할 때마다, 해당 서비스의 정책을 월드 모델에 추가한다. 지능형 로봇은 실시간으로 변화하는 환경을 인식하고 이 결과로 상황 정보를 생성한다. 새로운 상황 정보가 생성되거나 기존의 정보가 갱신될 때마다, 작업 추론기는 해당 정보를 자신의 월드 모델에 추가하고 해당 상황 정보와 관련된 정책이 있는지 확인한다. 관련된 정책이 존재할 경우, 해당 정책 모델의 값 결정 수식을 재계산한다.

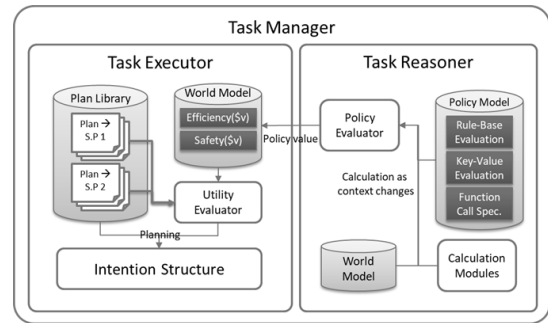


Fig. 2. Policy Model-Driven Meta-Planning Method

$$P = \sum_{i=1}^n r_i C_i + \sum_{i=1}^n r_i P_i \quad (1)$$

P=Policy value

r_n=Ratio of Context or Policy Value

C_n=Context Value

하나의 정책 모델값은 다양한 상황 정보 값 및 다른 정책들의 값을 조합하여 결정된다. 상황 정보는 단순히 숫자 값뿐만 아니라 다양한 타입의 인자를 가질 수 있으며, 각 정책 모델은 자신이 연관된 상황 정보의 인자를 수식에 활용하기 위한 값으로 변환하는 방법을 포함한다.

상황 변화에 따라 하나의 정책 값이 갱신된 이후, 재귀적으로 해당 정책과 관련된 다른 정책의 값들도 계산 과정을 수행한다. 이렇게 갱신된 정책 값들은 일반적인 추론 과정을 수행하는 모듈에 전달된다. 해당 모듈은 전달받은 정책 값을 활용하여 각 작업 계획의 유틸리티 수식을 계산한다.

$$U = \sum_{i=1}^n r_i P_i \quad (2)$$

U=Utility value

r_n=Ratio of Policy value

P_n=Policy value

유틸리티 결정 함수는 정책 값들을 이용한 선형방정식의 형태로 구성되며, 이들을 조합하는 비율의 총합은 1이다. 결정된 유틸리티 값은 각 작업의 우선도를 나타내며, 작업 추론 과정에서 활용된다.

4.2 범용 정책 모델

범용 정책 모델은 서비스의 범용적인 유용성을 평가하기 위한 기준을 제시하며, 이를 기반으로 도메인 서비스를 위한 정책을 쉽게 정의하고 확장할 수 있도록 지식 계층 구조를 활용하여 설계되었다.

범용 정책 모델은 효율성(Efficiency)과 안정성(Safety)의 두 가지 측면에서 서비스의 유용성을 표현한다.

지능형 로봇은 사용자의 요구를 만족시킬 수 있는 다양한 서비스 중 현재 수행하기에 가장 적합한 서비스를 선택할 수



Fig. 3. Global Policy Schema

있어야 한다. 시간이 넉넉하거나 사용 가능한 모듈이 많은 경우, 효율에 대해서 고려하지 않고 최고의 만족도를 제공하는 서비스를 수행하는 것이 바람직하다. 하지만 상황에 따라선 효율을 중시하여 최소한의 자원을 소모하는 서비스를 선택해야 할 수도 있다. 효율성은 이러한 서비스 효율의 중요도와 같은 측면에서 서비스의 유용성을 표현한다. 더 나아가 효율이라는 것은 중점적으로 고려되는 항목의 측면에 따라 다른 값으로 평가될 수 있어야 한다. 이를 위해 서비스의 결과(Quality), 서비스 수행 시간(Time) 그리고 서비스 수행 과정에서 활용되는 자원(Resource)의 측면에서 이를 표현할 수 있도록 항목을 정의하였다.

현대 사회에서 상품 혹은 서비스를 개발할 때, 사용자에게 대한 안전을 고려하는 것은 자명한 일이며, 또한 로봇의 서비스에도 적용될 수 있다. 안정성은 이러한 사용자에게 대한 안전과 같은 측면에서 서비스의 유용성을 표현한다. 더 나아가 서비스의 안전성은 단순히 서비스 사용자의 안전뿐만 아니라 서비스를 제공하는 로봇이 해당 서비스를 실제로 끝까지 수행 가능한지에 대한 여부나, 혹은 서비스 수행에 대한 제약에 관해서 서술할 수 있어야 한다. 이를 위하여 서비스 보안성(Security)과 서비스 수행 신뢰성(Reliability)을 정의하였다.

범용 정책 모델 항목들은 각각의 관계를 통해 지식 계층 구조를 이루고 있으며, 이들은 기본적으로 로봇 내에 포함되어 있다. 이를 통해 개발자가 새로운 서비스를 개발하더라도, 범용 정책 값의 조합을 통해 도메인 서비스에 대한 객관적인 유용성을 제시할 수 있으며, 서로 다른 도메인 서비스 간의 일관적인 유용성 비교를 가능하게 한다.

4.3 정책 모델의 확장과 재활용

범용 정책 모델은 지식 계층 구조를 활용하여 설계되었다. 이로 인해 자유롭게 정책을 확장할 수 있으며, 더 나아가 기존의 정책을 재활용할 수 있다. 새로운 도메인 서비스를 위하여 정책 모델을 확장하는 과정은 도메인 정책 클래스 정의, 정책 값 결정방법 정의, 정책 간 관계 정의의 세 단계로 이루어진다.

1) 도메인 정책 클래스 정의

정책 모델은 상황에 따른 작업 계획 혹은 서비스의 유용성을 판단하기 위한 기준을 서술한다. 새로운 서비스를 위한 정

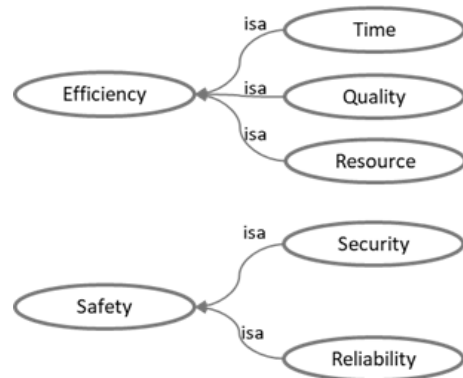


Fig. 4. Relation of Global Policy

책을 정의하기 위해선, 해당 서비스 도메인의 작업 계획과 상황 정보를 고려하여야 한다. 각 상황에 따라 우선되어야 하는 행위를 판단하고, 행위의 방향성을 나타낼 수 있는 이름으로 새로운 클래스를 정의하여야 한다.

2) 정책 값 결정방법 정의

정책의 값은 상황 정보 혹은 다른 정책들의 값을 조합하여 값을 결정된다. 도메인 정책은 해당 도메인 서비스의 상황 정보 및 정책의 조합으로 수식을 작성한다. 도메인 정책의 값 결정방법은 기존의 정책 수식을 도메인 서비스에 맞게 수정하여 적용하거나, 혹은 별도의 학습 모델을 활용하는 등 새로운 방법을 정의하여 활용할 수 있다.

3) 정책 간 관계 정의

각 정책의 방향성을 고려하여 새롭게 정의된 항목들과 기존 모델과의 관계를 정의한다. 이때, 새로운 항목들은 기초 항목과 반드시 일대일 관계로 정의할 필요가 없으며, 필요하다면 일대다 혹은 다대다 관계도 정의할 수 있다.

개발자는 새로운 도메인 서비스를 위해 자신이 직접 도메인 서비스를 위한 정책을 정의하고 이를 적용하여 정책 모델을 확장하여 활용할 수 있다. 더 나아가 정책은 온톨로지를 활용하여 정의되기 때문에, 단순히 범용 정책 모델뿐만 아니라 다른 개발자가 정의한 정책을 재활용하여 확장할 수 있다.

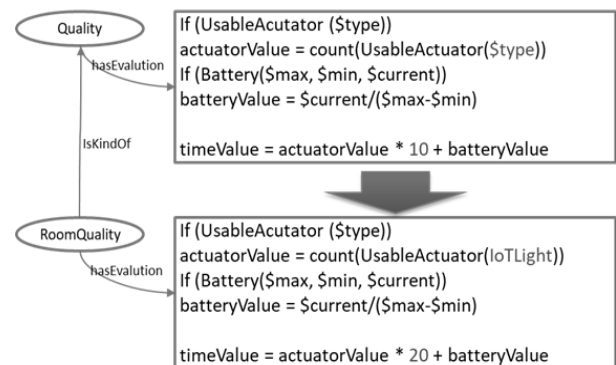


Fig. 5. Expansion of Policy

5. 실험

본 논문에서 제시된 정책 모델과 정책 모델 주도 메타-플래닝이 요구사항을 만족시키는지 검증하기 위하여 다양한 서비스를 활용한 시나리오 기반의 실험을 수행하였다. 이 시나리오는 기업환경 내에서 로봇이 사용자의 요청에 따라 회의실을 준비하는 서비스와 로봇이 지정된 위치로 이동하여 배터리를 충전하는 서비스로 구성되어 있다.

회의실 준비 서비스는 사용자의 선호도에 따라 자연광을 통한 회의실 준비 혹은 조명을 활용한 준비 두 가지 작업을 통해 수행될 수 있도록 구성하였다. 각 작업 계획은 확장된 정책을 활용하여 유틸리티값이 결정되며, 이는 회의실 준비를 요청한 사람의 선호도에 영향을 받도록 설계하였다.

Table 1. Policy of Meeting Room Preparation

Class	Property	Related Class
Resource	isa	Efficiency
UserPreference	isa	Resource
NaturePreference	isKindOf	UserPreference
FluorescentLight Preference	isKindOf	UserPreference
ResourceConservation	isKindOf	UserPreference

배터리 충전 서비스는 로봇 자신의 배터리가 일정 이하로 떨어질 경우, 지정된 장소로 이동하여 충전하는 단순한 과정으로 구성하였다. 이 서비스는 별도의 확장된 정책 모델을 활용하지 않고 범용 정책 모델을 적용하였다.

```

policy NaturePreference {
  precondition {UserNaturePreference($person, $boolean);
  ServiceUser($user); }
  evaluation {
    if ($user = $person && $boolean == true) {
      $value = 100;
    } else {
      $value = 0
    }
  }
}
    
```

Fig. 6. Example of Policy Evaluation

시나리오에 포함된 각 서비스는 범용 정책 모델을 활용한 수식을 통해 유틸리티가 결정되도록 구성하였다. 각 서비스 유틸리티 수식은 다음과 같다.

$$\begin{aligned}
 U_1 &= 0.5 * Resource + 0.5 * Time \\
 U_2 &= 0.3 * Reliability + 0.7 * Safety
 \end{aligned}
 \tag{3}$$

U1 = Utility of Meeting Room Preparation Service
 U2 = Utility of Battery Charging Service

이러한 다양한 도메인 서비스를 포함한 시나리오 수행 과

정을 통해 상황 변화에 따라 메타-플래닝 과정을 수행하여 작업의 유틸리티가 변화하고, 범용 항목을 활용하여 결정된 서비스 유틸리티를 통해 상황에 적합한 서비스를 수행하며, 도메인 서비스를 위해 확장된 정책 모델이 적용되고 활용되는 것을 확인할 수 있었다.

```

Assert context : ServiceUser "Emitsun"
ServiceUser
JAH handled message : agent://www.arbi.com/taskManager (context (ServiceUser "Emitsun"))
find related policy
true
start Evaluate Policy :NaturePreference
Precondition is not satisfied or there is not related policy : NaturePreference
data = (context (UserNaturePreference "Emitsun" 90))
queue
Assert relation : RecievedMessage "agent://www.arbi.com/taskManager" "(context (UserNaturePreference "Emitsun" 90))
JAH received message : agent://www.arbi.com/taskManager (context (UserNaturePreference "Emitsun" 90))
parse : (context (UserNaturePreference "Emitsun" 90))
context
JAH assert context
Assert context : UserNaturePreference "Emitsun" "90"
UserNaturePreference
JAH handled message : agent://www.arbi.com/taskManager (context (UserNaturePreference "Emitsun" 90))
find related policy
true
start Evaluate Policy :NaturePreference
NaturePreference value is 90
policy updated : NaturePreference, 90
send to tm : policy, NaturePreference
(Policy (NaturePreference "MeetingRoomPreparation" "90"))
policy updated :
    
```

Fig. 7. Experiment of Policy Model Driven Meta-Planning

6. 결론

지능형 로봇은 사용자의 요구에 만족하는 서비스를 선택하고 수행할 수 있어야 한다. 이를 위해 현재 수행할 작업을 선택하는 작업관리 기능이 필요하며, 이를 구현하기 위하여 절차적 추론 시스템을 활용하고 있다. 절차적 추론 시스템은 지식 및 상황 정보를 통해 현재 수행할 작업을 추론하는 기능을 제공하며, 작업 내부에 정의된 함수를 통해 결정되는 작업의 우선도를 통해 현재 상황에 가장 적합한 작업을 선택할 수 있다.

하지만 이와 같은 기존의 연구는 시나리오 기반의 정해진 환경에서 하나의 서비스를 수행하는 것을 목적으로 하고 있으며, 지능형 로봇이 서비스를 수행하는 환경은 복수의 도메인 서비스를 동시에 수행하거나, 멀티 도메인 작업 간 우선도 비교를 수행해야 하는 상황이 발생할 수 있다.

이러한 상황에 있어 기존의 방법은 1) 서비스 개발자가 서비스를 개발하는 과정에서 작업의 유틸리티 함수를 작업 모델 내부에서 암시적으로 정의하기 때문에 일반적인 상황에 대해서 적용되는 다양한 서비스에 대해 완벽한 우선순위를 결정할 수 없는 한계가 존재하며, 2) 다양한 서비스가 각각 별도의 함수나 모델을 사용하기 때문에 다수의 서비스 간에 유용성을 비교하는 방법이 범용적이지 않고, 3) 나아가 서비스별로 어느 정도 유용성을 결정하는 방법이 존재하더라도 작업 모델과 결합 되어있어 기존의 방법을 확장하거나 재사용할 수 없다는 한계점을 가지고 있다.

본 논문에서는 도메인 서비스 및 작업의 우선도를 결정하기 위한 정책 모델을 정의하고 정책기반 메타-플래닝을 통해 유틸리티를 계산하는 방법을 제안하였다. 정책 모델은 작업의 유용성을 평가하기 위한 심벌릭 모델로서, 이는 상황 정보의 조합이나 또 다른 정책의 조합 혹은 학습 평가 함수 등을 통해 작업의 유용성 값을 지정하는 기능을 수행한다. 이러한 정책

모델을 활용한 메타-플래닝 과정에서 다양한 컨디션 혹은 외부 함수와의 조합을 통해 작업의 유틸리티를 계산한다, 이 결과로 거시적 관점에서 작업의 우선순위를 결정할 수 있다.

정책 모델은 작업 계획 추론기가 작업의 우선순위를 결정하기 위한 기본적인 척도를 제공하고, 이를 지식 모델로서 구축하여 확장할 수 있도록 구성한다. 이는 서비스 개발자가 작업의 유용성을 명시적인 모델을 통해 적용할 수 있는 방법을 제공하며, 지식 계층 구조를 통해 범용적인 유용성 판단을 위한 함수를 정의하는 방법을 제공하고, 계층을 확장하고 조합하여 새로운 서비스가 적용되기 위한 판단 기준을 확장할 수 있는 체계를 구축하였다.

이러한 정책 모델을 적용하여 우리는 지능체계 내에서 환경의 변화에 따라 작업의 유틸리티가 변화하고 이에 따라 다른 작업을 수행하는 과정을 실험을 통해 확인할 수 있었다.

하지만 현재의 실험에서는 특정 도메인 서비스 간 우선도 비교만을 수행하였기 때문에, 범용성에 대한 추후 개선이 필요하다고 판단된다. 향후 연구에서는 다양한 도메인 서비스를 대상으로 정책 모델을 적용하여 범용 정책 모델의 항목에 대해서 추가적인 검증은 시행할 것이다.

References

[1] M. Beetz, L. Mösenlechner, and M. Tenorth, "CRAM—A Cognitive Robot Abstract Machine for everyday manipulation in human environments," *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp.1012-1017, 2010.

[2] B. Kwak and J. Lee, "Task Management Framework for Intelligent Robots," in *Journal of the Institute of Information and Technology*, pp.21-24, 2006.

[3] M. J. Marcus, "JAM: A BDI-theoretic mobile agent architecture," *Proceedings of the Third Annual Conference on Autonomous Agents*. ACM, 1999.

[4] R. Wilensky, "Meta-Planing: Representing and using knowledge about planning in problem solving and natural language understanding," *Cognitive Science*, Vol.5, Issue 3, pp.197-233, 1981.

[5] M. Sloman and E. Lupu, "Security and management policy specification," in *IEEE Network*, Vol.16, No.2, pp.10-19, 2002.

[6] N. Damianou, N. Dulay, E. Lupu, and M. Sloman, "The ponder policy specification language," *Policies for Distributed Systems and Ntasks*. Springer, Berlin, Heidelberg, pp.18-38, 2001.

[7] M. Sloman, "Policy driven management for distributed systems," *Journal of Network and Systems Management*, Vol.2, No.4, pp.333-360, 1994.

[8] P. C. Fishburn, "Utility theory for decision making," No. RAC-R-105. Research Analysis Corp McLean VA, 1970.

[9] S. Russell and P. Norvig, "Artificial intelligence: A modern approach," Reading, MA: Prentice Hall, pp.480-509, 2009.

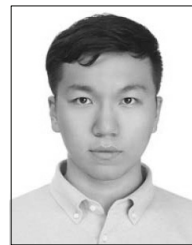
[10] M. Tenorth and M. Beetz, "KnowRob-knowledge processing for autonomous personal robots," *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on. IEEE*, 2009.



최 병 기

<https://orcid.org/0000-0003-4202-2489>
 e-mail : byunggi.choi@gmail.com
 2010년 서울시립대학교
 전자전기컴퓨터공학부(학사)
 2012년 서울시립대학교
 전자전기컴퓨터공학과(석사)

2012년 ~ 현 재 서울시립대학교 전자전기컴퓨터공학과 박사과정
 관심분야 : 인공지능, 소프트웨어 프레임워크, 작업관리



유 인 식

<https://orcid.org/0000-0001-8086-1342>
 e-mail : youin10uosai@gmail.com
 2018년 서울시립대학교 환경공학부(학사)
 2018년 ~ 현 재 서울시립대학교
 전자전기컴퓨터공학과 석사과정
 관심분야 : 인공지능, 지능형 시스템



이 재 호

<https://orcid.org/0000-0002-3332-3207>
 e-mail : jaeho@uos.ac.kr
 1985년 서울대학교 계산통계학과(학사)
 1987년 서울대학교 계산통계학과(석사)
 1997년 University of Michigan(박사)
 1998년 ~ 현 재 서울시립대학교
 전자전기컴퓨터공학부 교수

관심분야 : 인공지능, 지능 로봇