

# 로봇 어플리케이션을 위한 협업 프레임워크

이창묵\* · 권오영\*\*

\*한국기술교육대학교

## The Collaboration Framework for Robot Application

Chang-mug Lee\* · Oh-young Kwon\*\*

\*Korea University of Technology and Education

E-mail : atlantis13@kut.ac.kr\* · oykwon@kut.ac.kr\*\*

### 요 약

로봇 어플리케이션의 활용도는 사회 전반에서 점차 확대되고 있지만 로봇들의 컴퓨팅 자원 차이로 인해 한 로봇에서 다양한 어플리케이션을 실행하기는 힘든 실정이다. 본 논문에서는 로봇이 주변 장치와의 자원 공유를 통해 자원의 제약을 극복하는 프레임워크를 제안한다. 프레임워크는 협업에 요구되는 공통 구성 요소들을 정의한다. 또한 네트워크로 연결된 로봇과 다양한 장치의 예시를 통해 프레임워크의 동작 흐름을 보인다.

### ABSTRACT

The utilization of robot application is growing up in recent years, but there is a constraint to execute various application on the robot because of difference of robot resource. This paper presents the framework in order to solve the resource constraint by sharing resources with other devices near by robot. The framework defines common factors that are needed to collaboration work. Furthermore, We show the working flow of framework with an example consisted of robot and some devices in same network.

### 키워드

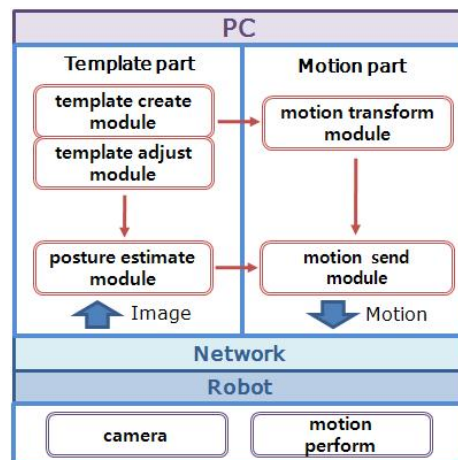
Collaboration Framework, Resource Sharing, Robot Application, Robot Middleware

## 1. 서 론

최근 로봇을 이용하여 다양한 서비스를 제공하는 어플리케이션이 주목받고 있다. 로봇은 점차 소형화, 고성능화 되고 있지만 성능과 비례하는 비용 상승의 문제[1] 등으로 인해 로봇에서 규모 있는 어플리케이션을 실행하기 어려운 경우가 많다. 이에 대한 해결책으로 로봇이 주변의 장치들과 협업을 통해 자원의 한계를 극복하는 모델을 고려해 볼 수 있다. 우리는 이러한 모델의 일환으로 체조 학습 시스템을 개발했다[2][3]. 이 시스템은 그림 1과 같이 각 기능별로 구성된 모듈들로 구성되며 네트워크를 통해 PC와 협력하여 프로그램을 실행한다.

이와 같은 어플리케이션의 개발을 쉽게 하기 위해서는 장치들 간의 협업에 필요한 공통 요소와 동작을 정립한 프레임워크 모델이 필요하다[4]. 본 논문에서는 협업 프레임워크 모델 설계와

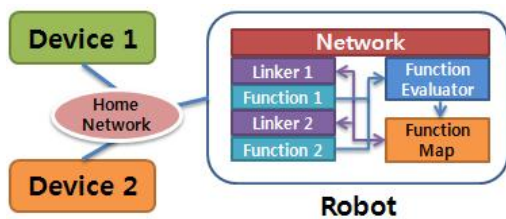
동작 흐름에 관해 기술한다.



(그림 1) 체조 학습 시스템의 구성

## II. 프레임워크의 구성

전체 프레임워크의 구성은 그림 2에 나타나 있다. 디바이스는 어플리케이션을 실행할 수 있는 모든 종류의 컴퓨팅 장치를 나타내며 로봇과 디바이스들은 로컬 네트워크를 통해 연결되어 있다. 디바이스와 로봇에는 그림 2의 우측의 구조로 작성된 동일한 어플리케이션이 각자 탑재되며, 이때 로봇의 어플리케이션은 전체를 실행하는 메인모드로 동작하고 디바이스의 어플리케이션은 로봇에서 요청한 기능을 대신 처리해주는 서브모드로 동작한다.



(그림 2) 프레임워크 구성

프레임워크는 Function, Linker, Function Evaluator, Function Map 으로 구성되며 이들 중 Function Evaluator와 Function Map은 어플리케이션이 메인모드로 실행될 때만 동작한다. 네 가지 구성요소의 상세한 설명은 다음과 같다.

### 1. Function

Function은 전체 어플리케이션을 기능별로 분할했을 때 하나의 기능을 말한다. 즉, 그림 1의 모듈들은 각각 하나의 Function에 대응한다. 하나의 Function은 다시 아래와 같은 요소들로 구성된다. Function들을 구분하는 ID와 어플리케이션의 고유 루틴, 루틴을 실행의 요구사항의 명세, 그리고 루틴을 해당 장치에서 실행했을 때 요구사항 명세를 충족시키는지 판단하여 수행 능력 측정 결과를 기록하는 기능이다. 예를 들어 어플리케이션 루틴이 카메라를 통해 영상을 입력받는다면 요구사항 명세는 카메라의 유무가 되고 현재 시스템에 카메라 자원이 가용한지 측정하게 된다. 측정된 결과는 어플리케이션이 로봇에서 메인모드로 실행중이면 자신의 Function Evaluator로 보내지고, 다른 디바이스에서 서브모드로 실행중이라면 네트워크를 통해 로봇으로 보내진다.

### 2. Function Evaluator

Function Evaluator는 각 Function에서 전송한 수행 능력 측정 결과를 종합하여 평가한다. 하나의 Function에 대해 다수의 디바이스들과 로봇에서 보낸 결과 중 가장 좋은 성능의 보인 결과를 선택하고 이 결과를 보낸 장치가 어디인지를 Function Map에 기록한다. 이후 각 디바이스의 Function에서 상태변동에 따라 새로운 측정 결과

를 보내온다면 재평가 작업을 거친 후 Function Map을 업데이트한다.

### 3. Function Map

Function Map은 장치의 주소와 Function ID로 구성된다. 장치의 주소에는 Function Evaluator가 선정한 측정 결과에 따라 최적 Function이 디바이스의 것이라면 디바이스 IP가, 로봇의 것이라면 로컬로 기록된다. Function ID는 Function을 직접 호출하기 위해 사용되며 장치의 주소와 한 쌍을 이루어 호출시 사용된다.

### 4. Linker

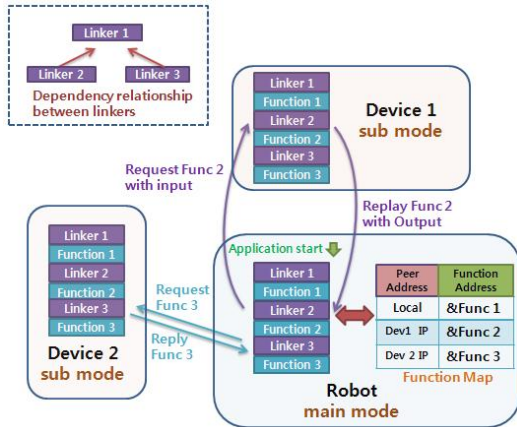
Linker는 Function을 실행하기 위해 호출하는 역할을 한다. 메인모드 동작에서는 어플리케이션의 흐름에 따라 Function과 Function을 이어주는 역할을 하며 서브모드에서는 로봇의 요청을 받아 Function을 실행하고 결과를 로봇으로 전송하는 역할을 한다. 하나의 Linker는 하나의 Function과 연결되어 있고 Linker의 구성요소로는 In/Output buffer, Function dependency가 있다. Function dependency는 Function들 사이에 미리 정의된 종속성에 따라 선행 Function의 종료를 기다리거나 독립적으로 실행함을 결정한다. In/Output buffer는 로봇과 디바이스 사이의 데이터 전송 시 네트워크 버퍼 역할을 하거나 종속성에 따라 Function이 대기 상태에 놓이면 입출력 데이터를 저장하는 버퍼 역할을 수행한다. 마지막으로 Linker는 Function Map을 참조하여 해당 Function에 대해 최상의 성능을 보인 장치의 IP와 Function ID를 사용해 호출한다.

## III. 프레임워크의 동작

로봇과 두 개의 디바이스가 홈네트워크로 연결된 환경에서 프레임워크의 동작은 그림 3과 같다.

세 장치에 동일한 어플리케이션이 탑재되어 있다고 가정하자. 로봇의 어플리케이션이 실행되면 두 디바이스로 Wake-up 메시지를 전달되고 로봇과 디바이스들은 자신의 Function 수행 능력 측정을 위해 각자가 가진 어플리케이션의 Function들을 독자적으로 한번 실행한다. 만약 Function에 카메라의 영상과 같은 특정한 입력이 필요하고 해당 장치에는 자원이 없다면, 네트워크로 연결된 모든 장치에 입력 데이터를 요청하는 메시지를 보낸다. 연결된 모든 장치의 수행 능력 측정 결과가 요구사항을 충족시키지 못하는 Function이 존재한다면 어플리케이션의 실행은 종료된다.

수행 능력 측정이 끝난 디바이스의 어플리케이션은 결과를 로봇에 전송한 후 서브모드로 대기하고 로봇의 Function Evaluator는 결과들을 종합한 후 Function Map을 작성하여 어플리케이션을 진행한다.



(그림 3) 프레임워크의 동작 흐름

그림 3에서 로봇의 Linker 들이 Function Map 을 참조하면 Linker 1은 로봇에서의 실행 성능이 가장 좋고 설정된 dependency가 없으므로 즉시 실행된다. Linker 2는 디바이스 1에 연산을 요청하고 연산 결과를 네트워크를 통해 전송받는다. Linker 3은 dependency 가 Linker 1로 설정되어 있으므로 Linker 2의 처리 완료와 관계없이 즉시 디바이스 2로 연산을 요청한다. 이 경우 Function 이 어디서 처리되느냐에 따라 동작에 차이가 발생한다. Linker 2가 로봇에서 실행된다면 어플리케이션의 순차적 흐름에 따라 Linker 2가 처리완료 될 때까지 Linker 3은 대기하게 되지만, Linker 2가 다른 디바이스에서 실행되는 경우 Linker 3은 즉시 실행된다.

#### IV. 결 론

각종 산업과 엔터테인먼트 분야에서 로봇은 더 이상 희귀한 장치가 아니게 되었으며 로봇을 활용한 어플리케이션의 요구도 높아지고 있다. 하지만 로봇마다 천차만별인 컴퓨팅 자원의 차이로 인해 한 로봇에서 다양한 어플리케이션을 실행하기 힘들다. 이러한 문제를 극복하기 위해 본 논문에서는 주변 컴퓨팅 장치들의 자원을 이용해 어플리케이션의 자원 요구를 충족시킬 수 있는 프레임워크를 제안했다.

프레임워크는 장치들 사이의 협업에 요구되는 공통 구성 요소들을 정의하여 협업 어플리케이션의 개발에 사용될 수 있는 모델을 제공한다. 또한 네트워크 안에서의 로봇과 디바이스에 관한 예시를 통해 전체적인 동작 흐름을 보였다. 향후 프레임워크 라이브러리의 개발과 협업 효율의 향상을 위한 연구를 진행할 것이다.

#### 참고문헌

- [1] Duckki Lee, Tatsuya Yamazaki, Sumi Helal, "Robotics Companions for Smart Space Interactions", IEEE Pervasive Computing, April-June 2009, pp.79-84
- [2] Chang-Mug Lee, Oh-Young Kwon, "Posture Recognition for Physical Training System with Adjusted Edge Template", The 31th KIPS Spring Conference 2009, pp.110-113.
- [3] Chang-Mug Lee, Oh-Young Kwon, "Automatic Motion Creating from the Posture Template for the Robot Physical Training System", ICHIT 2009
- [4] Jin Nakazawa Tokuda, H. Edwards, W.K. Ramachandran, U, "A Bridging Framework for Universal Interoperability in Pervasive Systems", Distributed Computing Systems, 2006