

# 역할기반 적응형 분산 시스템을 위한 조직 프레임워크

황성윤\*, 정종윤\*, 류기열\*\*, 이정태\*\*\*

\*아주대학교 대학원 컴퓨터공학과

\*\*아주대학교 정보컴퓨터공학과

\*\*\*아주대학교 소프트웨어융합학과

e-mail:ctrss@ajou.ac.kr

## An Organization Framework for Role-based Adaptive Distributed System

Seong-Yun Hwang\*, Jong-yun Jung\*, Ki-yeol Ryu\*\*, Jung-tae Lee\*\*\*

\*Dept. of Computer Engineering, Ajou University

\*\*Dept. of Information and Computer Engineering, Ajou University

\*\*\*Dept. of Software Convergence Technology, Ajou University

### 요 약

분산 응용 시스템의 컴퓨팅 환경이 유비쿼터스 네트워크 환경으로 바뀌면서 기존의 개방성 및 이질성과 같은 특성 외에 적응성이 중요한 이슈로 부각되었다. 적응성을 지원하기 위해 분산 응용 시스템은 컴포넌트를 동적으로 찾아 추가하거나 교체하고 이들 간의 상호작용을 지원해야 한다. 최근 이 같은 적응적 상호작용을 위한 역할기반 분산 시스템 모델이 제안되었다. 역할기반 분산 시스템 모델은 응용을 추상적인 역할들로 이루어진 조직으로 보고, 조직을 구성하는 역할과 실제 행위를 수행하는 행위자를 분리한다. 실행 시 응용은 행위자를 역할에 바인딩하고, 행위자는 수행할 역할 업무를 동적으로 응용으로부터 제공받아 실행함으로써 적응성을 높여주는 모델이다. 하지만 이런 역할기반의 분산 시스템 모델을 위한 실행환경의 구축에 대한 연구는 미비한 상태이다. 본 논문에서는 역할기반 분산 시스템 모델의 실행환경으로 응용 프레임워크를 제안한다. 응용 프레임워크는 응용에 필요한 역할들로 이루어진 조직을 구성하고 관리하는 조직 프레임워크와 행위자의 역할 수행과 역할 프로세스 관리를 수행하는 행위자 프레임워크로 나누어진다. 행위자 프레임워크에 대해서는 이전 연구에서 제안하고 있고, 본 논문에서는 이와 결합되는 조직 프레임워크를 구현하기 위해 조직 프레임워크의 요구사항을 분석하고 설계한다.

### 1. 서론

최근 주목받는 유비쿼터스 시스템(ubiquitous system)과 모바일 시스템(mobile system)은 다양한 컴퓨팅 개체들이 동적으로 참여하고 상호동작하는 분산 시스템이다. 이들의 발달로 분산 응용 시스템은 구성요소들과 그들 간의 관계가 비결정적(non-deterministic)인 특성을 가진다 [1]. 이런 비결정적 구성요소를 사용하기 위해 분산 응용 시스템의 실행시간에 적응성을 높이기 위한 연구가 활발히 이루어지고 있다.

적응성을 지원하기 위한 역할기반(role-based)의 분산 시스템에 대한 연구인 ROAD(Role-Oriented Adaptive Design)[2]에서는 각 컴퓨팅 개체들을 시스템 조직을 구성하는 역할(role)과 역할을 수행하는 행위자(player)로 분리하여 실행시간에 동적으로 연결하여 각각의 컴퓨팅 개체와는 독립적으로 조직(organization)을 생성하거나 변경할 수 있게 하였다. 기존의 역할 기반 연구에서는 시스템의 설계 단계에서만 역할과 행위자를 분리하여 실제 실행시간에서 적응성 확보가 미흡했다. ROAD에서는 설계 단계 뿐만 아니라 실행시간에서도 역할과 행위자를 분리하여

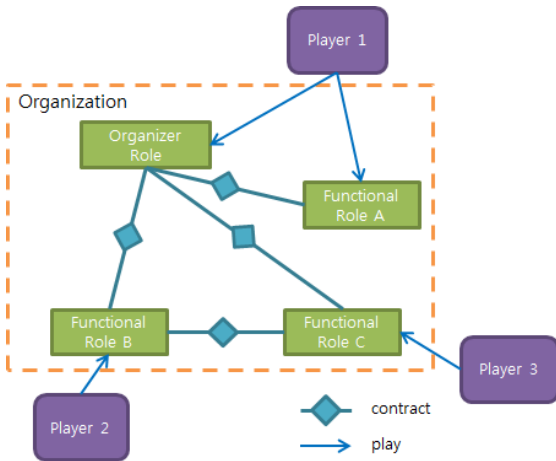
이를 보완하고 있다. 하지만 구현 단계에서 시스템을 실제로 구현하고 배치하기 위한 방안에 대해서는 고려하지 않고 있다.

ROAD의 모델에서는 행위자가 수행하는 역할에 대한 업무(process)를 행위자 내부에 정의하여 행위자는 사전에 정의된 역할의 업무만을 수행한다. 이렇게 행위자가 미리 정의된 기능만을 수행하는 방법은 높은 수준의 적응성을 요구하는 분산 응용 시스템에는 적합하지 못하다. 이전 연구에서는 ROAD에서 행위자가 수행하는 개별 역할의 업무를 역할 프로세스(role process)로 정의하고 역할 외부에 역할 프로세스를 두어 해당 역할의 행위자가 실행시간에 동적으로 역할 프로세스를 할당받아 수행하도록 하는 분산 시스템 모델을 제안하고 이를 행위자 측면에서 구현하기 위한 행위자 프레임워크를 제시하고 있다[3]. 하지만 이 모델을 토대로 실제 분산 응용 시스템을 구현할 때는 개별 행위자 측면에서의 구현 외에 조직을 구성하고 관리하는 조직 측면에서의 구현도 중요한 구현 이슈이다.

본 논문에서는 적응형 분산 시스템 모델을 구현하기 위한 응용 프레임워크를 제안한다. 응용 프레임워크는 조

직을 구성하고 관리하는 조직 프레임워크와 행위자의 역할 수행과 행위자에 할당된 역할 프로세스의 관리를 위한 행위자 프레임워크로 나눌 수 있다. 현재 응용 프레임워크를 구현하기 위해서는 행위자 프레임워크와 결합되는 조직 프레임워크의 구현이 필요하다. 본 논문에서는 조직 프레임워크와 행위자 프레임워크의 상호작용을 분석하여 조직 프레임워크의 요구사항과 기능을 정의하고 각 기능을 수행하기 위한 조직 프레임워크의 구조를 설계하여 구현 방안을 제시하고자 한다. 2장에서는 ROAD의 역할기반 모델과 이전 연구에서 제시한 역할기반 적응형 분산 시스템 모델과의 차이점을 설명한다. 3장에서는 이러한 적응형 분산 시스템 모델을 구현하기 위한 응용 프레임워크의 전체 구성을 보여주고 그 중 한 부분인 조직 프레임워크의 요구사항을 분석하여 설계한다. 마지막으로 4장에서는 이 프레임워크에 대한 고찰과 향후 연구의 방향을 제시한다.

**2. 역할기반 적응형 분산 시스템 모델**



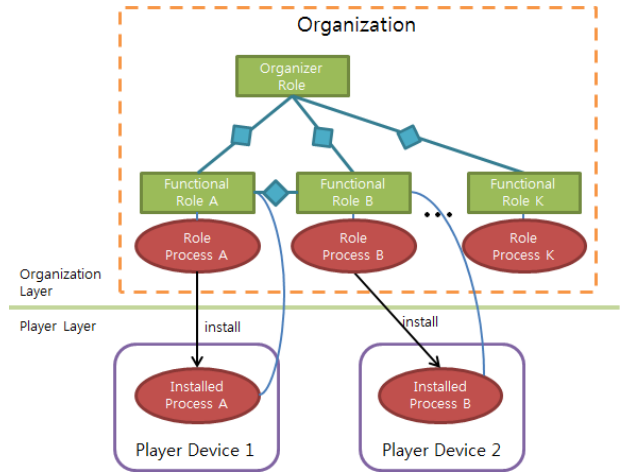
(그림 1) ROAD의 역할 모델

ROAD는 그림 1과 같이 응용 시스템을 하나의 조직으로 바라본다. ROAD에서 조직은 이를 구성하는 노드인 역할과 역할 사이의 관계인 계약으로 이루어져 있다. 그림에서 볼 수 있듯이 역할은 조직을 구성하기 위한 주체 역할인 조직자 역할(organizer role)과 일반적인 역할의 기능을 수행하는 기능적 역할(functional role)로 구분된다. ROAD는 역할과 행위자(player)의 관계, 역할과 역할의 관계를 직접적으로 연결하지 않고 추상적으로 연관시켜 조직의 구성 변경이나 행위자의 교체, 수정에 적응성을 높이고 있다.

ROAD는 역할이 수행하는 기능인 역할의 업무를 행위자 내부에 배치하거나 역할 내부에 배치하는 두 가지 방법으로 분류하고 있다. 역할의 업무가 역할 내부에 배치될 경우, 역할은 행위자와 동일한 환경에서 실행되어 역할이 행위자 환경에 대해 고려해야 하기 때문에 적응성을 높이기 위한 ROAD의 목적을 달성하지 못한다. 이런 이유로 ROAD는 행위자 내부에 프로세스를 배치하는 방법을 채

택하고 있다. 하지만 이는 각 행위자가 사전에 정의된 역할만을 수행할 수 있기 때문에 행위자의 다양한 역할 수행에 장애가 된다.

이러한 ROAD의 모델을 개선하기 위해 역할 프로세스를 역할의 외부에 정의해두고 실행시간에 행위자가 역할에 연결되면 이 역할 프로세스를 행위자 환경에서 내려받아 수행하는 그림 2와 같은 역할기반 적응형 분산 시스템 모델이 제안되었다[3]. 역할 프로세스란 역할에서 수행하는 업무만을 따로 분리한 것으로 각 역할은 맡은 행위자는 역할의 업무를 실행시간에 동적으로 할당받아 수행하게 된다. 이를 통해 행위자는 내려받는 업무에 따라 다양한 역할을 수행할 수 있고, 역할이 수행하는 업무가 변경된 경우에는 이 역할을 가지는 전체 행위자를 수정하지 않고 역할 외부에 정의된 역할 프로세스만을 수정하여 변경 사항을 적용할 수 있다.



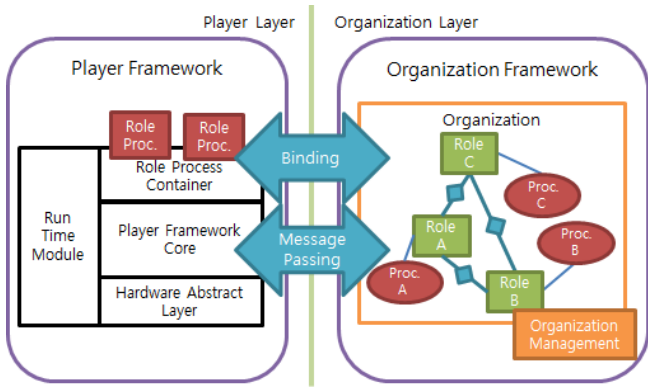
(그림 2) 역할기반 적응형 분산 시스템 모델

**3. 조직 프레임워크**

이 장에서는 위의 모델을 적용하는 응용 프레임워크에서 행위자 프레임워크와 조직 프레임워크와의 관계를 살펴보고, 조직 프레임워크의 요구사항을 분석하고 설계하여 조직 프레임워크를 실제 구현할 수 있도록 한다.

**3.1 응용 프레임워크의 개요**

역할기반 적응형 분산 시스템 모델을 적용하는 응용 프레임워크를 구현하기 위해서는 행위자 프레임워크와 조직 프레임워크의 구현이 필요하다. 행위자 프레임워크에서는 조직의 구성과는 관계없이 하드웨어 추상화와 역할 프로세스 관리 등의 실제 행위자 내의 실행시간에 필요한 모듈만을 가지고, 조직을 실제 구성하고 있는 조직 프레임워크에서는 행위자와의 연결 관리 외에도 조직의 구성과 조직 전체의 실행시간에서 필요한 모듈까지 포괄적인 모듈이 필요하다. 다음 절에서 구체적인 조직 프레임워크의 기능적 요구사항을 분석한다. 그리고 행위자 프레임워크와 조직 프레임워크는 그림 3에서처럼 역할과 행위자 간의



(그림 3) 행위자 프레임워크와 조직 프레임워크의 상호작용

바인딩과 실행시간의 메시지 전송(message passing)을 통해 상호작용을 수행하게 된다.

### 3.2 조직 프레임워크의 요구사항

적용형 분산 시스템 모델을 적용하기 위한 조직 프레임워크는 다음과 같은 요구사항을 만족해야 한다.

- 조직의 생성 및 관리: 조직 프레임워크에서는 조직을 구성하는 역할과 역할간의 관계를 기술하는 계약의 인스턴스를 생성하고 관리하는 모듈이 필요하다. 조직은 상황에 따라 변경되기도 하기 때문에 역할과 계약을 변경하는 모듈 역시 필요하다.
- 행위자 기능 확인 및 역할과의 연결: 조직 프레임워크에서는 행위자를 구분하기 위해 행위자가 가능한 기능 목록을 저장하고 이를 통해 가능한 역할을 연결시킬 수 있어야 한다. 행위자는 행위자 내부에서 수행할 수 있는 역할을 정의하지 않기 때문에 행위자 자신이 할 수 있는 기능적 요소를 서버에 등록하여 그 기능으로 수행할 수 있는 역할과 연결될 수 있도록 해야 한다.
- 역할 프로세스의 할당: 역할 외부에 정의된 역할 프로세스는 역할과 행위자가 연결될 때 행위자에게 할당된다. 그리고 역할 프로세스가 변경될 경우, 해당 역할을 맡고 있는 행위자에게 이를 재할당하는 과정이 필요하다.
- 실행 시간의 메시지 통신: 응용을 실행하는 동안 역할과 역할, 역할과 행위자 사이의 데이터 전송이나 조직의 구성, 변경과 관련된 메시지를 전달하게 된다. 조직 프레임워크에서는 이를 수행하기 위한 모듈이 필요하다.

### 3.3 조직 프레임워크의 설계

이 절에서는 앞에서 언급한 적용형 분산 시스템 모델을 적용하기 위한 조직 프레임워크를 설계하고 필요한 라이브러리를 분석한다. 조직 프레임워크의 구조는 그림 4와 같다.

1) 실행환경 모듈(Run Time Module): 실행환경에서는 조직 프레임워크의 시작 및 초기화부터 종료까지의 전반적인 요소를 관리한다. 이 실행환경 모듈에서 프레임워크 코어 모듈을 불러오는 조직 프레임워크의 시작에서부터,

응용의 초기화, 실행, 종료 단계를 상태별로 관리하고 이를 위한 환경을 구성한다.

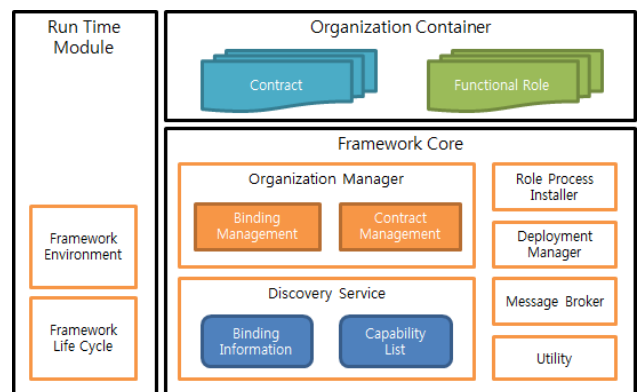
2) 조직 관리자(Organization Manager)와 조직 컨테이너(Organization Container): 조직 관리자는 응용의 초기화 단계에서 조직 컨테이너를 구성하는 역할과 계약의 인스턴스를 생성한다. 그리고 응용의 실행 단계에서 조직 관리자는 조직의 변경 관리, 역할과 행위자의 연결 관리를 하게 된다. 조직 컨테이너는 응용의 실행 단계에서 역할과 계약을 통하는 메시지에 대한 인터페이스 기능을 한다.

3) 발견 서비스(Discovery Service): 조직 관리자가 실행 단계에서 역할과 행위자의 연결을 관리할 때 발견 서비스 내에 행위자가 등록한 기능 정보를 이용하게 된다. 이 기능 정보를 통해 각 행위자에 대한 역할의 연결 가능성을 확인하고 역할과 행위자를 연결한다. 발견 서비스에서는 이렇게 연결된 연결 정보를 저장하여 역할과 행위자의 메시지 통신에 사용할 수 있도록 한다.

4) 역할 프로세스 설치자(Role Process Installer)와 배치 관리자(Deployment Manager): 역할과 행위자가 연결되면 배치 관리자에서 해당 역할의 역할 프로세스를 불러와 가지고 온 역할 프로세스를 역할 프로세스 설치자가 행위자 환경에 설치한다. 역할 프로세스 설치자는 행위자에게 이미 설치된 동일한 역할 프로세스가 있을 경우와 조직 내의 역할 프로세스가 변경되었을 경우를 고려하여 역할 프로세스를 설치한다.

5) 메시지 브로커(Message Broker): 조직 프레임워크의 실행 단계에서는 조직의 구성 변경이나 행위자의 데이터 전송 등 다양한 메시지가 전송된다. 이 메시지들을 메시지 브로커에서 수집하여 메시지를 분류하고 적절한 프레임워크 모듈에 메시지를 전송한다.

6) 유틸리티(Utility): 조직 프레임워크에서는 서버에서 사용하는 데이터 저장과 통신, 메모리 관리 등 다양한 자원에 대한 유틸리티를 제공한다.



(그림 4) 조직 프레임워크 구조

## 4. 결론 및 향후 연구

분산 컴퓨팅은 현재 유비쿼터스 컴퓨팅, 자율 컴퓨팅, 클라우드 컴퓨팅 등의 다양한 패러다임으로 연구되고 있다. 이러한 패러다임들은 분산 시스템의 적응성을 중요한

요소로서 사용하고 있다. 이처럼 적응성을 지원하는 다양한 프레임워크와 플랫폼들의 연구가 확산되고 있다[4][5]. 본 논문에서는 분산 시스템에서의 적응성 확보를 위해 역할 기반의 ROAD 모델에서 역할 프로세스를 정의하고 이를 적용한 적응형 분산 시스템 모델을 구현하기 위한 응용 프레임워크를 제안한다.

본 연구에서 설계한 조직 프레임워크는 이전 연구에서 설계된 행위자 프레임워크와 결합되어 분산 응용 시스템에서 적응형 분산 시스템 모델을 구현하기 위한 응용 프레임워크를 완성한다. 적응형 분산 시스템 모델은 ROAD의 적응성 문제를 최소화하여 ROAD보다 높은 적응성을 보여주고 있다. 그리고 구현 단계를 위한 프레임워크를 개발함으로써 시스템의 실제 구현과 배치 방안에 대해 보여주고 모델의 적용 가능성을 입증한다.

이후 본 연구에서는 조직 프레임워크의 프로토타입을 구현하고 행위자 프레임워크와 결합하여 응용 프레임워크를 완성하고 이를 활용한 실제 분산 응용 시스템을 개발하고자 한다. 향후에는 프레임워크를 다양한 도메인에 적용하여 프레임워크의 유용성을 검증하고 다양한 도메인에 대해 구체적으로 평가하는 것이 필요하다.

### 참고문헌

- [1] Minh H. Tran, Jun Han, "Social Context, Supporting Interaction Awareness in Ubiquitous Environments," *MobiQuitous 6th Annual International*, 2009.
- [2] Alan Colman, "Role-Oriented Adaptive Design," *Swinburne University of Technology, PhD Thesis*, 2006.
- [3] 홍상준, 류기열, 이정태, "역할 기반의 분산 시스템 구현을 위한 컨테이너 프레임워크", *한국정보기술응용학회, 2011년도 춘계학술대회, 정보기술 융합과 미래의 신성장 동력*, pp. 231-236. 2011년 6월.
- [4] Andreas Frei, Gustavo Alonso, "A Dynamic Lightweight Platform for Ad-hoc Infrastructures," *Proceedings of the 3rd IEEE Int'l Conf. on Pervasive Computing and Communications 2005*.
- [5] Vaidy Sunderam, Dawid Kurzyniec, "Lightweight Self-organizing Frameworks for Metacomputing," *Proceedings of the 11th IEEE International Symposium on High Performance Distributed Computing HPDC-11*, 2002.