

논문 2008-45CI-5-10

# 다자간 협업 환경에서 에이전트 기반 서비스 합성

## ( Agent-based Service Composition in Multi-party Collaboration Environments )

한 상 우\*, 김 중 원\*\*

( Sangwoo Han and JongWon Kim )

## 요 약

지역적으로 분산되어 있는 지식 노동자들 간에 진보된 협업을 지원하기 위해서, 유비쿼터스 컴퓨팅의 범위 내에서 많은 연구가 진행되어 왔다. 특히 전통적인 회의실 기반의 협업 환경 (편리하지 못한 시청각 자료의 공유, 협업 도구의 어려운 조작 등)에서 이미 알려진 여러 문제점 들을 해결하기 위해, 개념적인 프레임워크들이 설계되었다. 실용적이고 편리한 협업 환경 구성을 위해, 본 논문에서는 에이전트 기반의 서비스 합성 모델을 개념적으로 설계한다. 사용자 입장에서 복잡한 협업 환경의 내부 구조를 알 필요 없이, 상위 수준의 작업 지시를 하면 협업 환경에서 주어진 작업을 수행하기 위한 서비스들을 자동적으로 발견하고, 연결하기 위한 모델을 설계한다. 본 서비스 합성 모델을 기반으로 다중 에이전트 시스템 기반의 다자간 협업 환경 관리 툴킷을 개발하여, 직관적인 GUI를 통해 협업 노드 상의 다양한 서비스들을 설정, 운용하는 방법을 제공하고, 주어진 작업을 수행하기 위해 필요한 서비스들을 발견하고 연결하는 기능을 제공한다. 구현된 툴킷의 실용성을 실증하기 위하여, 개발된 툴킷을 기반으로 영상 회의를 위한 기능을 제공하는 협업 서비스들을 구현한다.

## Abstract

To support advanced collaboration among knowledge workers distributed geographically, there have been extensive researches under the scope of ubiquitous computing environments. Especially, to cope with several known problems in traditional room-based collaboration environments such as uncomfortable sharing of visuals and documents and difficult operation of collaboration tools, several conceptual frameworks are designed and prototyped. Focusing on practical and interactive collaboration with remote nodes, in this paper, we conceptually design an agent-based service composition model for multi-party collaboration environments. The proposed model is designed to automatically discover and combine services to achieve given tasks in a collaboration environment by using high-level user commands (without the knowledge of internal architecture). Based on the service composition model, we develop a multi-agent-based management toolkit for multi-party collaboration environments. It provides easy-to-use GUI to operate various services in an environment and perform service composition algorithm to discover appropriate services and combine them. To explore the possibility of the toolkit, we implement collaboration services to support video conference by using the toolkit.

**Keywords :** 다자간 협업 시스템, 에이전트 기반 서비스 합성, 서비스 지향 컴퓨팅, 멀티 에이전트 시스템, 미들웨어.

## I. 서 론

고성능 컴퓨팅 및 네트워킹 기술은 기업체 및 연구

기관을 중심으로 진보된 협업 시스템의 보급을 확산시켜 왔다. 특히, 상용 시스템을 이용한 비즈니스 회의, 중업원 교육을 비롯해 연구 지향적인 협업 시스템을 이용한 과학적 공동 실험, 원격 강의 등 다양한 응용분야에서 폭 넓게 활용되고 있다. 대표적인 상용 협업 시스템인, Cisco의 TelePresence와 HP의 Halo는 다중 디스플레이 장치 및 고해상도 카메라를 이용하여 원격지의 지식 노동자들 간에 현장감 있는 원격 회의 환경을 제공하고 있다. 또한 아르곤 국립 연구소의 Access Grid와

\* 학생회원, \*\* 종신회원, 광주과학기술원 정보기전공학부 (School of Information and Mechatronics, Gwangju Institute of Science and Technology)

※ 본 논문은 21세기 프론티어 연구개발사업의 일환으로 추진되고 있는 지식경제부의 유비쿼터스컴퓨팅 및네트워크원천기술개발사업의 지원(UCN08B3-O2-10M)에 의한 것임.

접수일자: 2008년8월20일, 수정완료일: 2008년9월8일

마이크로소프트 리서치의 ConferenceXP는 멀티캐스트로 연결된 다수의 협업 공간들 간에 음성/영상/데이터 공유 서비스를 제공할 수 있는 협업 환경을 구성하며, 사용자들이 용도에 맞게 확장된 협업 환경을 구축하도록 추가적인 서비스들을 개발할 수 있는 프레임워크를 제공하고 있다.

이와 같은 진보된 협업 시스템의 하나로써, SMeet (Smart Meeting Space) 다자간 상호작용 협업 환경이 개발되었다<sup>[1]</sup>. SMeet는 고품질의 영상, 음성, 그래픽스의 공유뿐만 아니라, 다중 포인팅 및 위치 추적 장치를 이용하여 상호작용 가능한 초고해상도 격자형 디스플레이를 제공함으로써, 보다 현장감있는 그룹형 공동작업 환경을 제공하고 있다. 그러나 이와 같은 형태의 협업 환경은 수많은 하드웨어 및 소프트웨어들로 구성되기 때문에, 협업 시스템의 조작에 익숙하지 않은 사용자가 일일이 모든 구성요소들을 직접 조작하는 일은 어려울 뿐만 아니라 비효율적이다. 이러한 이유로, 협업 환경의 관리 및 운용 측면에서 사용자 편의성을 향상시킬 수 있는 방법에 대한 연구는 중요하다고 할 수 있다.

협업 환경에서 구성 요소들의 관리를 편리하게 해주는 추상화 기술은 여러 연구진들에 의해 연구, 개발되고 있다. GAIA<sup>[2]</sup>에서는 각 구성요소를 Active Space API (application programming interfaces)를 통해 고유 기능에 접근할 수 있도록 만든 후, 객체지향 프로그래밍에서 널리 활용되는 MVC (model, view, controller) 모델을 확장한 응용 프레임워크를 통해 사용자가 구성 요소들 간에 상호작용을 프로그래밍 할 수 있는 소프트웨어 구조를 개발하였다<sup>[3]</sup>. 또한 Easy Meeting<sup>[4]</sup>에서는 주어진 회의 시나리오에 맞게 각 서비스 에이전트들 간에 상호작용을 워크플로우로 정의하여, 설정된 방식에 따라 서비스 에이전트들이 유기적으로 협력하여 서비스를 제공하는 기술을 개발하였다. 또 다른 연구로써, Access Grid 노드를 구성하는 분산된 멀티미디어 하드웨어 및 소프트웨어를 파일 시스템처럼 계층적으로 관리하기 위한 추상화 기능을 제공하는 INDIVA는 상위 수준의 명령어를 통해서 모든 자원들을 발견, 연결, 조작할 수 있는 미들웨어를 구현하였다<sup>[5]</sup>. 그러나 이러한 방법들은 이용하기 위해서는, 수많은 구성요소들 중에서 사용할 서비스들의 선택, 서비스들의 연결, 서비스들의 상호작용 순서를 결정하는 절차를 사용자가 사전에 정해야 한다. 이러한 제약으로 인해 참여 노드들의 실질적인 성능 특성, 시간에 따라 변화하는 가용자원 상태, 서로 다른 사용자 의도 등 가변적인 협업 문맥에 동

적으로 대응하기 어렵다는 단점이 있다.

따라서 성공적인 협업 환경을 구성하기 위해서는 참여자들이 협업 환경의 구성 및 운용에 많은 노력을 기울이지 않고 공동 작업을 할 수 있도록 의도된 협업 요구사항들을 자율적으로 만족시키는 것이 매우 중요하다<sup>[6]</sup>. 이를 위해, 분산된 하드웨어 및 소프트웨어들의 상태 정보 (문맥)를 용이하게 수집/관리할 수 있고, 이들의 기능을 표준화된 방법에 따라 통제할 수 있는 방법이 마련되어야 한다. 또한 협업 환경의 문맥 (예, 사용자 기호, 가용 자원 상태, 달성되어야 할 협업의 임무 등)에 따라 복합적인 협업 응용 서비스(즉, 단일 서비스로는 제공할 수 없으나 다수의 서비스들이 협동하여 제공할 수 있는 확장된 서비스)를 제공하기 위한 협업 환경의 자율적 구성 및 관리 방법이 고려되어야 한다.

위와 같은 요구사항을 해결하기 위해, 본 논문에서는 서비스 지향 구조 (service-oriented architecture)에 기초한 에이전트 기반의 서비스 합성 모델을 제안한다. 기본적인 접근방법은 다음과 같다. 먼저, 기존 협업 소프트웨어의 데이터 및 인터페이스를 네트워크로 접근 가능하도록 만들기 위하여 컴포넌트 기반의 소프트웨어인 서비스로 만든 후, 협업 환경에서 관리 가능한 구성요소로 등록시킨다. 다음으로, 복합적인 협업 응용을 제공하기 위한 서비스 합성 알고리즘을 통해, 서비스 요구사항에 맞는 이용 가능한 서비스들을 발견하고 이들을 서로 연결함으로써 주어진 협업 응용을 제공하기 위한 서비스들의 상호작용을 돕는다.

제안하는 서비스 합성 모델은 다음과 같은 특징을 가진다. 첫째로, 서비스들은 상호 연결되어 전체적인 협업 공간의 정보를 (온톨로지를 이용한) 정형화된 형식에 따라 공유할 수 있다. 이는 협업 환경이 각 서비스들이 관심있는 서비스들의 상태 정보를 공유하고, 이를 통해 협업 맥락에 따른 서비스 간 상호작용이 가능하게 된다. 둘째로, 복합적인 협업 응용의 구체적인 내용에 따라 이용 가능한 서비스들을 합성함으로써 협업 환경의 운용 및 관리를 위한 사용자 개입을 최소화하면서, 주어진 목적에 맞게 협업 환경을 자동적으로 구성할 수 있다. 제안하는 구조의 가능성을 검증하기 위하여, SMeet 시스템에서 서비스들을 추상화하고, 주어진 협업 응용에 따라 서비스를 합성하기 위한 SMeet 관리 툴킷을 개발한다. 다중 에이전트 시스템 미들웨어인 JADE (Java Agent DEvelopment Framework)<sup>[7]</sup>를 기반으로 본 툴킷 및 관련 서비스들을 구현한 후, 주어진 협업 응용 서비스 시나리오 (예, 영상 회의)에 따라 결

표 1. 다자간 협업 환경을 위한 서비스 합성 구조의 개념적 모델

Table 1. Conceptual model of service composition architecture for multi-party collaboration environment.

기능적 구분 추상화 수준	상호작용 모델	환경 모델	표현 모델	응용 모델	데이터 모델
작업 계층 (Task layer)	공유 객체별로 특화된 조작을 위한 상호작용 인터페이스	작업과 연관된 공유 객체들을 처리하기 위한 서비스 관리자들의 정보	주어진 작업의 진행 상황 및 작업과 관련된 공유 객체들의 상태 표현	작업과 연관된 공유 객체들의 조작 방법	작업과 연관된 공유 객체들에 대한 데이터
서비스 관리자 계층 (Service manager layer)	서비스 관리자의 운영 방식 및 협력할 대상 서비스 관리자를 지정하기 위한 인터페이스	주어진 공유 객체를 처리하기 위한 서비스 정보	주어진 공유 객체를 처리하기 위한 서비스 연결 상태 표현	주어진 공유 객체를 처리하기 위한 서비스들의 합성 논리 제공	합성된 서비스로 부터 수집된 통합 데이터
서비스 계층 (Service layer)	서비스의 시작, 설정, 종단을 위한 인터페이스	협업 툴이 제공하는 능력 및 설정 정보	서비스에 대한 상태 정보 표현	서비스가 제공하는 협업 기능	서비스의 운용 상태 및 수행 데이터

과를 검증한다.

본 논문의 구성은 다음과 같다. 다음 절에서는 다자간 협업 환경을 위한 서비스 합성 모델과 각 요소별 명세에 대해서 설명하고, 이를 구현하기 위한 에이전트 기반의 소프트웨어 구조 설계 및 서비스 합성 알고리즘을 제안한다. 이 후, 구현 결과를 설명한 뒤, 결론으로 본 논문을 맺는다.

## II. 다자간 협업 환경을 위한 서비스 합성 모델

본 절에서는 다자간 협업 환경에서 복합적인 응용 서비스를 제공하기 위한 서비스 및 서비스 합성 모델을 개념적으로 제시하고, 각 구성요소를 기술하는 방법을 설명한다.

### 1. 서비스 합성을 위한 개념적 모델

다자간 협업 환경을 위한 서비스 합성 구조를 설명하기 위하여, BEACH 응용 모델 및 소프트웨어 프레임워크<sup>[6]</sup>를 활용하여 제안하는 서비스 합성 구조의 개념적인 모델을 표 1과 같이 기술한다. 표 1에 나타난 기능적 구분은 다음과 같이 정의된다. 기능적 구분에서 데이터 모델은 사용자가 만들거나 상호작용할 수 있는 데이터의 종류를 뜻하며, 응용 모델은 데이터 모델의 조작에 필요한 기능을 정의한다. 환경 모델은 협업 환경에서 기능 수행에 영향을 미치는 여러 가지 요인들에 대한 정보를 기술하며, 표현 모델은 사용자가 데이터 모델의 상태를 보여주기 위한 요소를 나타내고, 상호작용 모델은 응용 모델과 연계하여 데이터 모델의 상태를 변경하

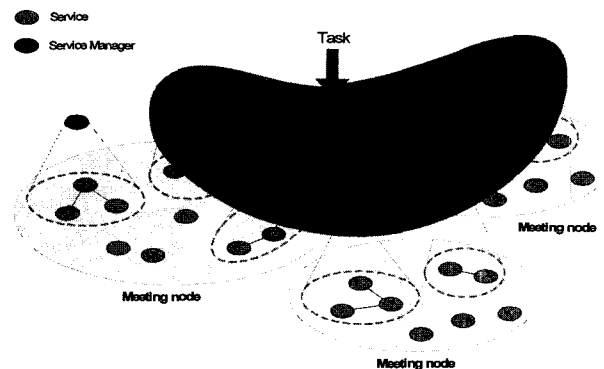


그림 1. 서비스 합성 모델의 계층적 구조  
Fig. 1. Hierarchical structure of service composition model.

기 위한 방법을 결정한다<sup>[8]</sup>. 추상화 수준은 작업 계층, 서비스 관리자 계층, 서비스 계층으로 구성되며, 그림 1과 같이 서비스 합성 모델을 위한 계층적 구조로 표현된다. 작업(task)에서 원하는 복합적인 협업 응용 서비스의 특성을 기술하고, 관련된 서비스 관리자(service manager)들에게 이를 전달함으로써, 서비스(service) 구성 및 운용에 관한 책임을 부여한다. 전달된 작업 내용에 입각하여, 서비스 관리자는 사전에 준비된 서비스 합성 논리를 기반으로, 필요한 서비스들을 발견하고 실행 순서에 따라 연결한 후, 통합적으로 실행한다.

#### 가. 작업 계층

작업 계층에서는 협업 환경에서 필요한 응용 서비스 (예를 들면, 영상회의, 공동문서 작업, 가시화 콘텐츠 공유 등)에서 공유되어야 할 콘텐츠를 사용자 친화적인 방법으로 다루기 위한 추상화된 기능을 제공한다. 여기

서 공유되어야 할 콘텐츠를 공유 객체(shared object)라고 부른다. 이 계층에서 데이터 모델은 작업과 연관된 공유 객체들에 대한 정보를 기술한다. 예를 들어 발표자와 관련된 콘텐츠들을 공유 객체라고 가정한다면, 발표자 모습을 바라보는 시점, 발표자의 모습에 포함되는 배경 정보, 발표자의 음성, 발표 자료, 손동작을 이용한 가상 물체와의 상호작용 등이 공유 객체의 데이터 모델로 표현될 수 있다. 응용 모델에서는 공유 객체를 조작(예를 들면, 발표자를 바라보는 시점 변경)하기 위한 방법을 제공한다. 표현 모델에서는 주어진 작업에 특화된 GUI를 통해 관련된 공유 객체들의 상태를 표시하는 방법을 정의한다. 즉, 주어진 작업과 관련된 모든 콘텐츠들의 상태를 표현한다. 환경 모델에서는 공유 객체들을 사용자의 요청에 맞게 처리하기 위해서 이용할 수 있는 서비스 관리자들의 정보를 기술한다. 상호작용 모델에서는 공유 객체들을 조작하기 위한 특화된 인터페이스를 기술한다. 예를 들면, 발표자의 자료를 원하는 위치에 배치시키는 시각화 공유 (visual sharing) GUI 등을 고려할 수 있다.

#### 나. 서비스 관리자 계층

주어진 작업에서 요구하는 공유 객체들을 처리하기 위해서는, 공유 객체의 처리 요구사항과 협업 환경의 상황의 이해를 기반으로 관련된 서비스들을 자동적으로 선택하고 이들에게 적절한 행위를 주문할 수 있는 상위 수준의 관리 기능이 필요하다. 이와 같이 주어진 공유 객체를 처리하기 위한 상위 수준의 관리 개체를 서비스 관리자라고 부른다. 서비스 관리자는 미디어, 네트워킹, 디스플레이, 상호작용과 같은 공통된 기능 그룹으로 분류된 관련 서비스들을 관리하는 소프트웨어로써, 공유 객체를 처리하기 위해 사전에 설정된 운용 방식에 따라 서비스들의 행위를 설정/제어하는 역할을 담당한다. 이러한 자동적인 서비스 설정/제어는 사용자에게 복잡한 서비스 조작을 숨길 수 있다.

서비스 관리자의 데이터 모델에서는 주어진 공유 객체를 처리하기 위해 실행되고 있는 서비스들의 데이터를 수집하여, 통합된 형태의 데이터로 관리한다. 예를 들어, 사용자의 위치에 따라 디스플레이 장치들 사이에서 시각 콘텐츠를 이동시키는 follow-me display 작업을 수행하는 경우, 사용자의 위치를 파악하는 위치 추적 서비스로부터 사용자 위치 정보를 수집하고, 표현해야 하는 영상에 대한 식별자를 연계하여 구조화된 정보로 저장한다. 응용 모델에서는 주어진 공유 객체에 대

한 조작 행위를 정의하며, 각 조작행위에 따라 관련 서비스들에게 적절한 지시를 내린다. 이때 관련 서비스들의 선택과 이들의 (실제적인 인터페이스 연결에 의한) 조합을 기술하는 워크플로우를 포함한다. 표현 모델에서는 공유 객체를 처리하기 위한 관련 서비스들 간에 연결 관계를 (기능적 의존 그래프 형식으로) 표현한다. 환경 모델에서는 주어진 공유 객체를 처리하기 위한 가용 서비스들의 상태 정보를 기술하며, 상호작용 모델에서는 주어진 공유 객체에 대한 상위 수준의 기능 제어 및 관리를 위한 인터페이스를 기술한다.

#### 다. 서비스 계층

서비스는 사용자들이 이용하기 위한 기본적인 협업 요소 기능들을 제공한다. 이를 위해서, (음성/영상 전송 프로그램과 같은) 협업 응용 소프트웨어에 직접 접근할 수 있는 부가적인 소프트웨어인 서비스를 구현한다. 서비스는 분산 컴포넌트 기술을 기반으로 개발되므로, 기본적으로 네트워크상에서 다른 서비스들의 접근을 허용하고 서비스 기능을 이용할 수 있도록 인터페이스를 제공한다. 단, 실제적인 접근과 제어는 주어진 권한에 의해 제약을 받을 수 있다. 네트워크를 통한 서비스 기능의 접근방식은 서비스들이 상호작용을 통해 새로운 협업 응용 서비스로 유연하게 만들 수 있도록 한다.

서비스의 데이터 모델은 서비스의 상태 (실행/대기 상태) 정보와 서비스가 수행하면서 획득한 문맥 (예를 들어, 네트워크 모니터링 서비스의 경우 패킷 손실률과 같은 모니터링 결과)을 포함한다. 응용 모델에서는 서비스가 제공하는 기능을 실행하기 위한 행위를 기술하며, 표현 모델에서는 서비스 운용 상태에 대한 정보와 서비스 실행 시 획득되는 결과를 사용자에게 보여주기 위한 방법을 제시한다. 환경 모델에서는 서비스가 이용하는 기존의 협업 응용 소프트웨어가 가지는 능력 및 설정 정보를 제공하며, 상호작용 모델에서는 응용 모델에서 제공하는 기능을 외부에서 활용하기 위한 인터페이스를 기술한다.

## 2. 계층별 구성요소의 기술방법

### 가. 작업

작업(task)는 협업 환경에서 협업의 목적을 완수하기 위하여 (지역 또는 원격 협업 노드들 상에서) 서비스 관리자들 간에 협력관계를 작업 그래프(task graph)를 이용하여 표현한다. 이를 위해, 작업 그래프는 서비스

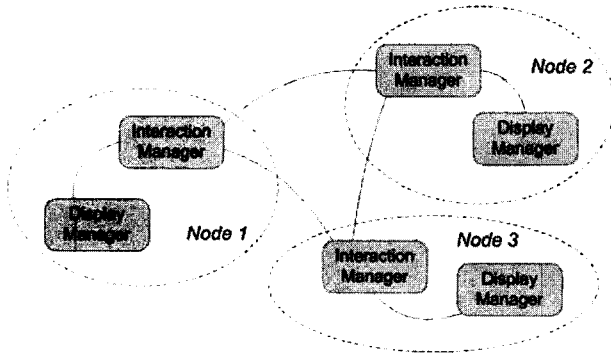


그림 2. 작업 그래프의 예  
 Fig. 2. An example of a task graph.

```

<task name="interaction-shared display task">
<components>
<manager name="mgr1" type="interaction manager"
mode="pointer-location-mode" usercontext="TBD"/>
<manager name="mgr2" type="interaction manager"
mode="pointer-location-mode" usercontext="TBD"/>
<manager name="mgr3" type="display manager"
mode="tiled-display-mode" usercontext="TBD"/>
</components>
<composition>
<adjacency src="mgr1" dest="mgr2"/>
<adjacency src="mgr2" dest="mgr1"/>
<adjacency src="mgr2" dest="mgr3"/>
</composition>
</task>
  
```

그림 3. 작업 그래프를 표현하는 XML 문서  
 Fig. 3. An XML document representing the task graph.

관리자의 목록과 서비스 관리자의 운용 방식, 그들 간에 상호 연결 관계, 사용자 문맥을 기술한다. 서비스 관리자들 사이에 상호 연결 관계는 서비스 관리자들 간에 직접적인 상호작용이 가능한 지 여부를 표시한다. 사용자 문맥은 사용자가 협업 노드에서 가지는 권한, 역할, 서비스 기호 등을 포함하며, 이 문맥을 기반으로 서비스 관리자는 차등화된 협업 서비스 기능을 제공할 수 있다.

그림 2에서는 디스플레이와 함께 공유되는 상호작용 정보를 원격 참여자들에게 전달하는 상호작용 공유 디스플레이 작업을 실현하기 위해 서비스 관리자들 간의 연결 예제를 보여준다. 이 예제에서는 상호작용 장치를 사용하는 사용자들에게 지역과 원격 노드들의 디스플레이 이들과 상호작용할 수 있도록 상호작용 관리자와 다른 협업 노드에 있는 디스플레이 관리자들을 서로 연결한다. 서비스 관리자들을 연결하기 위해서, 그림 3과 같이 인접 리스트 방식에 의한 작업 그래프를 XML 문서로 작성한다.

### 나. 서비스 관리자

주어진 작업의 수행에 참여하는 과정에서 서비스 관리자들은 다른 서비스 관리자들과 협력한다. 서비스 관리자들은 중요한 협업 문맥을 공유하고, 다른 서비스 관리자들과 협상을 통해 협력의 범위, 방법 등 세부적인 내용들을 조율한다. 예를 들면, 우리는 포인팅 장치를 이용해 네트워크 디스플레이 상에 표현되는 영상을 이동하는 작업을 가정해 보자<sup>[9]</sup>. 만약 상호작용 관리자가 영상에 대한 정보를 요구한다면, 디스플레이 관리자는 그 정보를 제공한다. 만약 상호작용 관리자가 영상을 이동하는 요청을 하면, 디스플레이 관리자는 이 요청에 대한 수락 여부를 결정한다 다음, 이를 처리한다.

이 때, 각 협업 노드는 서로 다른 자원 상황과 작업 수행에 따른 서비스 가용성에 제약을 가지므로, 서비스 관리자들 간에 서비스 이용권한을 선점하기 위한 경쟁이 유발되어, 안정적인 협업 응용 서비스를 제공하는 데 문제가 발생할 수 있다. 이를 해결하기 위해, 서비스 관리자는 운용 방식과 사용자 문맥에 기초하여 서비스 이용에 대한 우선순위를 결정한다. 구체적으로 서비스의 인스턴스 개수, 서비스가 이용 가능한 QoS 파라미터의 수치적 범위를 운용 방식과 사용자 문맥에 따라 차등화 함으로써 서비스 이용 시 발생하는 문제점을 조정할 수 있다. 서비스의 QoS 파라미터의 예를 들면, 미디어 공급자 서비스는 영상 프레임율의 상한선을 가질 수 있고, 미디어 수요자 서비스는 영상의 복호화를 위해 이용가능한 장치의 개수를 정할 수 있으며, 포인팅 서비스는 포인팅 정확도에 상한선을 정할 수 있다. 만약 서비스 관리자가 판단하기에 작업에서 요구하는 기능을 협업 노드가 충분히 제공할 수 없다면, 서비스의 QoS 파라미터를 하향 조정하여 서비스 기능을 제공한다.

주어진 작업을 처리하기 위해서, 서비스 관리자는 워크플로우를 기반으로 서비스 합성을 수행한다. 워크플로우는 상호 의존적인 관계를 가지는 서비스들 간에 작업 흐름을 어떻게 조정할 것인지 정형화된 방법에 따라 기술한다. 특정한 운용 방식에 의해서 선택된 워크플로우에 따라 서비스간 인터페이스 연결하여 실행하도록 중재하고, 정상적으로 요구된 기능이 수행되는지 감시한다. 예를 들면, 격자형 네트워크 디스플레이 상에 특정 공유 객체를 표현하고자 한다면, 그림 4에 나타난 디스플레이 관리자는 협업 노드에 존재하는 다양한 해상도와 크기를 가진 복수의 디스플레이 서비스를 조합하여, 네트워크 디스플레이 상에서 적절한 위치와 크기를 가진 콘텐츠를 표현한다.

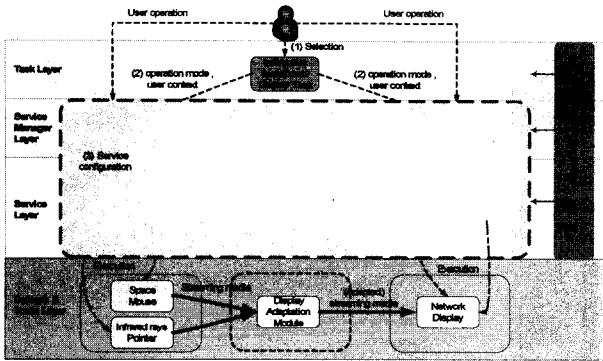


그림 4. 서비스 관리자와 서비스간의 상호작용  
Fig. 4. Interaction between service managers and services.

협업 노드의 변화에 대하여, 워크플로우에 대응되는 서비스들의 선택과 파라미터 값의 결정은 자율적인 방식에 따라 설정되어야 한다. 이는 현재 시간에 존재하는 이용 가능한 서비스들 중에서 가장 적절한 서비스를 찾은 후 현재 상황에 맞는 파라미터 값을 입력해야 함을 의미한다. 시간에 따른 문맥의 변동성에 대응하기 위해, 워크플로우는 서비스 발견, 정합, 합성, 모니터링을 지원하는 SMeet 미디어이터의 API를 통해서 기술되어야 한다. 이 방법에서, 만약 워크플로우가 서비스의 적절한 조건을 기술한다면, SMeet 미디어이터는 주어진 조건을 만족하는 서비스를 발견하여 워크플로우에 대응시켜준다.

다. 서비스

서비스는 서로 다른 능력과 전용 인터페이스를 가진 (오픈 소스) 협업 응용 소프트웨어들의 다양한 집합을 쉽게 수용해야 한다. 이러한 요구사항을 해결하기 위해,

```

<service>
<name>AudioService</name>
<role>prosumer</role>
<type>RAT-audio</type>
<maxInstanceNumber>1</maxInstanceNumber>
<cfg>
<tool>C:\AudioService\rat.exe</tool>
<workdir>C:\AudioService</workdir>
<argument>$(address)/$(port)</argument>
<transport>udp</transport>
<address>224.1.1.223</address>
<port>4321</port>
</cfg>
</service>
    
```

그림 5. 오디오 서비스 기술서  
Fig. 5. Audio service description.

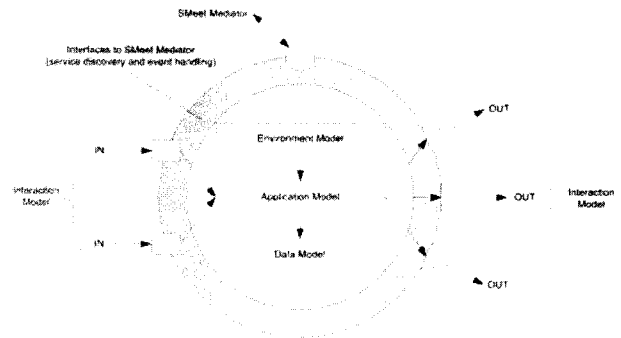


그림 6. 서비스 모델  
Fig. 6. Service model.

서비스는 협업 응용 소프트웨어들의 속성과 사전에 정해진 방식에 따라 이들을 실행하기 위한 인터페이스들을 이해하는 것이 필요하다. 서비스는 그림 5에서 기술되는 정보 요소들(즉, 이름, 역할, 유형, 서비스 인스턴스의 최대 수, 설정)을 포함하는 서비스 명세에 의해서 특성화된다. 역할은 이 서비스가 콘텐츠를 공급하는 지, 수신하여 활용하는 지, 아니면 두 가지 모두를 처리하는 지 설명하는 요소이다. 유형은 콘텐츠의 형식을 지정하는 요소이다. 서비스 인스턴스의 최대 수는 최대한 허용 가능한 서비스 인스턴스의 수이다. 설정은 추가적인 정보들, 예를 들면 전송 프로토콜, 커맨트라인 인터페이스, QoS 인자 등 협업 응용 소프트웨어를 조작하는데 필요한 요소들을 포함한다.

III. 에이전트 기반의 소프트웨어 구조 설계 및 서비스 합성 알고리즘

이 절에서는 다자간 협업 환경을 위한 서비스 합성 모델을 실체화하는 SMeet 관리 툴킷에 대해 소개하고 구성요소별로 설계 및 구현 상황을 설명한 뒤, 에이전트 기반의 서비스 합성 알고리즘을 제안한다.

1. 구성 요소의 설계

그림 7에서 제안하는 소프트웨어 구조는 고유한 협업 요소 기능을 가진 각각의 서비스들을 이용하여, 노드의 능력과 사용 방식에 따라 서비스들 선택하여 협업 노드를 유연하게 구축하도록 한다.

SMeet 미디어이터는 서비스들을 발견/설정/합성하는 등의 협업 노드 내 서비스들을 관리하는 역할을 맡는다. 협업 노드들과 상호 협력하기 위해서, SMeet 미디어이터는 그들과 협상(주어진 작업을 위한 협력 전략의 조율) 또는 정보를 교환(협업 맥락의 공유)하기 위해 협

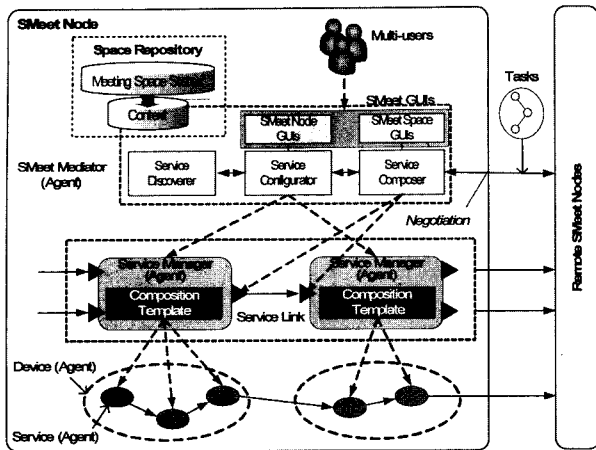


그림 7. 에이전트 기반의 SMeet 관리 툴킷 구조  
 Fig. 7. Agent-based SMeet management toolkit architecture

업 노드를 대표한다. 협업 맥락 (서비스 능력, 서비스 상태, 사용자 기호와 같은 상황 정보 등)은 서비스들로부터 수집되어, 협업 상황정보 저장소 (space repository)에 저장된다. SMeet 미디어이터는 제공하는 협업 환경에 대한 모든 서비스들의 구성 정보를 가지고 있으므로, 사용자 또는 서비스 관리자들이 가용 서비스 상태를 확인하고, 서비스들에게 특정 기능을 요구하고자 한다면, SMeet 미디어이터의 도움을 받아서 진행할 수 있다. SMeet 미디어이터를 구성하는 주요 기능 모듈은 다음과 같다.

Service discoverer는 노드에서 이용가능한 서비스를 발견하고, 해당 서비스에 대한 서비스 능력과 서비스 상태를 지속적으로 보고받아서 space repository에 저장한다.

Service matchmaker는 서비스 관리자가 원하는 조건 (서비스 능력 또는 상태)을 충족시키는 서비스를 찾기 위하여, 원하는 조건과 (완전히 또는 부분적으로) 일치하는 서비스들을 검색한다.

Service composer는 주어진 작업을 수행하기 위해서, 작업 그래프에 대응되는 서비스 관리자들이 있는 지 다른 협업 노드들의 SMeet mediator에게 질의하여, 그 결과에 따라 서비스 관리자들 간에 연결 관계를 구성하고, 실제적인 인터페이스 바인딩을 통해 서비스들을 합성한다.

SMeet node GUI는 노드를 구성하는 디바이스와 각 디바이스에서 이용 가능한 서비스들의 목록을 보여주고, 이들의 운용(설정, 시작, 종료)과 관련된 기본기능을 제공한다.

SMeet space GUI는 원하는 작업을 선택하기 위한 인터페이스를 제공하며, 실제 공동 작업이 실행되었을 때 유저가 공유 객체에 대한 세부적인 조작을 할 수 있도록 도와주는 기능을 제공한다.

디바이스 에이전트 (device agent)는 협업 환경에서 동작하는 PC, 이동단말, 내장형 장치에 탑재되어 장치

표 2. 서비스 에이전트가 제공하는 API  
 Table 2. API provided by a service agent.

메소드	파라미터	행위
init		Device 에이전트에 의해서 서비스가 추가되었을 때, 호출되는 메소드임.
start		SMeet 미디어이터 또는 다른 서비스, 서비스 관리자에 의해서 서비스가 시작되어 협업 응용 소프트웨어를 실행할 때, 호출되는 메소드임.
stop		SMeet 미디어이터 또는 다른 서비스, 서비스 관리자에 의해서 서비스가 중단되어 협업 응용 소프트웨어를 종료할 때, 호출되는 메소드임.
remove		Device 에이전트에 의해서 서비스를 제거할 때, 호출되는 메소드임.
serviceLinkAdded	Agent ID	다른 서비스 또는 서비스 관리자와 연결되었을 때 호출되는 메소드임. 파라미터는 연결되는 에이전트의 ID임.
serviceLinkRemoved	Agent ID	다른 서비스 또는 서비스 관리자와 연결이 해제되었을 때 호출되는 메소드임. 파라미터는 연결 해제되는 에이전트의 ID임.
getServiceOntologies		서비스 또는 서비스 관리자의 데이터 모델을 표현하는 온톨로지를 설정하는 메소드임.
startProcess		협업 응용 소프트웨어를 실행을 시작함.
stopProcess		협업 응용 소프트웨어를 실행을 중단함.
startServiceAgent	Agent ID	지정된 에이전트 ID를 가지는 서비스를 시작함.
stopServiceAgent	Agent ID	지정된 에이전트 ID를 가지는 서비스를 중단함.
search	Role / Type	지정된 역할과 유형을 가지는 모든 서비스들을 검색함.

를 SMeet 미디어이터에게 등록시키고, SMeet 미디어이터의 명령에 따라 서비스 또는 서비스 관리자를 추가, 삭제하는 역할을 담당한다. 또한 장치의 시스템 성능 (CPU 점유율, 메인 메모리 사용량 등) 및 서비스 상태 정보를 파악하여 SMeet 미디어이터에게 보고한다.

서비스 에이전트 (service agent)는 협업 응용 소프트웨어를 협업 노드에서 사용할 수 있도록 관리 가능한 대상으로 만들어주는 소프트웨어이다. 서비스 에이전트는 협업 응용 소프트웨어 기술서를 기반으로 서비스 능력을 파악하고, 실행 시 전달받은 서비스 문맥을 수집하여 SMeet 미디어이터 및 서비스 관리자에게 보고한다. 또한 SMeet 미디어이터 또는 서비스 관리자로 부터 수신한 서비스 조작 명령에 따라 협업 응용 소프트웨어 기능을 제어한다. 서비스는 데이터 모델을 기술하기 위하여 온톨로지를 활용하여, 응용 모델을 실현하기 위해서 에이전트 통신 프로토콜 (질의, 요청, 계약 네트워크 등)을 기반으로 서비스 인터페이스를 구현한다. 서비스 관리자 또는 서비스는 서비스 에이전트를 상속받아서 구현된다. 따라서 서비스 에이전트는 여러 가지 기능들을 표 2로 정리된 API로 구현하여, 서비스 발견/설정/합성에 관한 SMeet 미디어이터의 도움을 받을 수 있도록 지원한다.

2. 서비스 합성 알고리즘

제안하는 서비스 합성 알고리즘에서는 멀티 에이전트 시스템에서 작업 공유 및 결과 공유를 위해 널리 쓰이는 상호작용 프로토콜의 하나인 계약 네트워크 (Contract Net)<sup>[10]</sup>를 이용하여 유형 (type) 기반의 서비스 관리자간 합성 알고리즘을 그림 8과 같이 제시한다. 유형 기반의 서비스 합성 알고리즘은 합성 패턴에서 서

로 연결될 개체의 유형이 무엇인지를 정의하면, 이 유형에 일치하는 개체들을 찾아서 합성 시키는 방법을 의미한다.

(1) 작업 공지: 작업 그래프 상에 정점 (vertex)으로 표시된 서비스 관리자들 중에서 자기 노드에 포함되어 있는 것을 파악한다. 그런 다음, 그들과 연결되어야 하는 서비스 관리자들을 찾기 위해 작업에 이미 참여하고 있는 SMeet 미디어이터들에게 자신과 연결될 서비스 관리자를 찾기 위한 Call for Proposal 메시지를 전달한다.

(2) 제안: 작업에 이미 참여하고 있는 기존 노드들의 미디어이터 중에서 요청받은 서비스 관리자를 가지고 있는 경우, 자기 노드가 연결 대상이 될 수 있음을 알리기 위해 Proposal 메시지를 되돌려준다. 만약 작업에 새로이 참여하는 미디어이터 측에서 특정한 조건에 도달하지 못할 경우, 제안된 내용이 유효하지 못할 수 있다는 단서를 붙일 수 있으나, 본 논문에서는 이에 대한 내용은 다루지 않는다.

(3) 계약: 작업에 새로이 참여하는 미디어이터는 제안된 메시지들 중에서 연결 가능한 대상을 선택한 후, 서비스 합성을 위한 계약을 맺을 것을 accept 메시지를 통해 알린다. 이때 자신이 가지고 있는 서비스 관리자의 에이전트 ID를 덧붙여 보냄으로써, 상대방도 서비스 합성 내용을 업데이트할 수 있도록 한다.

(4) 결과보고: 계약을 통보받은 미디어이터는 자신이 가지고 있는 서비스 관리자의 에이전트 ID를 되돌려 주고, 현재 합성된 서비스들의 연결 관계에서 새로이 참여한 서비스 관리자를 추가하여 업데이트한 뒤, 업데이트 사실을 기존의 서비스 관리자들에게 알린다.

이와 같은 절차를 거치게 되면, 각 노드의 입장에서 서비스 연결 정보를 유지할 수 있게 되고, 결과적으로 주어진 작업에 참여하는 전체 노드의 미디어이터들과 서비스 매니저들 간에 서비스 합성을 위한 에이전트간 네트워크가 구성된다.

이후 서비스 매니저의 serviceLinkAdded 메소드가 호출되면서, 연결되는 원격 노드의 서비스 관리자에 대해서 알 수 있게 된다. 서비스 관리자는 원격 노드의 서비스 관리자와 협상하여, 주어진 워크플로우에 따라 실행되는 서비스들을 원격 노드의 서비스들과 연결시킨다. 연결된 서비스간 온톨로지에 의해 기술된 서비스 콘텍스트를 서로 교환하면서, 공유 객체의 처리를 위한 상호작용을 시작한다.

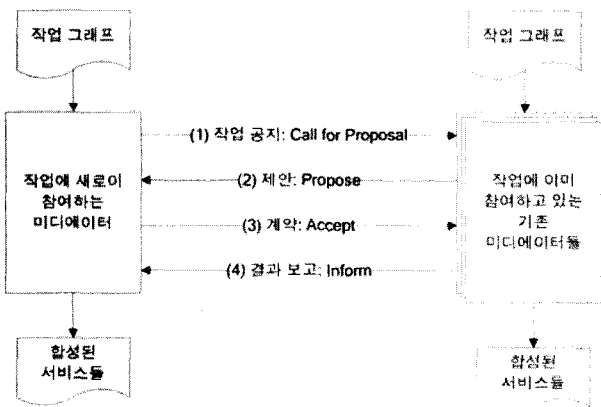


그림 8. 서비스 합성을 위한 계약 네트워크  
Fig. 8. Contract Net for service composition.



### IV. 구현 및 결과

제안하는 설계의 효용성을 검증하기 위하여, 우리는 JADE (Java Agent DEvelopment Framework) 미들웨어를 기반으로 SMeet를 위한 다중 에이전트 기반의 관리 툴킷을 개발 중이다. 현재 본 툴킷은 13,000 라인의 자바 코드로 작성되었으며, SMeet 테스트베드 하에서 설치되어 실험적으로 활용되고 있다.

참여자들은 SMeet 미디어이터를 통해서 디바이스 에이전트에 의해 조작되는 다양한 서비스 에이전트에 접근할 수 있다. 그림 9에서 보여지는 바와 같이, SMeet 미디어이터가 디바이스 에이전트를 발견하면, SMeet Node GUI 상에서 디바이스의 이름과 제공 가능한 서비스 목록을 보여준다. 따라서 SMeet Node GUI를 통해 참여자들은 연결된 장치들을 볼 수 있고 지원 가능한 서비스들을 확인할 수 있다. SMeet Node GUI를 이용하여, 사용자는 디바이스 에이전트 상에서 동작하는 서비스 에이전트들을 추가/삭제할 수 있다. 어떤 서비스 에이전트가 추가되었을 때, 서비스는 SMeet 미디어이터에 등록되며, SMeet 미디어이터로부터 조작 명령을 받아 처리한다.

SMeet 노드의 모든 에이전트들은 HTTP 기반의 분산 에이전트 통신 프로토콜을 사용하여 통신한다. 에이전트들 간에 메시지 전송이 FIPA 호환 에이전트 통신

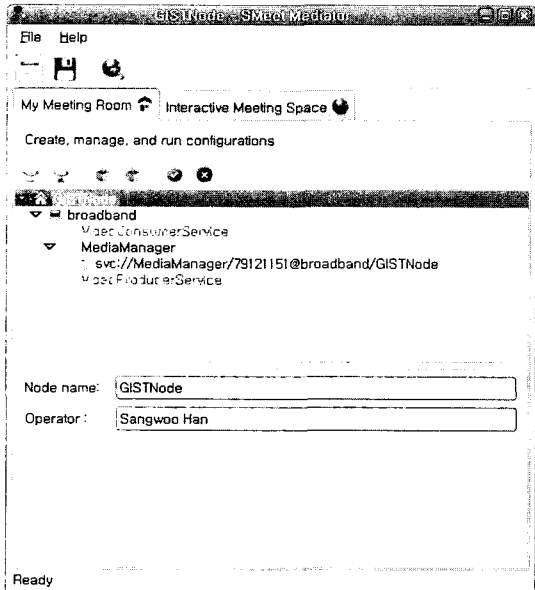


그림 9. 이용가능한 장치 및 서비스를 보여주는 SMeet Node GUI  
 Fig. 9. SMeet Node GUI representing available devices and services.

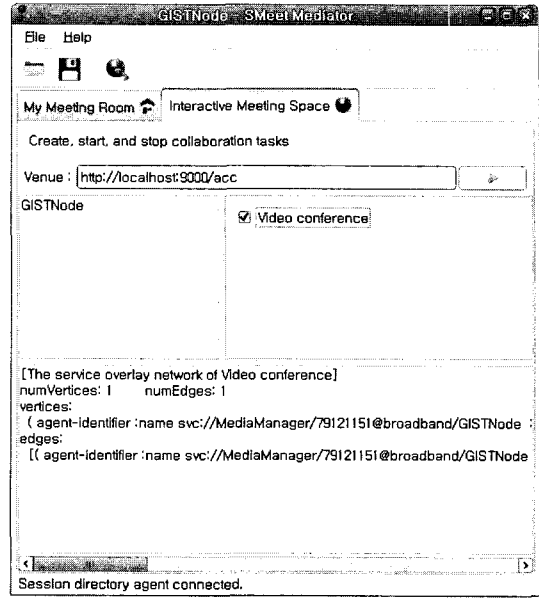


그림 10. 작업 선택을 위한 SMeet Space GUI  
 Fig. 10. SMeet Space GUI for task selection.

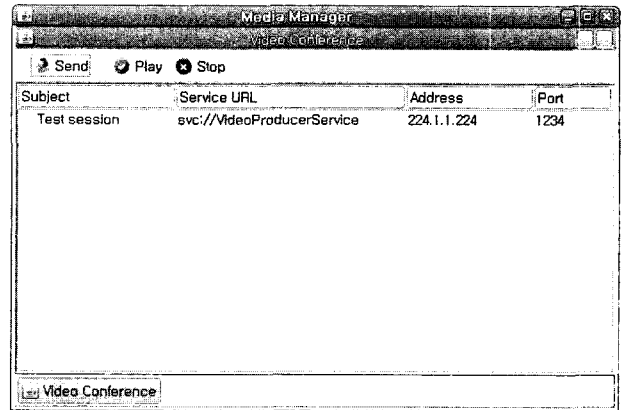


그림 11. 미디어 서비스 매니저의 GUI  
 Fig. 11. GUI of media service manager.

프로토콜을 기반으로 하고 있기 때문에, 온톨로지 기반의 협업 문맥과 함께 통신하는 것이 가능하다. 노드에서 서비스 조작을 위해 사용되는 노드 온톨로지와 노드 간 서비스 합성을 위해 사용되는 공간 온톨로지로 구성된 초기 수준의 온톨로지를 개발하였으며, 향후 더욱 견고한 협업 온톨로지를 설계할 것이다.

협업 노드들 간에 협업을 위해서, 참여자들은 협업 세션 디렉토리 에이전트에 접속한다. 그림 10에서는 접속된 협업 세션에서 선택 가능한 작업 목록이 나타나는데, 현재는 video conference 작업을 지원한다. 사용자가 이 작업을 선택하면, 이 작업 그래프에 해당되는 서비스 관리자가 자동으로 선택되어 서비스 관리자들에 연결이 이루어진다. 각 서비스 관리자는 그것이 속한 노드 안에서 (워크플로우에서 지정한 바에 따라) 합

성될 서비스를 찾고 이들을 사전에 정의된 서비스 파라미터에 따라서 설정한 후, 실행한다.

시연을 위해서 우리는 몇 가지 서비스들(미디어 공급자 서비스와 미디어 수요자 서비스)과 서비스 관리자(미디어 공급자 서비스와 미디어 수요자 서비스들을 조합하는 것에 의해서 음성/영상 회의 작업을 실행하기 위한 미디어 매니저)를 개발하였다. Video conference 작업은 미디어 서비스 매니저에 의해서 수행된다. 미디어 매니저는 미디어 공급자 서비스와 미디어 수요자 서비스들을 서로 연결시킴으로써, 참여자들 간에 영상회의를 할 수 있도록 한다. 그림 11에서는 구현된 미디어 서비스 매니저의 실행화면을 보여주며, Send 버튼을 누르면 영상을 송신할 수 있는 능력을 가진 (즉, 역할이 공급자이고, 유형이 미디어인) 서비스들을 검색하여 사용자가 선택하도록 한다. 선택된 미디어 공급자 서비스의 주요 내용은 참여중인 모든 노드의 미디어 관리자 GUI 상에 나열되며, 사용자가 이를 선택하고 Play 버튼을 누르면 선택된 미디어 공급자 서비스로 부터 영상 스트림을 수신할 수 있는 능력을 가진 미디어 수요자 서비스들을 검색하여 사용자에게 보여준다. 사용자는 그 중 하나를 선택하여 영상을 수신할 수 있다.

## V. 결 론

본 논문에서 우리는 다자간 협업 환경에서 에이전트 기반의 서비스 합성 모델을 제안하고, 이를 실현하기 위한 SMeet 관리 툴킷의 설계 및 구현 내용을 소개하였다. 제안하는 방법은 선택된 서비스들과 함께 협업의 목적에 맞도록 유연하게 협업 공간을 구성하는 것이 가능하다. 또한 복합적인 협업 작업을 제공하기 위한 서비스의 선택, 조합, 실행 방법에 대해 설계하였다. 제안하는 설계의 가능성을 확인하기 위하여, 다중 에이전트 기반의 협업 환경 관리 툴킷을 개발하였으며, 본 툴킷을 이용하여 미디어 관리자와 영상 회의와 관련된 서비스들을 구현하였다. 향후 연구로써 다중 에이전트 기반의 서비스 갈등 문제를 해결하기 위한 방법에 대해 해결할 예정이며, 1) 동시다발적으로 일어나는 다중 작업들 간에 서비스 선점 경쟁으로 인한 갈등 해소, 2) 이질적인 능력을 가진 협업 노드들 간에 협상 절차에 대한 방법론을 설계하고자 한다. 궁극적으로는 사용자의 노력을 최소화하면서 분산된 형태의 협업 환경을 쉽고 편하게 이용할 수 있도록 협업 환경의 관리 기능을 추상화하고자 한다.

## 참 고 문 헌

- [1] Han, S., Kim, N, Choi, K. and Kim, J.W. Design of multi-party meeting system for interactive collaboration. In Proc. of the *Second IEEE International Conference on Communication System and Software and Middleware*, Bangalore, India, 2007.
- [2] Román, M., Hess, C. K., Cerqueira, R., Ranganathan, A., Campbell, R. H. and Nahrstedt, K. Gaia: a middleware infrastructure to enable active spaces. *IEEE Pervasive Computing*, 1, 4 (2002), 74-83.
- [3] Ranganathan, A., Chetan, S., Al-Muhtadi, J., Campbell, R. H., and Mickunas, M. D. Olympus: a high-level programming model for pervasive computing environments. In Proc. of the *Third IEEE International Conference on Pervasive Computing and Communications*, Kauai Island, HI, 2005.
- [4] Chen, "Intelligent Agents Meet the Semantic Web in Smart Spaces," *IEEE Internet Computing*, pp.69-79, 2004.
- [5] Ooi, W. T., Pletcher, P., and Rowe, L. A. INDIVA: Middleware for managing a distributed media environment. *BMRC Technical Report*, 2002.
- [6] Masuoka, R., Parsia, B. and Labrou, Y. Task computing - the semantic web meets pervasive computing. In Proc. of *Second International Semantic Web Conference*, Sanibel Island, FL, 2003.
- [7] Bellifemine, F., Poggi, A., Rimassa, G. Developing multi-agent systems with JADE. In Proc. of *International Workshop on Agent Theories, Architectures, and Languages*, Boston, MA, 2000.
- [8] Tandler, P. The BEACH application model and software framework for synchronous collaboration in ubiquitous computing environments, *The Journal of Systems and Software*, 69 (2004), 267-296.
- [9] Ko, S., Kim, N., and Kim, J.W. Design of interaction manager supporting collaborative display and multimodal interaction for advanced collaborative environment. In Proc. of *SPIE*, 6777, 67770S (2007), Boston, MA.
- [10] FIPA Contract Net Interaction Protocol Specification.  
<http://www.fipa.org/specs/fipa00029/SC00029H.html>

저 자 소 개



한 상 우(학생회원)  
 2003년 중앙대학교  
 컴퓨터공학과 학사  
 2005년 광주과학기술원 정보통신  
 공학과 석사  
 2005년 3월~현재 광주과학기술원  
 정보기전공학부 박사과정

<주관심분야 : Network-based Collaboration Systems, Application QoS Negotiation and Provision, Service-Oriented Computing, Multi-Agent Systems.>



김 종 원(종신회원)  
 1987년 서울대학교 제어계측  
 공학과 학사  
 1989년 서울대학교 제어계측  
 공학과 석사  
 1994년 서울대학교 제어계측  
 공학과 박사

1994년 3월~1999년 7월 공주대학교 전자공학과  
 조교수  
 1997년 8월~2001년 7월 University of Southern  
 California 연구 조교수  
 1999년12월~2000년 7월 Technology Consultant  
 for VProtect Systems Inc.  
 2000년 7월~2001년 6월 Technology Consultant  
 for Southern California Division of  
 InterVideo Inc.  
 2001년 9월~2008년 3월 광주과학기술원  
 정보기전공학부 부교수  
 2008년 4월~현재 광주과학기술원 정보기전  
 공학부 교수

<주관심분야 : Networked Media Systems and  
 Protocols focusing "Reliable and Flexible  
 Delivery for Integrated Media over  
 Wired/Wireless Networks", (네트워크미디어:  
<http://netmedia.gist.ac.kr>)>