

## 동적 가중치 변화를 통한 추천 기반의 역할 할당 기법

# A Recommendation based Role-Assignment Method by Adapting Dynamic Weight Changing

이건수\*, 노승민\*\*, 김민구\*

Keon-Soo Lee\*, Seung-Min Rho\*\* and Min-Koo Kim\*

### 요 약

협업과정에 있어서, 협업의 대상으로 공간 안에 분포되어 있는 컴퓨팅 개체들을 사용하는 경우, 협업에 필요한 업무를 올바르게 수행할 수 있는 컴퓨팅 개체를 선택하여 협업에 임하는 것은 성공적인 협업 결과를 보장하기 위한 기본 조건이 된다. 협업 과정에서 수행해야 하는 작업을 그 협업에서의 역할이라고 할 때, 이 역할을 잘 수행할 수 있는 컴퓨팅 개체를 선택하기 위해서는 역할 수행에 필요한 기능의 소유, 협업 작업에 대한 충성도, 협업에 참여하고 있는 다른 개체들과의 연결성 등을 고려해서, 성공적인 협업 완료를 위해 현 상황에서 가장 적합한 개체를 선정할 수 있어야 한다. 이에 본 연구에서는 협업에 참여시킬 컴퓨팅 개체를 선택하기 위해, 개체의 속성 분석을 통해 적합도를 결정하고, 속성에 따른 가중치를 역할 할당의 상황에 따라 동적으로 변화시킴으로써, 최적의 협업 대상을 선택할 수 있는 역할 할당 방법을 제안한다.

### Abstract

In the process of cooperation which can be the best proposals for resolving complex problems in computing domain, the way of team organizing is one of the most important aspects for succeeding the goal. Especially in ubiquitous computing environment, where the participants of a team are selected from the heterogeneous computing objects which are deployed by other providers for their own goals, finding the relevant teammate can be regarded as the most important factor for determining the success or failure of the given problem. In this paper, we propose a method of finding teammate and assigning a role, which is a sub task of cooperation, by comparing the attributes of the computing object and the requirement of the role such as capability of functions, loyalty for the given team, and harmony with other teammates. By considering the situationally changing weights of each attributes, this method can be suited for dynamic computing environment where the cooperation should be executed with dynamically in/out computing objects and satisfy the dynamically chaining constraints.

Key words : Cooperation, Dynamic Allocation, Ubiquitous Computing

### I. 서 론

협업을 통한 문제 해결 방법은 복잡한 문제를 보

---

\* 아주대학교 정보통신전문대학원 (Graduate School of Information and Communication, Ajou University)

\*\* 고려대학교 전기전자전파 공학부(School of Electrical Engineering, Korea University)

· 제1저자 (First Author) : 이진수

교신저자 : 노승민

· 투고일자 : 2010년 12월 20일

· 심사(수정)일자 : 2010년 12월 20일 (수정일자 : 2011년 2월 21일)

· 게재일자 : 2011년 2월 28일

다 저렴한 비용으로 해결하기 위한 방법으로 연구되어 왔다. 단일 시스템의 성능 향상을 통한 문제 해결보다, 상대적으로 낮은 성능의 시스템들의 조합을 통해 주어진 문제를 해결하는 것이 효율적인 문제 해결 방법으로 인정받기 위해서는, 협업으로 인해 발생하는 협업 관리 비용이, 단일 시스템에게 요구되는 고성능의 비용보다 저렴하다는 것이 보장되어야 하고, 이러한 보장은, 다양한 컴퓨팅 도메인에서 증명되었다. 분산 시스템[1], 그리드 컴퓨팅[2], 멀티 에이전트 시스템[3]들을 대표적인 협업 기반 문제 해결 방법으로 꼽을 수 있는데, 이러한 협업 시스템들은 처음 시스템을 개발하는 과정에서, 각각의 역할이 고정된 상황으로 개발이 되는 특성을 갖고 있다. 가령, 특정한 목적을 위한 시스템을 분산 시스템으로 구현한다고 하면, 필요 기능에 대한 분석과 설계과정을 통해 각각의 기능을 응집도와 결합도를 고려해 독립적인 하위 모듈로 구성하고, 각각의 모듈을 개발하고 배치하는 과정을 통해 시스템을 완성하게 된다. 즉, A라는 분산 시스템의 A-1이라는 하위 구성요소는 주어진 작업을 수행하기 위해 설계 단계에서 정의되고, 시스템 구축 후에도, 그 정의된 작업 이외의 것을 수행하지 못하고, 시스템의 일부분으로서의 존재 의미를 갖게 된다. 그러나, 유비쿼터스 컴퓨팅 환경으로의 변화됨에 따라, 효과적인 협업 기반의 문제 해결을 위해서는 이 같은 시스템에 종속된 모듈들의 분산 배치 이상의 것이 필요하다. 기존에 이미 배치되어 있는 컴퓨팅 개체들의 재활용이 그것이다. 가령, <A, B, C>의 모듈로 구성된 시스템이 설치된 공간에, <A, D, E>의 모듈로 구성된 시스템이 새로 설치가 된다면, <A> 모듈을 새로 추가할 필요 없이 기존의 모듈을 참조할 수 있어야 한다. 이를 통해, 보다 효율적인 협업 공간에 대한 관리가 가능하게 될 것이지만, 이를 위해서는 다른 시스템의 <A>라는 모듈이 내가 필요로 하는 <A>라는 모듈과 동일한 기능의 모듈인지를 판단할 수 있어야 한다. 또, 내가 원하는 순간에 그 모듈을 사용할 수 있어야 한다는 것을 확신할 수 있어야 하고, 그렇게 사용하는 모듈의 성능을 확신할 수 있어야 한다. 본 논문에서는 이상의 제약 사항을 고려하여, 동일 공간 안에 존재하는 컴퓨팅 개체들 중에서 자신의 협업에 필요한 개체를 선정하여 그 협

업의 역할을 할당하기 위한 방법을 제안한다.

## II. 배경 지식

일반적으로 협업은 다음의 4단계의 흐름을 통해 이루어진다. 우선, 협업을 위해, 각각의 구성원들에게 나눠줄 작업을 분할한다. 그 후, 분할된 작업에 적합한 구성원에게 그 작업이 할당되면, 각각의 구성원들은 자신의 작업을 수행한 뒤, 그 결과물을 반환하게 되고, 그 결과물들은 통합되어 원래 문제의 협업을 통한 해결책으로 제시된다. 협업 수행에 있어서 모든 구성원들이 자율적으로 자신의 역할을 찾아 수행하고, 그 수행 결과물이 스스로 통합되도록 하기 위해서는 개별 구성원들의 작업 처리 능력이 뛰어나거나, 원 문제가 단순해야 가능하고, 이는 복잡한 문제를 낮은 수준의 개체들의 협업을 통해서 해결하려는 처음 의도에 부합되지 않기 때문에, 협업을 관리하기 위한 협업 관리자를 가정한다.

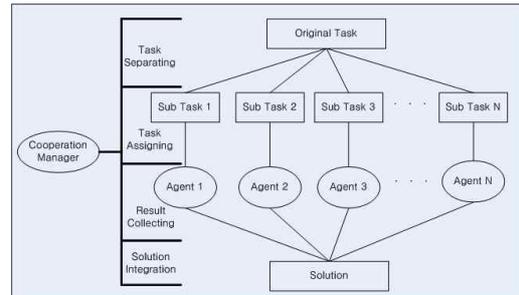


그림 1. 협업 흐름의 구조

Fig 1. General Process of Cooperation

협업 관리자가 적절한 협업 구성원을 선택하기 위해서는, 현재 주어진 공간 안에서 사용가능한 후보 개체들이 어떤 것들이 존재하는지를 알고 있어야 한다. 이에, 공간 안에 존재하는 개체들의 정보를 관리하고 있는 공간 관리자를 추가한다. 이를 통해, 하나의 공간 안에 설치되는 컴퓨팅 개체들은 설치되는 순간 공간 관리자에게 자신의 정보를 등록하게 되고, 협업 관리자가 새롭게 생성되면, 그 협업 관리자는 공간 관리자의 등록 정보를 참조하여, 원하는 협업 구성원을 선택할 수 있다. 협업 관리자가 자신의 협

업에 적합한 개체를 찾아 역할을 할당할 때, 협업의 참여관계에 대한 역학관계를 고려할 필요가 있다. 즉, 일상에서의 협업에서처럼, 팀장과 팀원 사이의 권력구조가 생기는 것과는 반대로, 기 분포되어 있는 다른 개체의 도움을 받아 문제를 해결하려는 협업 관리자에게 다른 개체들은 도움을 줄 수도, 주지 않을 수도 있고, 협업에 참여한 도중에도 언제든지 탈퇴하거나, 다른 작업을 먼저 수행할 수도 있다. 즉 기존 분산 시스템들처럼, 구성 모듈에 대한 독점적 사용 권한을 가정하고 있지 않기 때문에, 협업 구성원들의 선택 과정에서의 현재 상황에 따른 변화에 능동적으로 대처할 수 있어야 한다.

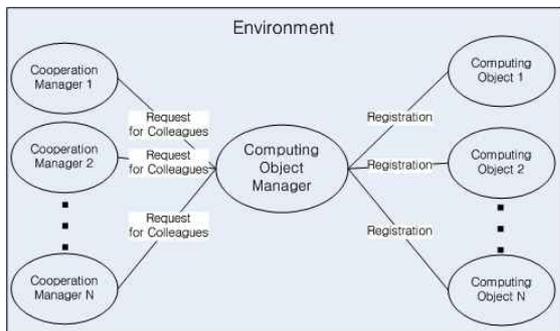


그림 2. 공간 관리자와 협업 관리자  
Fig 2. Relation between Cooperation Manager and Computing Object Manager

### III. 제안 방법

이러한 환경, 즉 공간 안에 상이한 기능의 컴퓨팅 개체들이 서로 연결되어 있고, 이들 기 분포되어 있는 장치들의 조합을 통해 새로운 문제를 해결하기 위한 시스템이 만들어지는 환경, 또는 유비쿼터스 혹은 퍼베이시브 컴퓨팅 환경으로 일컬어지는 미래 컴퓨팅 환경에서, 협업 관리자가 자신의 문제 해결을 도와줄 협업 구성원을 올바르게 선택하고, 역할을 맡기기 위해서는 새로운 역할 할당 방법이 필요하다. 이 방법은, 첫째, 기존에 존재하는 컴퓨팅 개체들의 기능이 협업의 역할을 수행하는데 있어서 요구되는 기능에 부합되는지를 확인할 수 있어야 한다. 둘째, 동적으로 변화하는 상황 변화를 감지하고, 그에 따른 기능의 변화를 고려해야 한다. 가령, A라는 기능을

갖고 있는 개체가 지금 다른 협업 과정에 포함되어 있다면, 혹은 그 기능의 외부 제약이 걸려있는 상황이라면, A의 작업을 위해 그 개체에게 역할을 할당할 수 없다. 셋째, 협업 관리자와 협업 구성원 사이의 신뢰도를 고려할 수 있어야 한다. 협업 구성원은 특정 협업에 참여하는 순간 그 협업에 종속되지 않는다. 따라서 협업 관리자는 가능한 신뢰할 수 있는 구성원을 선별해 낼 수 있어야 한다.

#### 3-1 역할과 기능 비교

역할을 수행할 수 있는 개체를 선택하기 위해서는, 역할의 요구 기능과 개체가 공간 관리자에게 등록하는 개체의 기능 정보를 비교하여 둘 사이의 포함 관계가 존재하는지를 찾을 수 있어야 한다. 일단 하나의 역할은 하나의 개체에게 할당되는 것을 가정하고 있기 때문에, 부분 포함 관계에 따른 동일 역할을 둘 이상의 개체가 나눠 담당하는 경우는 존재하지 않는다고 가정한다. 역할과 개체 사이의 포함관계를 찾아내기 위해서, 온톨로지를 사용한다. 온톨로지는 기능을 기술하기 위한 기능 온톨로지라, 개체의 상태 정보를 나타내기 위한 상태 온톨로지라 분류할 수 있다. 온톨로지를 통해 기술되는 협업과 개체의 정보는 다음의 구조를 갖는다. 역할 기술과 각 개체들의 기능 기술 정보를 비교하여 포함 관계를 갖는 개체들이 그 역할을 수행할 수 있는 후보 개체로 선정된다. 포함 관계는 구조적 포함 관계 (Structural Subsumption)를 기반으로 계산한다 [7].

표 1. 온톨로지 구조  
Table 1. Proposed Ontology

개념	정의
협업	{협업 이름, 역할 목록}
역할	{역할 이름, 기능 목록}
기능	{기능 이름, 선행 조건, 후행 조건}
개체	{개체 이름, 기능 목록, 상태 목록}
상태	{상태 이름, 정보 타입, 상태의 값}
선행 조건	{조건 이름, 상태 목록}
후행 조건	{조건 이름, 상태 목록}
정보 타입	{구간적도, 이항형, 명목형, 순서형, 비울척도}

3-2 상황 변화에 따른 동적 가중치 변화

역할과 기능의 포함 관계를 계산함에 있어서, 특정 기능이 현재 상황에서 사용 가능한 기능을 기준으로 포함 관계가 계산되는데, 이를 확인하기 위해 기능의 선행 상태와 후행 상태 정보를 참조하여, 현재 사용 가능한 기능인지를 판단한다. 이후, 명시적으로 주어지는 상태 정보를 제약 정보로 활용한다. 가령, A라는 역할을 수행하기 위해, A1, A2, A3이라는 기능을 필요로 하고, 이 기능을 갖고 있는 개체들 중에서, 특정 상태의 값이 "S1"의 값을 갖고 있는 개체로 제약 조건을 추가한다. 이상의 방법으로 협업 관리자는 공간 안의 개체들 중에서, 특정 역할을 할당 받을 수 있는 개체들의 후보 목록을 추출해 낼 수 있다. 이후, 후보들 중에서 순위를 생성하고, 순위에 따라 구성원으로 영입한다.

표 2. 개체의 순위 결정식  
Table 2. Candidated Objects' Ranking Method

$$\text{Rank} = \text{SR} * w_1 + \text{AL} * \sum w_n + \text{R} * w_2$$

순위 결정 공식은 <표 2>와 같다. SR(subsumption relation)은 기능 부합도를, AL(attribute list)은 각각의 속성 목록의 가중치를, R(reputation)은 개체의 평판 정보에 대한 가중치의 조합의 합으로 결정된다. SR의 경우, 각 역할에서 요구되는 기능에 근접할수록 높은 가중치를 갖는다. 이는 높은 컴퓨팅 능력을 갖고 있는 개체가 모든 역할 수행에 우선 책정되는 것을 방지하기 위함이다. 상태 정보에 대한 가중치는 명시적으로 주어지는 정보와 암묵적으로 주어지는 정보를 분리하기 위해 사용된다. 우선 명시적으로 주어지는 상태 비교는 절대 가중치를 갖는다. 즉, 아무리 기능상으로 올바르다고 해도, 명시적인 제약을 만족시키지 못하면, 그 역할을 할당 받을 수 없기 때문이다. 암시적인 상태 비교의 경우, 현재 등록된 개체들 사이의 속성을 비교하여, 상태 속성들 사이의 편차 값에 의해 그 가중치가 변화된다. 이는 기본적으로 작은 편차의 속성은 개체를 구분하기 위한 분별력이 떨어지기 때문이다. 이를 위해, 상태 속성의 편차를 기준으로 정규화 하여, 상태 속성의 유사도에 대

한 가중치를 생성한다. R로 표현되는 개체 평판은 그 개체의 신뢰도에 대한 값으로, 그 개체의 이전 협업 참여 기록의 바탕으로, 얼마나 협업 작업에 능동적으로 참여하는지를 판단하기 위함이다. 이를 위해, 매 협업이 완료되는 시점에서, 그 협업에 대한 평가를 실시하고, 협업의 성공과 실패에 따라 그 개체의 협업 평판도를 조절한다.

3-3 구성원의 신뢰도 검증

구성원의 신뢰도는 평판(Reputation)정보를 통해 계산된다. 평판 정보는 각 개체가 협업에 참여했던 경험에 의해서 계산된다. 평판도를 결정하기 위한 규칙은 <표 3>에 기술되어 있다. 표에서 보이는 것처럼, 평판도는 개체들 사이의 유사도와 협업의 결과를 제공하는 사용자들 사이의 유사도 정보를 기반으로 계산된다. 이를 위해서, 사용자들에 대한 군집화 작업을 수행하고, 개별 협업 기록을 관리한다. 즉, 유사한 사용자 그룹에 대한 협업이 수행될 때, 동일 그룹의 사용자를 대상으로 진행된 협업에 참여했던 개체, 동일 사용자에 대한 협업에 참여했던 개체는 다른 개체들보다 그 사용자를 위한 협업에서 보다 높은 우선 순위를 갖는다.

표 3. 평판 정보 결정 규칙  
Table 3. Decision Rules for Reputation Value

규칙
어떤 협업에라도 참여한 적이 있는 개체는 평판 정보는 증가한다.
동일한 협업에 참여한 적이 있는 개체의 평판 정보는 증가한다.
동일한 사용자를 대상으로 하는 협업에 참여했던 개체는 그 사용자에게 제공되는 협업에 대해서 높은 평판을 갖는다.
유사한 사용자를 대상으로 하는 협업에 참여했던 개체는 유사 그룹의 사용자를 대상으로 하는 협업 참여시 높은 평판을 갖는다.
이전에 참여했던 협업이 실패했던 경우, 그 개체의 평판은 감속한다.

IV. 실 험

이상의 방법을 적용하기 위해 “방문 간호 서비스”라는 가상 협업 서비스를 생성하였다. 원격지의 환자의 간호를 위해, 도우미, 의사, 간호사의 역할로 구성

된 서비스 팀을 구성하고, 특정 환자를 대상으로 등록된 여러 의료인들의 목록을 참조하여, 적합한 서비스 팀을 구축하는 과정에서 이상의 방법을 사용하였다.

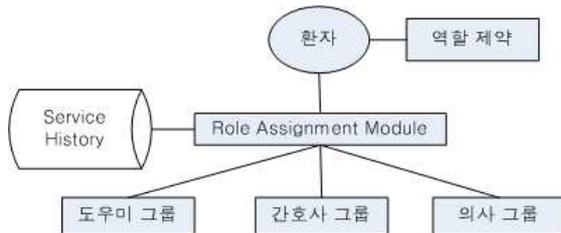


그림 3. <방문 간호 서비스> 구조도  
Fig 3 Architecture of 'Visit based Nursing Service'

각 역할 그룹에 소속된 개체들은 X-means 방법을 사용하여 군집화를 수행하였고, 서비스 히스토리는 R-DB를 사용해 관리하였다. 특정 환자를 위한 서비스 구성원들은 그 환자의 명시적인 제약 조건과 각각의 역할 그룹들 사이에서의 속성 가중치, 그리고 서비스 히스토리를 통한 평판 정보를 조합하여 그 환자에게 적합한, 각각의 도우미, 간호사, 그리고 의사를 추출해 서비스 팀을 구성해준다.

## V. 결 론

협업은 복잡한 문제를 보다 저렴한 비용으로 해결하기 위한 방법으로 원래의 문제를 보다 작은 문제들로 분할하고, 각각의 분할된 문제를 해결한 뒤, 그 결과를 통합하는 과정으로 이루어진다. 특히 유비쿼터스 컴퓨팅 환경처럼, 다양한 협업 기반 시스템들이 공존하는 환경에서, 새로운 시스템을 구현할 때, 그 시스템이 사용하려는 하위 작업을 해결할 수 있는 컴퓨팅 개체가 이미 다른 시스템에 의해 사용되고 있다면, 그 컴퓨팅 개체를 자신의 협업 구성원으로 포함시켜 문제를 해결함으로써, 전체 공간의 자원을 보다 효율적으로 사용할 수 있을 것이다. 이를 위해서는 환경 안에 존재하는 다양한 컴퓨팅 개체들이 어떠한 기능을 수행할 수 있고, 현재 어떤 작업을 수행하고 있으며, 지금 자신의 협업에 구성원으로 포함시킬 수 있을지를 판단하고, 협업에서의 역할을 할당하고, 그 결과를 받아들일 수 있어야 한다. 결국 협업이란, 협업

에서의 역할을 할당 받은 각각의 구성원들이 얼마나 자신의 일을 잘 수행하는지에 따라서, 그 결과가 달라지기 때문이다. 본 연구에서는 이처럼, 협업을 위한 역할을 할당하는 과정에서, 현재 상황에서 가장 그 역할에 적합한 컴퓨팅 개체를 선택하는 방법을 제안하고 있다. 이 방법은, 첫째, 온톨로지를 이용해 역할과 기능을 기술하는 어휘를 통일하였고, 둘째, 이들 기술 내용을 바탕으로 역할들과 개체들을 각각 군집화 하였다. 이후, 실제 역할을 할당 받아 이루어진 협업 데이터를 바탕으로 역할 그룹과 개체 그룹 사이의 연관 규칙을 생성하고, 개개의 개체들에 대해서 협업에 대한 경험치 요소의 값을 설정하였다. 마지막으로 이상의 정보들과 실제 역할 할당 과정에서 명시적으로 주어지는 협업 관리자의 속성 정보를 바탕으로, 주어진 역할에 적합한 컴퓨팅 개체들의 순위를 결정하여, 높은 순위의 개체들을 추천해준다. 이상의 방법은 동적으로 변화하는 상황의 흐름에 맞춰, 각 역할이 요구하는 속성의 중요도를 변화시켜줌으로써, 상황 변환에 따른 최적의 협업 구성원을 찾아 줄 수 있다.

## 참 고 문 헌

- [1] Andrews, Gregory R. (2000), Foundations of Multithreaded, Parallel, and Distributed Programming, Addison-Wesley, ISBN 0-201-35752-6
- [2] M. Baker, R. Buyya, and D. Laforenza, "Grids and Grid Technologies for Wide-Area Distributed Computign", *International Journal of Software:Practice and Experience*, Volumn 32, Issue 15, pp.1437-1466, Wiley Press, USA, Decempber 2002
- [3] Michael Wooldridge, "An Introduction to MultiAgent Systems", *John Wiley & Sons Ltd, 2002, paperback, 366 pages*, ISBN 0-471-49691-X.
- [4] N. Fornara and M. Colombetti. "Multi-Agent Systems: Semantics and Dynamics of Organizational Models, chapter Formal specification of artificial institutions using the event calculus" *IGI Global*, 2009.

[5] Razmerita, L., Angehrn, A., & Maedche, A. "Ontology-Based User Modeling for Knowledge Management Systems". In: Lecture Notes in Computer Science: 213-17. 2003

[6] Gruber, T.R "The Role of Common Ontology in Achieving Sharable, Reusable Knowledge Bases", *Principles of knowledge representation and reasoning: Proceedings of the Second International Conference, Cambridge, MA*, pp 601-602, 1991

[7] F. Baader, R.Kusters, and R. Molitor, "Structural Subsumption considered from an Automata Theoretic Point of View" *In Proceedings of the International Workshop on Description Logics, Trento, Italy*, 1998

이 건 수 (李建樹)



2002년 2월 : 아주대학교 미디어학 (공학사)

2004년 2월 : 아주대학교 정보통신공학(공학석사)

2004년 3월 ~ 현재 : 아주대학교 정보통신공학 박사과정

관심분야 : 지식표현 및 추론, 멀티 에이전트, 기계학습

노 승 민 (盧承民)



2001년 8월 : 아주대학교 정보 및 컴퓨터공학(공학사)

2003년 8월 : 아주대학교 정보통신공학 (공학석사)

2008년 8월 : 아주대학교 정보통신공학(공학박사)

2008년 10월 ~ 2009년 9월 : Carnegie Mellon University, Visiting Scholar

2009년 10월 ~ 현재 : 고려대학교 전기전자전파공학부  
관심분야 : 데이터베이스, 음악검색, 멀티미디어시스템, 기계학습, 지식관리

김 민 구 (金民九)



1977년 2월 : 서울대학교 계산통계학과 (이학사)

1979년 2월 : 한국과학기술원 전산학과 (공학석사)

1989년 2월 : 펜실베니아 주립대 전산학과(공학박사)

1997년 3월~ 1998년 2월 : 한국 정보과학회 인공지능 위원회 운영위원장

1999년 3월 ~ 2000년 2월 : Louisiana 대학, CACS연구과학자

1981년 3월 ~ 현재 : 아주대학교 컴퓨터공학과 (교수)

관심분야 : 지식표현, 추론, 멀티에이전트, 정보검색