

# 다중 사용자 환경에서의 에이전트를 이용한 소프트웨어 아키텍처 재구성

김상길<sup>o</sup> 안치돈 왕창종  
인하대학교 전자계산공학과

{skkim, cdahn}@selab.cse.inha.ac.kr, cjwangse@inha.ac.kr

## Software Architecture Restructuring using Agent on Multi-user Environment

Sang-Kil Kim<sup>o</sup> Chi-Don Ahn Chang-Jong Wang  
Dept. of Computer Science & Engineering, Inha Univer

### 요 약

다중 사용자 환경에서 새로운 소프트웨어 개발하기 위해서는 특정 사용자 관점이 아닌 다양한 관점에서의 의견이 최대한 반영된 소프트웨어 아키텍처를 설계할 필요가 있다. 그리고 설계 과정에서의 이러한 다양한 관점에서의 의견들을 조정하기 위한 협동 작업을 자동화해 줄 수 있는 메커니즘 또한 필요하다. 이 연구에서는 소프트웨어 아키텍처 설계 과정에서 사용자들의 의견을 최대한 반영한 아키텍처 설계를 위해 기존의 다중 계층 소프트웨어 아키텍처 관리 방법을 사용하고, 이러한 다중 계층 소프트웨어 아키텍처의 효율적인 재구성을 위한 자동화 방안으로 에이전트 기법을 도입한다.

## 1. 서론

분산 협동 처리(distributed co operative processing)에서 사용자가 원하는 작업을 자동적으로 해결해 줄 수 있는 소프트웨어가 요구됨에 따라 에이전트 기법이 등장하게 되었다[1]. 또한 에이전트간 통신(inter-agent communication) 개념은 에이전트 기반 분산 협동 처리에 필수적인 요소로 작용하고 있다. 그리고 CSCW(Computer-Supported Cooperative Work) 시스템은 정보 공유 구조에서 필수적인 기술로 자리잡고 있으며, 이를 위한 사용자간 공동 작업 기능과 정보 공유 기능이 목적이 되고 있다[2].

다중 사용자 환경에서 새로운 소프트웨어 개발을 위해 기존의 아키텍처에 대한 재사용성을 향상시킬 수 있는 방법으로 특정 사용자 관점이 아닌 다양한 관점에서의 의견이 최대한 반영된 소프트웨어 아키텍처를 설계할 필요가 있다. 그리고 설계 과정에서의 이러한 다양한 관점에서의 의견들을 조정하여 하나의 아키텍처로 재구성하는 작업을 자동화해 줄 수 있는 메커니즘 또한 필요하다.

이 연구에서는 소프트웨어 아키텍처 설계 과정[3, 4]에서 사용자들의 의견을 최대한 반영한 아키텍처 설계를 위해 기존의 다중 계층 소프트웨어 아키텍처 관리 방법[5]을 사용하고, 이러한 다중 계층 소프트웨어 아키텍처의 효율적인 재구성을 위

한 자동화 방안으로 에이전트 기법을 도입한다.

에이전트 기술을 도입하여 새로운 소프트웨어 개발을 위한 사용자들의 다양한 설계 관점을 최대한 반영할 수 있는 방법론에 필요한 기능들을 정의하고, 정의된 기능을 기반으로 하여 전체적인 개발 절차에서 에이전트의 역할들에 대해 기술한다.

## 2. 관련 연구 고찰

### 2.1 에이전트 기법

에이전트를 간단히 표현하면 사용자가 원하는 작업을 자동적으로 해결해 주는 소프트웨어라고 말할 수 있다. 에이전트는 주로 인지적인 측면(cognitive aspect)에서 지식 표현과 추론, 자연어 처리 등에 대한 연구 개발이 이루어졌고, 이를 바탕으로 전문가 시스템이나 지식 베이스 시스템 등의 어플리케이션 시스템이 개발되었다. 하지만 나날이 다양해지고 복잡해지는 사용자의 요구를 해결하기 위해서는 단독 에이전트의 능력으로는 한계가 있었고 그 해결 방법으로 지역적으로 분산되어 있는 다른 에이전트의 도움을 받아 처리하는 분산 협동 처리의 개념이 등장하였다. 여러 에이전트의 분산 협동 처리를 위해서는 에이전트간 통신이 필수적이며, 그 목적은 정보나 작업 처리의 공유와 교환에 있다. 즉 모든 작업을 완벽하게 처리할 수 있는 에이전트의 개발은 불가능하므로 자신이 가지고

있지 않은 정보에 접근하거나 자신이 해결하지 못하는 작업을 처리하기 위해 다른 에이전트에게 처리를 요구하는 단계가 필요하게 되는데, 여기에 에이전트간 통신을 이용하게 되는 것이다.

에이전트간 통신에서 가장 큰 고려사항은 각각의 에이전트가 가지고 있는 이질성(heterogeneity)에 대한 것이다[6]. 에이전트는 서로 다른 개발자에 의해, 그리고 서로 다른 시기에, 서로 다른 플랫폼을 바탕으로, 서로 다른 목적으로 개발되었기 때문에 이들간의 통신을 위해서는 상호 이해 가능한 언어와 프로토콜을 필요로 한다.

사용자의 의도대로 능동적으로 행동하는 에이전트 기술은 이제 적용되지 않는 분야가 없을 정도로 확산되고 있다. 인터넷 정보 검색, 온라인 쇼핑, 메시징, 네트워크 관리 등 에이전트를 필요로 하는 분야는 다양하며, 이론과 구조 등을 기반으로 하는 실제 사용 가능한 어플리케이션 소프트웨어 개발을 위해 많은 노력이 이루어지고 있다.

2.2 CSCW 어플리케이션

컴퓨터 시스템의 성능 향상과 정보 통신망의 확대로 네트워크 서비스를 이용하는 컴퓨터 사용자들이 많아지고 있다. 자신의 작업 공간에 있는 정보만을 사용하여 의사 결정을 하던 과거와는 달리 최근에는 개방된 환경 하에서 지역적으로 분산되어 있는 사람들이 서로의 정보를 공유하여 의사 결정을 하는 형태로 작업 환경이 바뀌고 있다. 또한 단순한 정보의 공유와 이용 뿐만 아니라 초고속 통신망을 통해 여러 사용자들이 공동 작업을 수행할 수 있는 환경을 쉽게 제공할 수 있게 되었다[7]. 이러한 통신망에 연결된 컴퓨터를 통해 그룹 활동을 지원하고자 하는 요구로 CSC W(Computer Supported Cooperative Work), 또는 그룹웨어(Groupware)라 불리는 새로운 어플리케이션 분야가 생겨났다.

CSCW 어플리케이션으로는 화상 회의 시스템, 상호 협력적 원격 교육, 원격 진료, 그룹 게임 분야 등으로 실생활의 거의 모든 분야로 확대되고 있다. 그러나 이러한 CSCW 응용을 개발하기 위해서는 정보 처리 기술, 고속 및 그룹 통신 기술, 분산 처리 기술 등의 다양한 기술적 사항들이 해결되어야 한다. 또한 CSCW 응용이 갖고 있는 공통적인 요구 사항들이 많이 존재하므로, 이러한 요구 사항들을 응용을 개발할 때마다 중복하여 개발한다는 것은 많은 개발 비용을 소요하게 된다. 따라서 CSCW 응용이 갖는 공통적인 요구 사항들과 어플리케이션을 보다 경제적으로 개발할 수 있는 체계적인 개발 환경이 필요하다. 이러한 개발 환경은 여러 가지 이질적인 환경에 응용할 수 있는 개방형 구조와 다양한 분야의 공동 작업을 구현하기 위한 동적 정책 모델을 가질 필요가 있다.

3. 분산 환경에서 에이전트를 이용한 아키텍처 재구성

3.1 에이전트를 이용한 공동작업

다중 사용자 환경에서 새로운 소프트웨어 개발하기 위해서는 우선 사용자들의 의견이 최대한 반영된 개발에 적합한 소프트

웨어 아키텍처를 설계하여 이후의 아키텍처 재사용을 위한 효율적인 메커니즘을 제공할 필요가 있다. 아키텍처 설계에 참여하는 사용자들의 전문성을 고려할 때 특정 사용자의 의견을 참조하는 것은 전체적인 의견이 반영된 소프트웨어 아키텍처 설계를 위해서 최적의 방법을 제공할 수 없다. 이 연구에서는 설계 과정에서 각각의 사용자들의 의견을 최대한 반영한 아키텍처 설계를 위해 기존의 다중 계층 소프트웨어 아키텍처 관리 방법을 사용하고, 이러한 다중 계층 소프트웨어 아키텍처의 효율적인 재구성을 위해 에이전트 기법을 도입한다.

다중 계층 소프트웨어 아키텍처 재구성 방법은 설계 단계에서 질의 생성을 위한 재사용 관점으로 적용될 수 있으며, 개발에 적합한 아키텍처 재구성을 위한 아키텍처와 컴포넌트 검색 과정 수행 이후 관리 목적으로 또한 적용될 수 있다. 다중 사용자 환경에서 새로운 소프트웨어 개발에 적합한 아키텍처를 재구성하기 위해서는 설계 단계에서 사용자들의 요구 사항을 분석하여 적절하게 조정하는 과정이 필요하다.

의견 조정을 위해 이 연구에서는 요구 사항들을 비교 분석하여 공통 부분을 추출하는 방법으로 사용자들의 의견이 최대한 반영될 수 있는 아키텍처 재구성 방법을 사용한다. 설계 과정에서 재구성된 다중 계층 소프트웨어 아키텍처는 이후의 재사용을 위해 아키텍처 저장소에 저장된다. 다음으로 적절한 사용자 요구사항 질의가 선택되면 이는 개발에 적절한 아키텍처 재구성을 위해 기존의 컴포넌트 검색 시스템에 입력된다. 재구성된 소프트웨어 아키텍처는 다중 계층 아키텍처 관리 방법에 의해 저장소에 저장되고, 또한 사용자들에게 뷰를 통해 검증 받게 된다. 그림 2는 다중 사용자 환경에서의 소프트웨어 아키텍처 재구성 절차를 순서 다이어그램으로 나타낸 것이다.

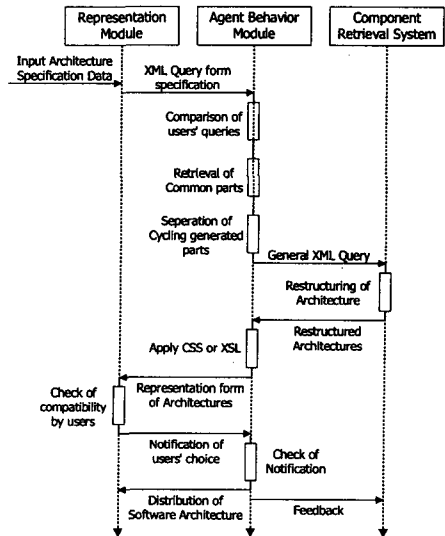


그림 1 소프트웨어 아키텍처 재구성 과정

이 연구에서는 각 사용자들의 다양한 의견이 포함된 질의 처리를 위한 도구 개념으로 에이전트 기술이 도입되었다. 에이

전트는 다중 계층 소프트웨어 아키텍처 구성을 위해 공통 부분을 추출하고, 의견 충돌이 발생한 부분에 대한 관리 방안에 대한 기능 정의가 필요하고, 재구성된 질의나 소프트웨어 아키텍처에 대한 검증 방법으로 재구성된 아키텍처를 사용자들에게 보여줄 수 있는 메커니즘에 대한 정의 또한 요구된다.

3.2 공동작업을 위한 에이전트 기능 정의

이 연구에서는 다중 사용자 환경에서 개발자들의 소프트웨어 아키텍처를 재구성하기 위해 에이전트 기법을 도입한다. 이를 위해 서로 다른 관점에서 설계된 개발자들의 요구사항을 최대한 반영할 수 있는 다중 계층 소프트웨어 아키텍처 재구성을 위한 에이전트의 기능에 대해 정의한다.

우선 에이전트는 각각의 사용자가 설계한 소프트웨어 아키텍처에 대해 비교 연산을 수행한 후 공통적인 부분을 추출하고, 기존 다중 계층 아키텍처 관리 방안에서 정의된 사이클 발생 부분에 대해 다른 계층으로 분리하는 기능에 대한 정의가 필요하다. 이 때 다른 계층으로 분리된 아키텍처나 컴포넌트, 또는 인터페이스에 대한 연결 정보는 상태 정보로서 유지되어야 한다. 소프트웨어 아키텍처 설계 과정에서 생성된 아키텍처가 컴포넌트 검색 시스템을 통해 재구성되면 이는 사용자들의 확인을 위해 웹 상으로 표현할 수 있는 방법이 제공되어야 한다. 따라서 에이전트는 CSS, 또는 XSL 과 같은 스타일시트를 적용하여 웹 상에서 사용자들에게 아키텍처 재구성 결과를 보여주는 기능에 대한 정의가 필요하다. 새로운 소프트웨어 개발을 위해 적합한 아키텍처가 재구성되었을 경우 사용자는 개발에 적용하고 에이전트는 재사용을 위해 버전 정보를 추가하여 저장소에 저장하는 기능을 수행하게 된다. 개발에 적합한 아키텍처가 구성되지 않았을 경우 에이전트는 컴포넌트 검색 단계로 피드백하여 적합한 소프트웨어 아키텍처 재구성을 위해 검색 과정을 다시 수행할 수 있어야 한다.

<표 1> 을 통해 다중 사용자 환경에서의 소프트웨어 아키텍처 재구성을 위한 에이전트의 기능성에 대해 정의하였다.

<표 1> 아키텍처 재구성을 위한 에이전트 기능 정의

공통부분 추출	컴포넌트와 커넥터 개수 비교(포함관계) 단순 컴포넌트명 비교(동일한 단순 컴포넌트) 반환 타입 비교 커넥터의 소스와 목표 컴포넌트 비교
사이클 발생 부분 분리	비교 연산 수행 후 사이클 발생 부분 검색 다른 평면으로 분리 공통 평면과의 연결 정보 유지
아키텍처 재구성	생성된 질의에 대해 검색 시스템을 사용하여 아키텍처 재구성 재구성 연결 정보와 함께 아키텍처 전달
웹으로 표현	CSS, 또는 XSL 을 적용 웹 상에서 사용자들에게 표현
저장 및 피드백	개발에 적합한 소프트웨어 아키텍처에 대해 버전 정보를 추가하여 저장소에 저장 부적합한 아키텍처에 대해서는 아키텍처 재구성 단계로 피드백하여 과정 재수행

4. 결론 및 향후 연구 과제

이 연구에서는 다중 사용자 환경에서의 새로운 소프트웨어 개

발을 위한 소프트웨어 아키텍처 설계 단계에서 사용자들의 다양한 관점을 최대한 반영할 수 있는 방법으로 기존의 다중 계층 소프트웨어 아키텍처 관리 방법을 사용한다. 그리고 이러한 사용자들의 서로 다른 관점에서 설계한 소프트웨어 아키텍처를 조정하기 위한 재구성 작업을 자동화하기 위한 메커니즘으로 에이전트를 사용하였다.

에이전트 기술을 기반으로 하는 새로운 소프트웨어 개발의 전체적인 개발 과정에서 사용자들의 다양한 의견을 조정하여 질의를 생성하는 단계에서부터 최종적으로 재구성된 소프트웨어 아키텍처들을 사용자가 보게 되는 과정까지 에이전트의 역할에 대해 살펴보았고, 이를 기반으로 이 연구에서 필요로 하는 에이전트의 기능에 대한 세부적인 사항들을 정의하였다.

이 연구에서는 다중 사용자 환경에서의 사용자들의 서로 다른 관점에서의 소프트웨어 아키텍처 설계에 대한 협동 작업과 관리 방법, 그리고 아키텍처 재구성 및 웹을 통해 사용자들이 확인할 수 있는 방법까지의 모든 기능을 에이전트가 담당함으로써 새로운 소프트웨어 개발에 필요한 모든 작업을 자동화할 수 있는 방법을 제공한다.

향후 연구 과제로는 다중 계층 소프트웨어 아키텍처를 효율적으로 관리할 수 있는 저장소 구조에 대한 명확한 정의가 필요하고, 버전별 검색 기능을 향상시킬 수 있는 버전 정보에 대한 추가 정의가 필요하다.

5. 참고문헌

[1] S. Franklin, A. Graesser, "Is it an agent, or just a program? : A taxonomy for autonomous agents, " *Proc. 3rd International Workshop on Agent Theories, Architectures, and Languages*, 1996

[2] 김남용, 이승근, 왕창중, " CORBA 기반 멀티미디어 응용을 위한 공동작업 서비스 설계," *한국정보처리학회 정보처리 논문지 제5권 1호, PP. 72, 1998*

[3] Y. S. Lee, K. S. Yoon, C. J. Wang, "Component Retrieval Based on Architecture for Reuse, " *Proc. International Conference on Software*, 2000

[4] C. D. Ahn, S. K. Kim, C. J. Wang, "XML based Component Specification Repository, " *Proc. International Conference on Software*, 2000

[5] 김상길, 안치돈, 왕창중, " 다중사용자 환경 컴포넌트 기반 소프트웨어 개발 시스템," *한국정보처리학회 춘계 학술발표논문집 제 7권 제 1호, 2000*

[6] D. Schwartz, "Cooperating heterogeneous systems : A blackboard based meta approach, " *Technical Report*, Center for Automation and Intelligent Systems Research, Case Western Reserve University, 1993

[7] J. J. Trevor, "Infrastructure Support for CSCW, " *The degree of Doctor of Philosophy, Department of Computing Lancaster University*, 1994