

## 화상회의 시스템의 유연성 개선을 위한 에이전트 내 QoS 자동 튜닝법

### An Automatic QoS Tuning Method for Improving Flexibility in Agent-based Flexible Videoconference System

이성독\*, 강상길\*\*

한국정보통신대학교 공학부\*, 수원대학교 정보공학부\*\*

Sung-Doke Lee\*, Sang-Gil Kang\*\*

School of Engineering, ICU\*, College of Info. Engineering, The University of Suwon\*\*

E-mail : sdlee@icu.ac.kr\*, sgkang@mail.suwon.ac.kr\*\*

#### ABSTRACT

에이전트 기반 플렉시블 화상회의 시스템은 데스크 탑 컴퓨터로 화상회의 시스템을 이용할 때 발생하는 시스템의 내·외적인 다양한 변동 및 이용자에게 가해지는 부담의 감소를 목적으로 설계된 화상회의 이용 지원 환경이다. 본 논문에서는 플렉시블 화상회의 시스템의 유연성을 개선하기 위한 한 방법으로서 QoS(Quality of Service) 파라미터 자동 튜닝법을 적용한 화상회의 매니저 에이전트의 새로운 아키텍처를 제안한다. 제안한 화상회의 매니저 에이전트를 기반으로 구축된 플렉시블 화상회의 시스템은 일반 이용자들이 화상회의를 이용할 때 발생하는 서비스 품질 요구 변경 및 시스템 부담 등의 문제를 유연하게 해결한다. 매니저 에이전트는 다른 에이전트들과 프로토콜을 이용하여 긴밀하게 협조를 하며, 문제 해결의 진행상황을 고려하면서 서비스 품질 파라미터를 자동으로 조정한다. 조정된 파라미터에 의해서 시스템은 내·외적 변동 대한 유연한 대응이 가능하게 되었고 또한 이용자에게 가해지는 부담도 감소한다. 결국, 제안 아키텍처를 적용한 시스템이 기존 시스템과 비교하여 유연성이 증가되었음을 실험을 통하여 증명한다.

**Key words:** 에이전트 기반 플렉시블 화상회의, 매니저 에이전트, 서비스 품질, 자동 튜닝법

## I. 서 론

다양한 컴퓨터 및 네트워크 환경에서 화상회의 시스템을 이용하는 경우[1], 이용자의 시스템 및 상대편 시스템의 자원상황 그리고 컴퓨터의 화상회의 처리 능력 등을 고려해야 하는데 전문지식이 없는 일반 이용자에게 있어서 이런 작업이 많은 부담이 되었다. 에이전트 기반 플렉시블 화상회의 시스템은 일반 데스크 탑 컴퓨터로 화상회의 시스템을 이용할 때 발생하는 다양한 이용자 부담을 감소하는 것을 목적으로 한 화상회의 이용 지원 환경이다. 기존의 플렉시블 화상회의 시스템은 내적 또는 외적 변동에 대처하기 위한 구성이 고정적이었고, 발생한 문제에 대처하는 수단이 한정적이었기 때문에 중요도나 긴급도를 고려한 유연한 대응이 부족하였다[2, 3].

본 논문에서는 에이전트가 QoS 문제 해결을 수행할 때에 발생한 문제의 해결 진행 상황을 고려하고, 파라미터를 자동 조정하는 기법을 적용한 효과적인 에이전트 아키텍처를 제안한다. 또한, 제안 아키텍처를 기반으로 플렉시블 화상회의 시스템을 구축하고, 실험을 통하여 QoS 감소 등의 문제 등을 유연

하게 해결하는 등, 기존의 시스템 보다 유연성이 개선되었음을 보인다.

## II. 본 론

### 2.1 플렉시블 화상회의 시스템

에이전트 기반 플렉시블 화상회의 시스템 (Agent-based Flexible Videoconference System: FVCS)의 중요한 특징은 화상회의 중에 발생하는 CPU 부하 및 네트워크 자원 상황의 변화와 같은 시스템 내·외적 변동 또는 이용자의 QoS 변경 요구에 따라서 시스템이 제공 할 수 있는 QoS를 자율적으로 조정하는 것이다. 본 논문에서는 이와 같이 회의 중에 발생한 문제해결을 위한 유연성 개선에 초점을 둔다.

그림 1은 FVCS의 기본적인 아키텍처를 나타내고 있다. 화상회의 중에 발생하는 문제들은 그림에서 보여주고 있는 에이전트 작업공간(Agent Workspace) 내 존재하는 각 에이전트들 간의 협조에 의해서 실현된다. 구체적으로는 화상회의 서비스

전체를 관리하는 에이전트인 화상회의 매니저 (Videoconference Manager: VCM) 에이전트인 VCM-A 및 VCM-B를 중심으로 에이전트 그룹 간 협조에 의해서 문제 해결을 시도한다.

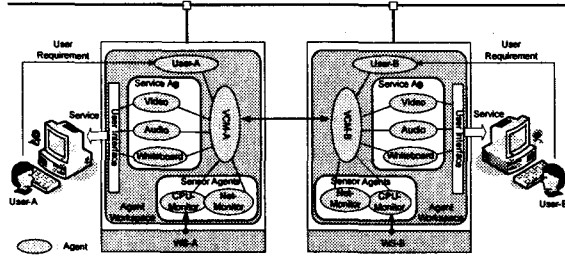


그림 1. 플렉시블 화상회의 시스템

즉, 매니저 에이전트는 사용자 에이전트(User Agent) 및 센서 에이전트 그룹(Sensor Agents: CPU-Monitor, Net-Monitor)으로부터 사용자 요구 정보 또는 자원상황 정보를 분석, 평가하고 그 변화에 대처하기 위한 동작 계획을 작성한다. 이 계획을 기본으로 화상회의 프로세스를 직접 제어하는 서비스 에이전트 그룹(Service Agents: Video, Audio, Whiteboard)에 제어동작 요구를 실행하는 것으로 사용자 요구를 가능한 한 충족시키면서 안정한 시스템 동작을 실현한다.

### 2.2 에이전트 지식처리의 개선 방침

본 절에서는 기존의 FVCS 문제점을 해결하기 위해 다음과 같은 방침에 의해서 기존 ADIPS[4] 지식처리의 아키텍처를 새롭게 구성한다.

- (1) 기존의 VCM 에이전트 구조를 개선한 새로운 아키텍처를 설계한다.
- (2) 문제해결 전략을 적절하게 바꾸기 위한 지식과 QoS 파라미터 자동 튜닝법을 도입한다.
- (3) 에이전트를 설계 할 때에 효율성, 재 이용성 그리고 기술성(記述性)을 고려하고, 이상의 지식이나 기구를 모듈화 한다.

### 2.3 화상회의 매니저 에이전트의 설계

2.2절에서 서술한 개선 방침을 기본으로 지식처리를 위한 화상회의 매니저(VCM) 에이전트의 아키텍처를 그림 2와 같이 구성한다.

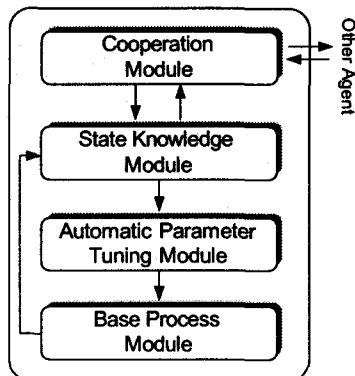


그림 2. VCM 에이전트 아키텍처

### 2.3.1 협조 모듈

협조 모듈(Cooperation Module: CM)은 발생된 문제의 성질이나 협조적 문제 해결 중의 목적 달성도, 문제해결 기한 등에 대하여 결정하여, 다른 에이전트와 메시지를 교환하고 협조하는 역할을 수행한다. VCM 에이전트 간 협조 프로토콜은 표 1에 나타난 프로토콜을 이용하여 다른 에이전트와 협조 처리를 수행한다. 작업 공간 내에서 VCM 에이전트와 서비스 에이전트 간에는 표 2의 프로토콜에 의해서 협조 처리를 수행한다.

표 1. VCM 에이전트 간 협조 프로토콜

| Performative                    | Summary  |
|---------------------------------|--|
| <i>Request-Make-Coop</i>        | S requests R to start cooperation                          |
| <i>Acceptance-Make-Coop</i>     | S accepts a request from R to start cooperation            |
| <i>Refusal-Make-Coop</i>        | S refuses a request from R to start cooperation            |
| <i>Request-Close-Coop</i>       | S requests R to terminate cooperation                      |
| <i>Acceptance-Close-Coop</i>    | S accepts a request from R to terminate cooperation        |
| <i>Refusal-Close-Coop</i>       | S refuses a request from R to terminate cooperation        |
| <i>Request-Change-Status</i>    | S requests R to change the parameters of QoS               |
| <i>Acceptance-Change-Status</i> | S accepts a request from R to change the parameters of QoS |
| <i>Refusal-Change-Status</i>    | S refuses a request from R to change the parameters of QoS |

표 2. 작업공간 내 에이전트 간 프로토콜

| Performative         | Summary                                       |
|----------------------|---|
| <i>RequestAction</i> | S requests R to do something                  |
| <i>Acceptance</i>    | S sends the <i>RequestAction</i> message to R |
| <i>Refusal</i>       | S refuses the <i>RequestAction</i> message    |
| <i>RequestConf</i>   | S requests the confirmation to R              |
| <i>AcceptConf</i>    | S accepts the confirmation                    |
| <i>RefusalConf</i>   | S refuses the confirmation                    |
| <i>Report</i>        | S sends some information to R                 |

### 2.3.2 상태 지식 모듈

상태 지식 모듈(State Knowledge Module: SKM)은 에이전트 간 협조에 의해서 실행된 사항들을 저장하거나 화상회의 중 발생한 상황에 따라서 에이전트를 구성하여 제공하는 역할을 한다. 그리고, 협조 모듈 및 저장소 내 지식을 이용하여 자동 조정 모듈에 QoS 정보를 제공하는 역할을 한다.

### 2.3.3 자동 파라미터 조정 모듈

자동 파라미터 조정 모듈(Automatic Parameter Tuning Module: APTM)은 CM으로부터 요구한 값들(CPU 부하, QoS 파라미터의 값, 제한시간 등)을 실제로 결정하는 역할을 수행한다. 여기에서 결정된 값들은 베이스 프로세스 모듈로 보낸다. 본 절에서는 FVCS의 유연성을 개선하기 위한 방법으로서 새롭게 제안 한 자동 QoS 파라미터 자동 튜닝법에 대하여 설명한다. QoS 파라미터는 화상회의 이용자의 파라미터 우선도에 따라서 차별하게 조정하여 현재의 자원상황 (CPU 부하, 네트워크 대역폭 등)을 원하는 상태에 맞도록 시도한다. 또한, 프레임 비율(frame rate), 화질(quality), 해상도(resolution)와 같은 QoS

파라미터를 조정하는데 있어서 그 조정 폭은 오실레이션 하는 상태를 피하기 위해서 점차로 줄인다. 시간  $t$ 에서 원하는 자원 상태를  $d(t)$ , 현재 상태를  $y(t)$ 로 한다면, 여기에서  $y(t)$ 는 식 (1)과 같이 QoS 파라미터 집합의 함수로 표현 될 수 있다. 왜냐하면 현재의 자원상황은 각 파라미터 값에 의존하기 때문이다. 만약 파라미터 값이 증가하면 증가 할수록 사용 가능한 자원상황이 감소되고, 반면에 파라미터 값이 감소하면 사용 가능한 자원 상황이 증가된다.

$$y(t) = f(A(t)) \quad (1)$$

여기에서,  $A(t) = [a_1(t), a_2(t), \dots, a_n(t), \dots, a_n(t)]$ 는 각 파라미터의 집합이고,  $a_i(t)$ 는 시간  $t$ 에서  $i$ 의 파라미터 값이다. 그리고 시간  $t$ 에서 원하는 자원 상황  $d$ 와 현재의 자원상황  $y(t)$  사이의 에러  $e(t)$ 는 다음과 같이 표현할 수 있다.

$$e(t) = |d - y(t)| \quad (2)$$

식 (2) 에서 보는 바와 같이  $y(t)$ 의 값은  $A(t)$ 의 값에 의해서, 그리고  $e(t)$ 의 값은  $y(t)$ 의 값에 의해서 결정된다. 한편, 각각의 파라미터 값들에 의해서  $e(t)$ 는 0 또는 이에 근사한 값의 방향으로 되도록 조정되어야 한다. 예를 들면 첫번째로 조정된 파라미터 값들의 집합을  $A(t+1) = [a_1(t+1), a_2(t+1), \dots, a_n(t+1), \dots, a_n(t+1)]$ 로 표현할 수 있다. 여기에서 괄호 내에 있는 1은 첫번째 조정을 의미한다. 그리고  $a_i(t+1)$ 는 다음의 식 (3)과 같이 표현할 수 있다.

$$a_i(t+1) = a_i(t) + \Delta a_i(t) \quad (3)$$

여기에서  $\Delta a_i(t)$ 는 이터레이션에 있어서 파라미터  $a_i(t)$ 의 조정 폭을 나타낸다. 본 논문에서  $\Delta a_i(t+1)$ 는 파라미터  $a_i(t)$ 의 비율의 식(4)와 같이 표현된다.

$$\Delta a_i(t+1) = \rho_i \cdot a_i(t) \quad (4)$$

여기에서  $\rho_i$ 는 조정자(scaling factor)이다. 조정자의 크기는 사용자의 파라미터 우선도에 의해서 결정된다. 만약 어떤 파라미터 조정자 값이 작으면 사용자는 화상회의에 사용되는 다른 파라미터 보다 더 높은 우선도를 제공한다. 조정자 값은 사용자의 파라미터  $a_i$ 의 우선도 또는 선호도에 의해서 다르게 된다. 예를 들면 만약 파라미터들의 우선 순위가 가장 낮은 것에서 가장 높은  $a_1(t), a_2(t), \dots, a_n(t), \dots, a_n(t)$  순서로 되었을 때 조정자 값은  $\rho_n, \rho_{n-1}, \dots, \rho_1(t), \dots, \rho_1(t)$ 와 같이 높은 값에서 낮은 값 순서로 된다. 파라미터의 우선에 따른 조정자 값은 임의로 결정할 수 있지만 이상적인 조정자 값은 반복된 실험경험에 의해서 결정된다. 한편, 파라미터 조정에 따른 현재의 자원상황 및 에러는 식 (5) 및 (6)과 같이 표현된다.

$$y(t+1) = f(A(t+1)) \quad (5)$$

$$e(t+1) = |d - y(t+1)| \quad (6)$$

여기에서 만약 에러  $e(t+1)$ 가 만족한 값에 도달 되었을 경우(0 또는 이와 근사한 값) 파라미터 조정 프로세스를 마친다. 그렇지 않으면 에러 값이 0에 근접할 때까지 이터레이션에 의해서 조정 프로세스를 계속 수행한다. 식 (4)에서 (6)까지 본

바와 같이 조정자 값은 프로세스 동작 중에 중요한 역할을 한다. 만약 조정자 값이 너무 크면 조정 속도는 빠르지만 파라미터의 최적 값으로 접근하지 않고 오실레이션한다. 또한 조정자 값이 너무 작으면 조정 속도가 느리고 최적 값에 도달하는 시간이 많이 소요된다. 이런 문제를 피하기 위하여 식 (7)에 나타난 바와 같이 반복 횟수가 증가 할수록 파라미터의 조정 폭을 감소시킨다.

$$\Delta a_i(t+k) = \Delta a_i(t+k-1)/2^{k-1} \quad (7)$$

여기에서  $k$ 는 반복 수이다. 본 기법에 의해서 결정된 CPU 부하, 네트워크 대역폭, 그리고 각 파라미터 값들은 계속 수정되어 다른 모듈로 보내지며, 화상회의의 시스템이 지속적 또는 유연하게 동작하도록 하는 역할을 한다.

### 2.3.4 베이스 프로세스 모듈

베이스 프로세스 모듈(Base Process Module: BPM)은 이용자 에이전트 및 센서 에이전트 그룹으로부터 이용자 요구 정보 또는 자원상황 정보를 분석, 평가한다. 그리고 APTM에서 결정된 최적의 파라미터 값으로 화상회의의 프로세스를 직접 제어하는 서비스 에이전트 그룹에 제어동작 요구를 실행하는 것으로 이용자 요구를 가능한 한 충족시키면서 안정한 시스템 동작을 실현한다.

## 2.4 실험 및 평가

### 2.4.1 실험환경

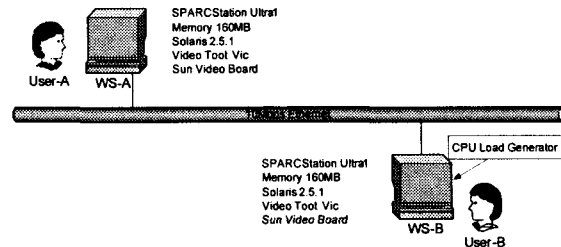


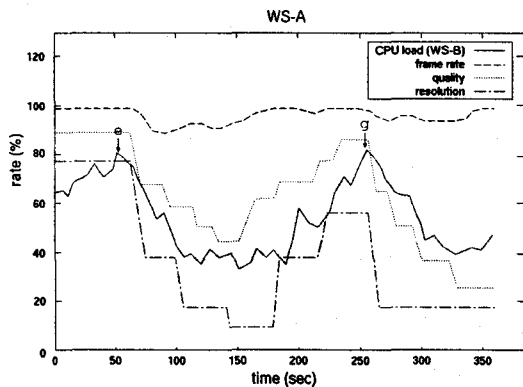
그림 3. 실험환경

본 논문에서 제안한 아키텍처의 유용성을 확인하기 위해서 그림 3의 환경에 적용하여 실험하였다. 구체적으로는 기존의 멀티에이전트 기반 프레임워크인 ADIPS[4]에서 문제가 되었던 저장소 및 지식 부분을 새롭게 설계한 VCM 에이전트를 플렉시블 화상회의의 시스템 VCM-A 및 VCM-B에 적용하였다. 서비스 에이전트가 조정하는 각 파라미터 조정을 위해서 기존에 개발된 화상회의의 용 vic[1]를 이용하였다. VCM 에이전트 및 센서 에이전트 그리고 서비스 에이전트는 Tc//Tk[5]를 이용하여 기술하였다. 각각의 하드웨어는 에이전트 작업공간 및 화상회의용 단말로서 SPARCStation Ultra1을 이용하였다.

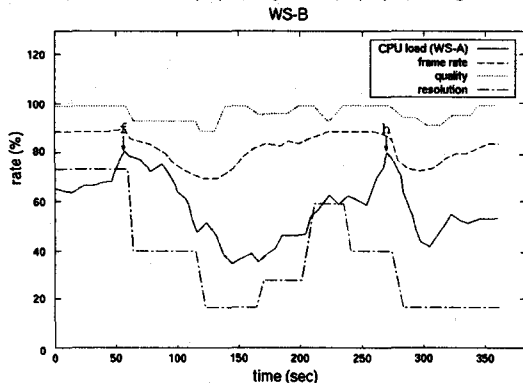
### 2.4.2 자원변동에 대한 시스템의 유연성 평가

화상회의 중에 발생하는 CPU자원이 동적으로 변화하는 상황에서 제안 아키텍처를 기반으로

구축한 FVCS의 유연성 동작을 관찰하였다. 먼저, 시스템 동작 중에 사용자 B(User-B)측 단말(WS-B)의 CPU에 대하여 부하 조정용 프로세스를 기동하여 부하를 주었을 경우 사용자 A(또는 사용자 B)에 제공되는 동화상에 대한 QoS 파라미터 동작 상황을 관찰하였다. 여기에서, WS-A측 동화상에 대한 QoS 우선도는 프레임 비율, 화질 그리고 해상도 순으로 하였다. 또한, WS-B측은 화질, 프레임 비율 그리고 해상도 순으로 하였다. 시스템 내 환경에 있어서 자원(예, CPU부하)의 부족 문제가 발생한 경우 그 문제를 해결하는 상황을 그래프로 표시하였다. 각 그래프에 있어서 x-축은 시간(초), y-축은 CPU 부하 및 QoS 파라미터 값에 대하여 각각의 값 (CPU load: 100%, frame rate: 30-fps, quality: 32-level, resolution: 5-level)을 100%로 했을 때의 비율을 나타낸다. CPU-Monitor는 부하가 80%를 넘었을 때에 감지하는 것으로 하였다. 목표로 설정한 CPU 부하는 40%-60% 사이로 하였다.



(a)WS-A에서의 QoS 파라미터 변동



(b)WS-B에서의 QoS파라미터 변동  
 그림 4 CPU 자원변동에 대한 대응

그림 4에서는 제안 QoS 파라미터 자동 튜닝법 및 아키텍처를 적용한 FVCS의 CPU 자원 변동에 대하여 대응 실험 결과를 나타내고 있다.

그림 (a)의 50초가 흐른 시점(e점)에서 CPU-Monitor-A에서 VCM-A에 메시지를 보내고, 이때 VCM-A는 에이전트 내부의 ATPM에서 각 파라미터들의 값을 동시에 조정한다. 이용자의 우선도에 따라서 다르게 조정된 파라미터 값에

의해서 CPU 부하를 감소시키고 있다. 그림 (b)의 경우는 55초가 흐른 시점(f점)에서 VCM-B의 ATPM에 의해서 조정된 파라미터 값들을 내려 CPU에 걸리는 부하를 감소시키고 있다. VCM-A 및 VCM-B 간의 협조에 의해서 부하를 감소하여 어느 정도 시간이 흐른 뒤, 재차 WS-B 측의 CPU에 부하를 걸었을 경우(g, h점)에도 에이전트 간 협조 및 각 VCM-A, VCM-B 내 ATPM의 동작에 의해서 CPU의 부하를 감소시키고 있다.

그림 4의 결과에서 보여지는 바와 같이 CPU 부하를 감소시키기 위한 CM의 에이전트 간 협조 및 ATPM에서 자동 QoS 조정 기법이 반복적으로 잘 동작하고, 결국 CPU 부하를 지속적으로 목표 설정 범위로 조정하고 있음을 알 수 있다.

### III. 결론

본 논문에서는 에이전트 기식기반 화상회의 시스템의 서비스 조정 능력 및 시스템의 유연성을 실현하기 위한 에이전트의 아키텍처를 제안하고, 발생한 문제의 성질이나 문제 해결의 진행상황을 고려하면서 전략에 따라서 QoS를 자동으로 조정하여 유연하게 문제 상황에 대처 가능하게 하는 자동 QoS 파라미터 튜닝기법을 에이전트에 적용하고 실험을 실행하였다. 그 결과 FVCS에 있어서 다양하고, 긴급을 요하는 조건의 경우에도 에이전트간 협조가 효과적으로 실행되었으며 기존의 FVCS보다 더욱 더 유연하게 QoS 제어가 가능 했음을 보여주었다.

이후 과제로서는 인터넷을 통한 다자간 화상회의 실험으로의 확장 및 보다 지능적인 QoS 제어를 위해서는 화상 회의 시스템 평가용의 평가 함수 등을 추가로 도입하여 유연성을 정량적으로 평가하는 것이 남아있다.

### IV. 참고문헌

- [1] S. MaCanne and V. Jacobson, "vic: a flexible framework for packet video", ACM Multimedia, Nov. 1995, Page(s): 511-522
- [2] T. Sukanuma, T. Kinoshita, K. Sugawara, and N. Shiratori, "Flexible Videoconference System based on ADIPS Framework", Proceeding of the 3<sup>rd</sup> International Conference (PAAM 98) 1998, Page(s): 83-100
- [3] S. D. Lee and D. S. Han, "Multiagent based Adaptive QoS Control Mechanism in Flexible Videoconference System", ICACT 2004, Vol. II, Feb. 2004, Page(s): 745-750
- [4] T. Kinoshita and K. Sugawara, "ADIPS Framework for Flexible Distributed Systems", Springer-Verlag Lecture Notes in AI, 1599, 1998, Page(s):18-32
- [5] Ousterhout, J. K., "Tcl and the Tk Toolkit", Addison-Wesley, 1994