퍼지 및 지능시스템학회 논문지 2005, Vol. 15, No. 5, pp. 605-614

화상회의 시스템의 유연성 개선을 위한 에이전트 지식 구성 및 평가

Construction and Evaluation of Agent Knowledge for Improving Flexibility in Videoconference System

이성독*, 강상길**

Sung-Doke Lee, Sang-Gil Kang

한국정보통신대학교*, 수원대학교**

School of Engineering, Information and Communications University*

Department of Computer Science, College of Information Engineering, The University of Suwon**

요 약

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 지원환경이다. 본 논문에서는 플렉시블 화상회의 시스템의 유연성을 개선하기 위한 한 방법으로서 서비스 품질 파라미터 조정 기법을 적용한 화상회의 매니저 에이전트 내 지식 부분을 새롭게 설계한 T-INTER(Tuning-INTER) 아키텍처를 제안한다. 제안한 아키텍처를 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 에이전트와 프로토콜을 이용하여 긴밀하게 협조를 하며, 문제 해결의 진행상황을 고려하면서 QoS 파라미터를 자동으로 조정한다. 조정된 파라미터에 의해서 시스템은 변동에 대한 유연한 대응이 가능하게 되었고, 또한 이용자에게 가해지는 부담도 감소한다. 결국, 제안 아키텍처를 적용한 시스템이 기존 시스템과 비교하여 유연성이 증가되었음을 실험을 통하여 증명한다.

Abstract

In this paper, we present the design and implementation of an agent knowledge and QoS tuning methodology to improve the flexibility of agent-based flexible videoconference system. In order to improve the flexibility during videoconferencing, we propose a new T-INTER(Tuning-INTER) architecture of knowledge part in videoconference manager (VCM) agent in which an automatic QoS parameter tuning method is imbedded. The flexible videoconference system structured based on the proposed architecture can cope with the changes in service quality required by users. The VCM agent cooperates with other agents by protocols and executes the automatic QoS parameter tuning task whenever needed. By the tuned parameters, the system is able to flexibly cope with the internal or external changes and the burden of users can be decreased. In the experimental section, it is shown that our proposed system outperforms the existing system.

Key Words : 플렉시블 화상회의 시스템, QoS 조정, 화상회의 매니저 에이전트, INTER 지식

1. 서 론

최근, 인터넷의 광범위한 확대와 더불어 이를 활용하여 미디어 전송 또는 화상회의 시스템을 이용하는 사람들의 폭발적인 증가에 따른 네트워크 대역폭 및 이와 관련된 문제들을 해결하기 위한 네트워크 기술이 점차로 개선되고 있다. 또한 인터넷을 이용하여 화상회의를 할 때 중요시 되고 있는 QoS(quality of service) 문제를 해결하기 위해서 시스템 및 이를 활용한 어플리케이션 레벨에서의 기술 개발도 필요하게 되었다. 이런 필요성에 의해서 기존에 다양한 컴퓨터 및 네트워크 환경에서 인터넷을 통하여 화상회의를 할 경우 이를 지원하기 위한 시스템 및 소프트웨어의 연구가 진행되어 왔다[1-3]. 그러나 이런 화상회의용 지원 시스템들은 자신 및 상대편 시스템의 자원상황 그리고 컴퓨터의 화상회의 처리 능력 등을 고려하지 않고 개발되었기 때문에 전문지식이 없는 일반 이용자에게 있어서 많은 부담이 되었다. 이를 개선하기 위해서 컴퓨터에 익숙하지 않은 일반 이용자가 원활한 협동작업을 하는 수단으로서 사용하는 에이전트를 활용한 플렉시블 화상회의 시스템의 연구가 진행되고 있다[4-7].

에이전트 기반 플렉시블 화상회의 시스템은 일반 데스크 탑 컴퓨터로 화상회의 시스템을 이용할 때에 발생하는 다양한 이용자 부담의 감소를 목적으로 구성된 화상회의 이용 지원 환경이다. 구체적으로는 기능의 유연성(flexibility)을 기존의 화상회의 시스템에 도입하여 이용자 요구나 네트워크

접수일자 : 2005년 8월 30일
완료일자 : 2005년 9월 28일

환경 등의 변화에 대응하고, 그 기능이나 성능이 자율적으로 변화하는 지원 환경이다. 여기에서 유연성이란 시스템의 내부 또는 외부에서 발생하는 변동에 대한 대응 능력 및 이용자가 사용하기 편리한 서비스 제공 능력을 말한다. 플렉시블 화상회의 시스템은 화상회의 중에 발생하는 이용자 요구 및 시스템 내·외부 네트워크 자원 상황 변동에 따라서 시스템을 구성하는 에이전트가 갖고 있는 전략적 지식을 기초로, 변동에 대처하기 위한 각종 QoS 파라미터의 조정을 수행한다. 그렇지만 기존의 플렉시블 화상회의 시스템은 이런 변동에 대처하기 위한 대응 방안이 고려되지 않았고, 발생한 문제에 대처하는 수단이 극히 한정적이었기 때문에 중요도나 긴급도를 고려한 보다 유연한 대응이 부족하였다.

이 문제를 해결하기 위하여 본 논문에서는 에이전트가 QoS 문제 해결을 수행 할 때에 발생한 문제의 해결 진행 상황을 고려하고, QoS 파라미터 값을 자동 조정하는 기법을 적용한 효과적인 에이전트 지식 부분의 아키텍처를 제안한다. 또한, 제안 아키텍처를 기반으로 플렉시블 화상회의 시스템을 구축하고, 실험을 통하여 QoS 감소 등의 문제를 유연하게 해결하는 등, 기존의 플렉시블 화상회의 시스템 보다 유연성이 개선되었음을 보인다.

본 논문의 구성은 다음과 같은 순서로 구성되어 있다. 제2장에서는 관련연구에 있어서 기존의 화상회의 시스템 및 에이전트 기반 플렉시블 화상회의 시스템에 관하여 기술한다. 제3장에서는 기존 플렉시블 화상회의 시스템을 이용한 실험에서 발생한 문제점의 개선 방침 및 QoS 파라미터 조정 기법을 적용한 매니저 에이전트 내 지식 부분 아키텍처에 관하여 기술하고, 에이전트 간 동작 예를 기술한다. 제4장에서는 제안 아키텍처를 에이전트 기반 플렉시블 화상회의 시스템에 적용하고, 실험 및 평가를 통하여 유효성에 관하여 기술한다. 제5장은 결론이다.

2. 관련연구

기존 화상회의 시스템(Videoconference System: VCS)의 어플리케이션 레벨에서 QoS제어에 관한 연구는 IVS[1] 및 프레임웍 기반 접근 방법[2] 등이 있다. IVS는 인터넷을 통한 화상회의를 목적으로 개발되었는데 네트워크 조건 변화에 대한 정보를 가지고 화상 코더에서 파라미터들의 제어로 데이터 전송률을 조정한다. 또한 매우 단순한 사용자 요구를 수용하고 있다. 한편 프레임웍 기반 접근 방법은 "네트워크 인식" 어플리케이션 구축을 위해서 두개의 기본적 시도를 제공하고 있다. 즉 관찰된 네트워크 서비스 속성으로 어떻게 동적 변화를 찾을 수 있으며, 네트워크 중심 속성에서 어떻게 어플리케이션 중심 속성으로 변화시킬 수 있는지에 관하여 서술하고 있다. 이런 시도들은 환경 변화에 대한 적응도의 부족 및 시스템 부하 균등 능력에 있어서 한계가 있었다. 화상회의 중에 발생하는 변화의 종류는 다양(CPU변동, 네트워크 상황 변화, 사용자 요구 변화 등등)하고, 그 대응 방법도 다양하게 요구된다. 즉, 문제가 발생했을 때 그 해결의 어려움, 필요성, 긴급도, 그리고 이에 요구되는 정확도 등을 고려한 유연한 대응이 필요하다. 따라서 이와 같은 문제점의 해결을 위한 한 방법으로서 에이전트 기반 플렉시블 화상회의 시스템이 설계되었다.

한편, 이전에 개발된 플렉시블 화상회의 시스템(Flexible Videoconference System: FVCS)[4-7]은 주로 에이전트 지식 기반 컴퓨팅 프레임웍(Agent-based Distributed Information Processing System: ADIPS, Distributed Agent System Based on Hybrid Architecture: DASH)[9-11]을 이용하여 구성되었다. FVCS에서는 작업의 위임이나 충돌 또는 지연 해결 등을 포함한 상위레벨 연산처리의 유연성이 요구되기 때문에 지능 정보 취급 능력과 처리간의 복잡한 정보 교환이 필요하다. 따라서 이런 이유에 적합한 방법으로서 기존의 VCS에 ADIPS 또는 DASH를 적용하고 있다. 이런 FVCS의 공통적인 특징은 에이전트 내 지식 부분이 룰 베이스 형식으로 구성되어 있다. 또한, 기존의 VCS에서 부족한 (F1) 화상회의를 시작 할 때 필요한 서비스 구성기능, (F2) 화상회의 중의 서비스 조정기능, 그리고 (F3) 화상회의 중의 서비스 재구성 기능을 도입하여 유연성을 실현 할 수 있는 장점을 가지고 있다.

그러나 이전에 개발되었던 플렉시블 화상회의 시스템의 실장 및 프로토타입 시스템의 동작 실험을 실행한 결과 화상회의 매니저 에이전트 내 지식 (구체적으로는 ADIPS 내 INTER 지식)을 기초로 수행하여지는 서비스 조정기능 (F2)에 관하여 다음과 같은 문제점이 있었다[9,10]. 참고로 여기에서 INTER 지식이란 ADIPS 내의 6개 지식(TASK, PROC, BID, RECOV, CHANGE, INTER) 중에서 에이전트 간 협조를 실현하기 위한 지식으로서 다른 에이전트에 요구를 전달하거나 또한 다른 에이전트로부터의 요구를 처리하는 지식이다. 보다 구체적인 내용은 문헌[9]에 설명되어 있다.

(P1) 발생한 문제의 특징에 대하여 유연한 대응 부족: 화상회의 중에 발생하는 변화의 종류는 다양(CPU변동, 네트워크 상황 변화, 사용자 요구 변화 등등)하고, 그 대응 방법도 다양하게 요구된다. 기존 화상회의 시스템에서는 설계자의 경험을 기반으로 설계된 미리 정해진 룰 지식에 의하여 선택되었기 때문에 문제가 발생했을 때 QoS 파라미터 조정에 있어서 선택의 폭이 한정되어 정확도 등을 고려한 유연한 대응이 필요하다.

(P2) 변화에 대하여 유연한 대응 알고리즘의 부족: 화상회의 중에 발생하는 상황 변화를 처리하고 있을 때에도 다른 변화가 속속 발생한다. 또한, 발생한 문제의 해결이 반드시 양호하게 처리된다고 한정 할 수 없기 때문에 변화에 대하여 자동으로 QoS 파라미터를 조정 할 수 있는 알고리즘이 필요하다.

위의 두 문제는 기존의 ADIPS를 이용한 FVCS 내 에이전트의 INTER 지식 구성에 있어서 변화에의 대응 전략이 부족하였던 것과, 변화를 판단하고 대처할 수 있는 판단 기준이 명확하지 않아서 유연한 문제해결 능력이 부족했다는 것에 기인한다. 따라서 위의 두 문제를 해결하기 위하여 에이전트 내 지식 구조를 새롭게 설계하고, 각 사이트의 자원 상황이나 이용자 요구를 고려하면서 상대측 간의 요구를 받으면서 고도한 협조를 실현하기 위한 기술이 필요하다.

3. 화상회의 시스템에서 매니저 에이전트 지식부분의 설계 및 적용

3.1 플렉시블 화상회의 시스템

에이전트 기반 플렉시블 화상회의 시스템(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 QoS

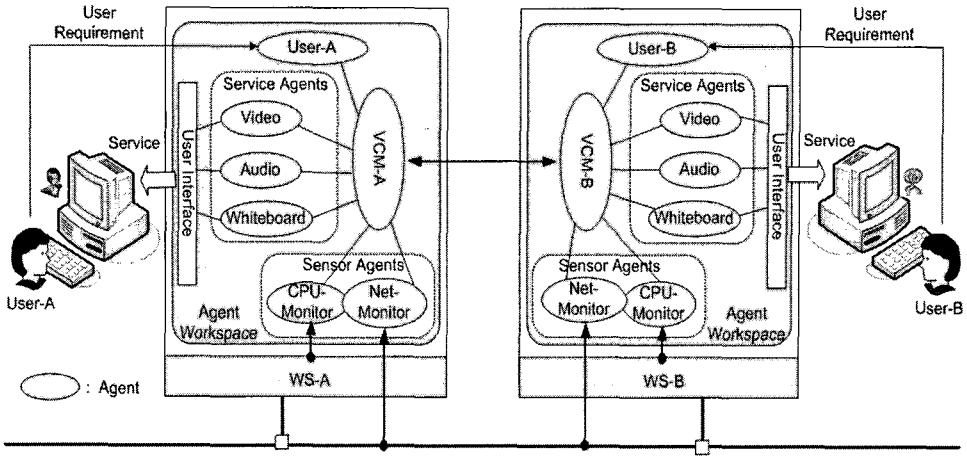


그림 1. 에이전트 기반 플렉시블 화상회의 시스템
Fig. 1. Agent-based flexible videoconference system

를 자율적으로 조정하는 것이다. 본 논문에서는 이와 같이 회의 중에 발생한 문제해결을 위한 유연성 개선에 초점을 둔다. 즉 CPU 이용률 또는 네트워크 이용 가능 대역폭의 변동 등과 같은 시스템 내부 또는 외부 변화에 대응하고, 안정하게 동작하는 성질을 말한다.

그림 1은 FVCS의 기본적인 아키텍처를 나타내고 있다. 화상회의 중에 발생하는 문제들은 그림에서 에이전트 작업 공간(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 간의 협조에 의해서 문제 해결을 시도한다. 즉, 매니저 에이전트는 이용자 에이전트(User-A, User-B) 및 센서 에이전트 그룹(Sensor Agents: CPU-Monitor, Net-Monitor)으로부터 이용자 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 이용자 요구를 가능한 한 충족시키면서 안정한 시스템 동작을 실현한다.

3.2 에이전트 지식처리의 개선 방침

본 절에서는 2 장에서 서술한 문제점들을 해결하기 위해서 다음과 같은 방침으로 지식처리의 아키텍처를 새롭게 구성한다.

(1) 기존의 VCM 에이전트 내 INTER 지식의 구조를 개선한 새로운 아키텍처를 설계한다.
(2) 해결 전략을 적절하게 바꾸기 위한 지식과 QoS 파라미터 조정 알고리즘을 도입한다[8].
(3) 에이전트를 설계 할 때에 효율성, 재 이용성 그리고 기술성(記述性)을 고려하고, 이상의 지식 및 기구를 모듈화 한다.

3.3 새로운 INTER 지식의 설계 및 FVCS에의 적용

기존의 ADIPS를 기반으로 구성한 FVCS 내 VCM 에이전트는 지식 모듈(Knowledge Module), 협조 모듈(Cooperation Module) 그리고 베이스 프로세스 인터페이스 모듈(Base Process Interface Module)로 구성되어 있다. 이들 모듈 중에서 지식 모듈은 지식 처리를 하는데 있어서 중요한 역할을 담당하며 에이전트 간 협조에 의해서 실행된다. 지식 모듈은 세분화 되어 TASK, PROC, BID, RECOV, CHANGE, INTER 지식으로 구성 되어 있다[9,10]. 특히, INTER 지식은 다른 에이전트와의 교섭, 평가 그리고 다른 에이전트로부터 동작 요구에 대한 처리를 담당한다. 본 절에서는 INTER 지식을 새롭게 구성한다.

3.3.1 QoS 파라미터 조정 기법을 적용한 T-INTER 지식의 설계

3.2절의 개선방침에 의해서 기존 에이전트 내 INTER 지식을 구체화 하고 새롭게 설계한 T-INTER 아키텍처(Tuning-INTER Architecture)를 그림 2에 나타낸다. T-INTER 내 각 모듈에 대한 설명은 다음과 같다.

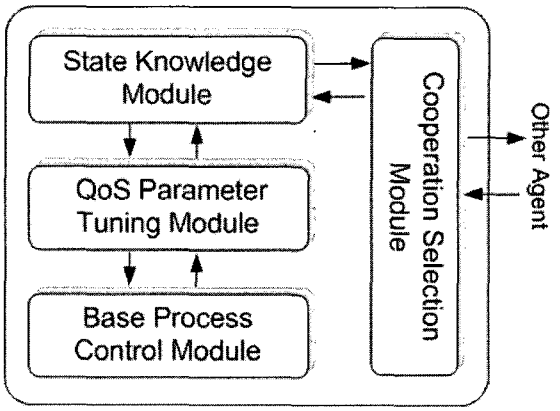


그림 2. T-INTER 아키텍처
Fig. 2. T-INTER architecture

(1) 협조 선택 모듈 (Cooperation Selection Module: CSM): 협조 선택 모듈은 발생한 문제의 성질이나 협조적 문제 해결 중의 목적 달성도, 문제해결 기한 등에 대하여 결정하며, 다른 에이전트와 메시지를 교환하고 협조하는 역할을 수행한다. 또한, 결정된 값들은 상태 지식 모듈에 저장하고, QoS 파라미터 조정 모듈로 보내서 조정을 수행하도록 한다.

(2) 상태 지식 모듈 (State Knowledge Module: SKM): 상태 지식 모듈은 저장소(Repository) 내에 클래스 에이전트 (Class Agents) 와 화상회의 시스템 제어에 이용되는 값 및 에이전트 간 협조에 의해서 실행된 QoS의 상태 지식(State Knowledges)값들을 에이전트 기술 언어 ADIPS/L[9]의 형식으로 저장하고 있다. 화상회의 중 발생한 상황에 따라서 저장소 내 클래스 에이전트들을 조립하여 인스턴스 에이전트 (Instance Agents)를 생성하며, 또한 지식정보를 다른 모듈에 제공하는 역할을 수행한다. 그림 3은 상태 지식 모듈 내 구성 및 지식 처리 상황을 나타내고 있는데 조립 환경(Assemble Environment) 및 저장소로 분할되어 있다.

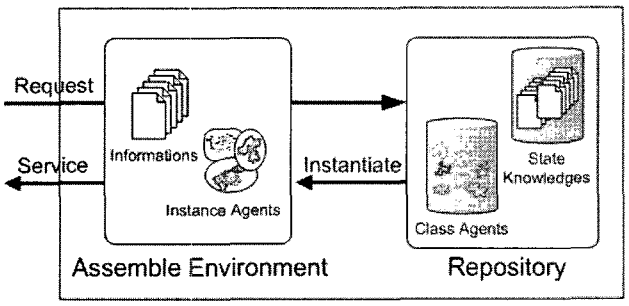


그림 3. 상태 지식 모듈
Fig. 3. State Knowledge Module

(3) QoS 파라미터 조정 모듈 (QoS Parameter Tuning Module: QPTM): 본 모듈은 CSM으로부터의 요구된 QoS (CPU 부하, QoS 파라미터 등)들을 실제로 조정하여 결정하는 역할을 수행한다. 여기에서 결정된 값들의 정보는 베이스 프로세스 제어 모듈로 보내어 진다. 이 모듈에서는 2장에서 기술하였던 변화에 대하여 유연한 대응 알고리즘의 부족 문제를 개선하기 위한 한 방법으로서 다음과 같은 QoS 파라미터 조정 기법을 적용한다.

- QoS 파라미터 조정 기법: QoS 파라미터는 화상회의 이용자의 파라미터 우선도에 따라서 차별하게 조정하여 현재의 자원상황 (예를 들면 CPU 부하, 네트워크 대역폭 등)을 원하는 상태에 맞도록 시도한다. 또한, 프레임 비율 (frame rate), 화질(quality), 해상도(resolution)와 같은 QoS 파라미터들을 조정하는데 있어서 그 조정 폭은 반복 (iteration)하는 동안에 오실레이션 하는 상태를 피하기 위해서 점차로 줄인다. 시간 t에서 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 가능 한 자원상황이 감소되고, 반면에 파라미터 값이 감소하면 사용 가능한 자원 상황이 증가된다.

$$y(t) = f(A(t)) \tag{1}$$

여기에서, $A(t) = [\, a_1(t),\ a_2(t),\ \cdots,\ a_i(t),\ \cdots,\ a_n(t)\,]$는 각 파라미터들의 집합이고, $a_i(t)$는 시간 t에서 i의 파라미터 값이다. 그리고 시간 t에서 원하는 자원 상황 d와 현재의 자원상황 $y(t)$사이의 에러 $e(t)$는 다음과 같이 표현할 수 있다.

$$e(t) = |d - y(t)| \tag{2}$$

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 한번 반복해서 조정된 파라미터 값들의 집합을 $A(t+1) = [\, a_1(t+1),\ a_2(t+1),\ \cdots,\ a_i(t+1),\ \cdots,\ a_n(t+1)\,]$로 표현할 수 있다. 여기에서 괄호 내에 있는 1은 한번의 반복을 의미한다. 그리고 $a_i(t+1)$는 다음의 식 (3)과 같이 표현할 수 있다.

$$a_i(t+1) = a_i(t) + \Delta a_i(t) \tag{3}$$

여기에서 $\Delta a_i(t)$는 반복함에 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.

$$\Delta a_i(t+1) = \rho_i . a_i(t) \tag{4}$$

여기에서 ρ_i는 조정자(scaling factor)이다. 조정자의 크기는 사용자의 파라미터 우선순위에 의해서 결정된다. 만약 어떤 파라미터 조정자 값이 작으면 사용자는 화상회의에 사용되는 다른 파라미터 보다 더 높은 우선도를 제공한다. 조정자 값은 사용자의 파라미터 a_i의 우선도 또는 선호도에 의해서 다르게 된다. 예를 들면 만약 파라미터들의 우선순위가 가장 낮은 것에서 가장 높은 $a_1(t),\ a_2(t),\ \cdots,\ a_i(t),\ \cdots,\ a_n(t)$순서로 되었을 때 조정자 값은 $\rho_n(t),\ \rho_{n-1}(t),\ \cdots,\ \rho_i(t),\ \cdots,\ \rho_1(t)$와 같이 높은 값에서 낮아지는 값 순서로 된다. 파라미터의 우선에 따른 조정자 값은 임의로 결정할 수 있지만 이상적인 조정자 값은 반복된 실험 경험에 의해서 결정될 수 있다. 또한 식 (3)과 식(4)에서 조정자의 부호는 파라미터를 찾는 공간에서의 최적한 파리미터 값을 찾는 방향을 의미하고, 최적의 값을 찾기 위해서 동적으로 조정된다. 예를 들면, 만약 조정자 부호가 플러스이면 파라미터는 증가되는 쪽으로, 마이너스이면 파라미터가 감소되는 쪽으로 조정한다. 한편, 파라미터 조정에 따른 현재의 자원상황 및 에러는 식 (5) 및 (6)과 같이 표현된다.

$$y(t+1) = f(A(t+1)) \tag{5}$$

$$e(t+1) = |d - y(t+1)| \tag{6}$$

여기에서 만약 에러 $e(t+1)$가 만족한 값에 도달 되었을 경우는 파라미터 조정 프로세스를 마친다. 그렇지 않으면 에러 값이 만족한 값에 근접할 때까지 반복함으로서 조정 프로세스를 계속 수행한다. 식 (4)에서 (6)까지로 본 바와 같이 조정자 값은 프로세스 동작 중에 중요한 역할을 한다. 만약 조정자 값이 너무 크면 조정 속도는 빠르지만 파라미터의 최적 값으로 접근하지 않고 오실레이션 한다. 또한 조정자 값이 너무 작으면 조정 속도가 느리고 최적의 값에 도달하는 시간이 많이 소요된다. 이런 문제를 피하기 위하여 식 (7)에 나타낸 바와 같이 반복 횟수가 증가 할수록 파라미터 조정의 폭을 감소시킨다.

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

여기에서 k는 반복의 수이다. 본 기법에 의해서 CPU 부하, 네트워크 대역폭, 그리고 각 파라미터 값들은 계속 수정되어, 화상회의 시스템이 지속적 또는 유연하게 동작하도록 하는 역할을 한다.

(4) 베이스 프로세스 제어 모듈 (Base Process Control Module: BPCM): 본 모듈은 QPTM에서 결정된 값으로 베이스 프로세스를 제어하는 역할을 수행한다. 센서 에이전트 그룹(Sensor Agents: CPU-Monitor, Net-Monitor)으로부터 자원상황 정보를 분석, 평가한다. 그리고 QPTM에서 결정된 최적의 파라미터 값으로 화상회의 프로세스를 직접 제어하는 서비스 에이전트 그룹(Service Agents: Video, Audio, Whiteboard)에 제어동작 요구를 실행하는 것으로 이용자 요구를 가능한 한 충족시키면서 안정한 시스템 동작을 실현한다. 센서 에이전트로부터의 QoS 정보를 SKM에 보내는 역할도 수행한다.

앞에서 서술한 INTER 지식의 구체화 및 이것을 이용하는 것에 의해서 다음과 같은 이점을 얻을 수 있다.

- 협조 선택 모듈(CSM)은 문제가 발생할 때에 VCM 에이전트 간 협조에 의해서 문제에 적합한 판단을 하기 때문에 발생한 문제의 특징에 대응하여 유연한 대처를 수행하는 것이 가능하다.
- 각 VCM 에이전트 내 QPTM은 CSM으로부터의 협조 상황 정보를 가지고 QoS 파라미터 조정을 수행하여 적절한 파라미터 값을 지정하기 때문에 기존의 시스템보다 문제해결 중의 변화에 대응하여 유연하고 정확한 처리를 수행 가능하다.
- 제안 아키텍처는 각 모듈의 역할이 분담되어 있어서 독립적으로 서술할 수 있다. 따라서 에이전트의 기술성(記述性,), 가독성(可讀性), 그리고 모듈의 재 이용성 향상을 기대할 수 있다.

3.3.2 FVCS에 적용 및 에이전트 간 동작

본 절에서는 3.3.1 절에서 설계한 T-INTER 아키텍처를 포함한 VCM 에이전트를 그림 1의 FVCS에 적용한다. 또한 FVCS에 적용하였을 때 사용되는 에이전트 간 프로토콜에 관하여 기술한다. 그리고 VCM 에이전트와 다른 에이전트 및 VCM 에이전트 간 동작상황에 대하여 서술한다. T-INTER를 도입한 VCM 에이전트 구조는 그림 4와 같이 나타낸다.

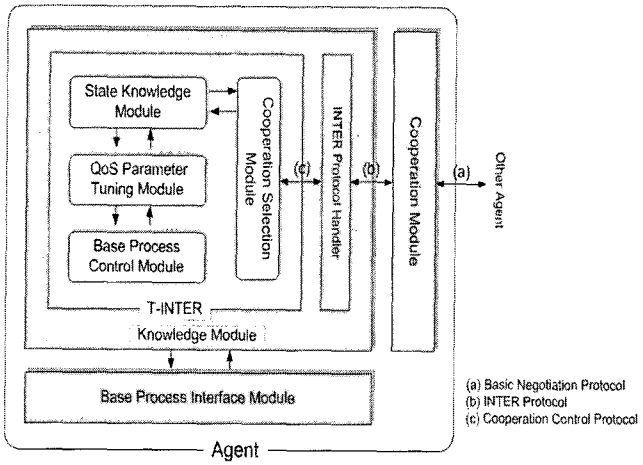


그림 4. VCM 에이전트 아키텍처
Fig. 4. VCM agent architecture

VCM 에이전트에서 협조모듈(Cooperation Module: CM) 및 베이스 프로세스 인터페이스 모듈(Base Process Interface Module: BPIM)은 기존의 ADIPS 내 에이전트 모듈을 사용하였다.

한편 위에서 서술한 모듈 간에는 다음과 같은 프로토콜에 따라서 메시지 교환을 한다.

(a) 기본 교섭 프로토콜(Basic Negotiation Protocol)

협조 모듈(Cooperation Module) 사이에서 이루어지는 메시지 교환용 프로토콜로서 ADIPS 프레임워크[9]에 의해 정의되어 있다. 표 1에 본 프로토콜의 퍼포머티브, 즉 에이전트 간 통신 사항을 표시한다. 또한, 표 1에서 "S"는 메시지의 송신측(Sender), "R"은 메시지의 수신측(Receiver)을 표시한다.

표 1 ADIPS 기본 교섭 프로토콜
Table 1. ADIPS basic negotiation protocol

Performative	Summary
Tell	S sends some objects to R
Ask	S requests some objects to R

(b) INTER프로토콜(INTER Protocol)

INTER지식에 기초로 에이전트 간 교섭에서 사용되는 기본적인 프로토콜 (표2)이다.

표 2 에이전트 작업 공간 내 INTER 프로토콜
Table 2. INTER Protocol in Agent Workspace

Performative	Summary
RequestAction	S requests R to do something
Acceptance	S sends the RequestAction message to R
Refusal	S refuses the RequestAction message
RequestConf	S requests the confirmation to R
AcceptConf	S accepts the confirmation
RefusalConf	S refuses the confirmation
Report	S sends some information to R

(c) 협조 제어 프로토콜(Cooperation Control Protocol)

에이전트 간에서 에이전트의 협조전략을 결정하기 위해 사용되는 프로토콜 (표3)이다.

표 3 VCM 에이전트 간 협조 프로토콜
Table 3. Cooperation control protocol between VCM agents

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

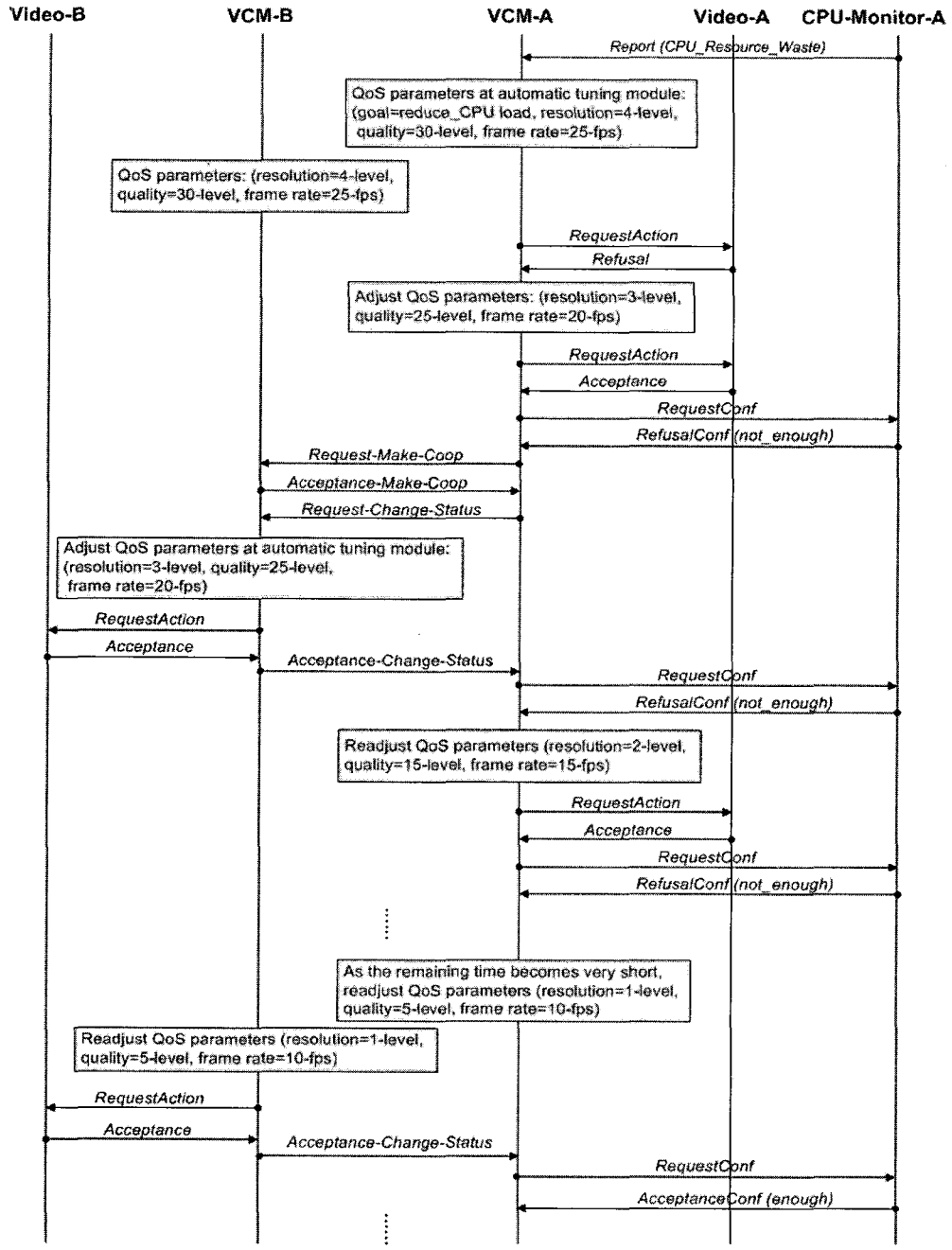


그림 5. T-INTER 아키텍처에 의한 에이전트의 협조 동작 예
Fig. 5. An example of agent's cooperation with T-INTER architecture

위의 VCM 에이전트 및 각 프로토콜을 이용하여 FVCS에 적용한 예를 설명하면 다음과 같다. 화상회의 시스템 동작 중에 WS-A의 CPU-Monitor-A 에이전트로부터 자원상황 변화에 대한 메시지가 발생하였을 경우 에이전트 간 협조 동작 처리 과정의 예를 그림 5에 나타낸다. 한편, 이용자 요구 변화에 대한 에이전트 간 협조 동작 처리도 이와 같은 협조 과정을 통하여 해결 할 수 있다.

(1) 자원상황 변화의 인식: CPU-Monitor-A가 자원상황 변화의 허용 범위로부터의 벗어남을 인식하면, VCM-A에 리포트(*Report*) 메시지로 보고한다.

(2) 초기 파라미터의 결정: VCM-A는 문제 해결을 위한 협조 전략을 결정한다. VCM-A내 CSM이 문제해결의 기한에 시간적인 여유가 있음을 판단하고 QPTM에 파라미터 값을 요구하면, QPTM은 각 QoS 파라미터 값을 결정한다.

(3) 작업공간 내 파라미터 조정: VCM-A는 결정된 파라미터 값에 의해서 Video-A에 *RequestAction* 에 의해서 파라미터 제어를 요구한다. 만약 Video-A가 *Refusal* 메시지를 보내면 VCM-A는 에이전트 내 모듈 QPTM에서 파라미터 값을 재조정하여 다시 요구한다. Video-A로부터 *Acceptance* 메시지가 오면 CPU-Monitor-A에 *RequestConf* 메시지를 보내 CPU 자원 상황을 의뢰한다. 만약 CPU-Monitor-A가 *RefusalConf(not_enough)* 메시지를 보내면 VCM-A는

WS-B측의 VCM-B에 협조를 요구한다.

(4) VCM 에이전트 간 협조에 의한 문제해결: VCM-A는 자신의 WS 내 에이전트 작업공간에서 CPU 자원 해방을 수행할 수 없을 때 WS-B측의 VCM-B로 *Request-Make-Coop* 메시지를 보낸다. VCM-B로부터 *Acceptance-Make-Coop* 메시지를 받으면 *Request-Change-Status* 메시지를 보낸다. VCM-B는 에이전트 내부 QPTM에서 파라미터 값을 조정하여, Video-B에 *RequestAction* 메시지를 보내 파라미터 값을 제어한다. VCM-B는 변화된 상황을 *Acceptance-Change-Status* 메시지에 의해서 VCM-A에 보내고, VCM-A는 CPU-Monitor-A에 자원 상황을 의뢰한다. 만약 CPU-Monitor-A로부터 *RefusalConf(not_enough)* 메시지를 받으면 VCM-A는 QPTM에서 파라미터 값을 재조정하여 CPU상황을 의뢰한다. 이와 같은 상황을 반복하여 VCM-A 및 VCM-B는 파라미터 값을 조정하면서 CPU-Monitor-A의 CPU 상황을 알아본다.

(5) 제한 시간으로의 근접: 만약 VCM 에이전트 간 협조에 의해서 파라미터 값을 조정 중에 제한 시간이 다가올 경우, 각 VCM 에이전트는 최저한의 파라미터 값으로 조정하여 CPU 자원 해방을 시도한다.

(6) 협조상태의 종료: 위에서 기술한 것처럼 VCM 에이전트 간 협조에 의해서 각 QoS 파라미터 값의 변경 동작이 반복되어 그 결과로서 CPU자원의 해방에 성공한 경우에는 협조상태가 해결된다.

4. 실험 및 평가

4.1 실험환경

본 논문에서 제안한 T-INTER 아키텍처의 유용성을 확인하기 위해서 그림 6의 환경에 적용하여 실험하였다. 구체적으로는 기존의 멀티에이전트 기반 프레임워크인 ADIPS [9,10]에서 문제가 되었던 저장소 및 지식 부분을 새롭게 설계한 VCM 에이전트를 플렉시블 화상회의 시스템(그림 1)의 VCM-A 및 VCM-B에 적용하였다. 서비스 에이전트가 조정하는 각 파라미터 조정을 위해서 기존에 개발된 화상회의 용 vic[3]를 이용하였는데, 서비스 에이전트 내 에이전트(Video-A, Video-B)는 vic의 정보를 VCM 에이전트에 제공하면서 각 파라미터 값 조정을 수행한다. 또한, CPU-Monitor 에이전트는 CPU 체크를 위해서 Unix 연산 시스템 표준 명령어인 "sar"를 이용하여 CPU 상황을 판단하고, 획득된 정보를 VCM에 제공하였다. 에이전트의 지식 모듈은 Tcl/Tk[12]를 이용하여 기술하였으며, 이용자 에이전트 및 센서 에이전트 그리고 서비스 에이전트 지식 또한 Tcl/Tk를 이용하여 기술하였다. 에이전트를 기술에 있어서 프로그램 총 사이즈는 약 2,300 스텝이었다. 화상회의에 사용된 각각의 하드웨어는 에이전트 작업 공간(Agent Workspace) 및 화상회의용 단말로서 SPARCstation Ultra1 (CPU clock 200MHz)을 이용하였다.

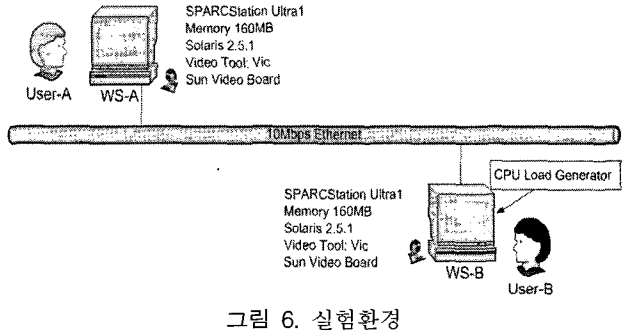
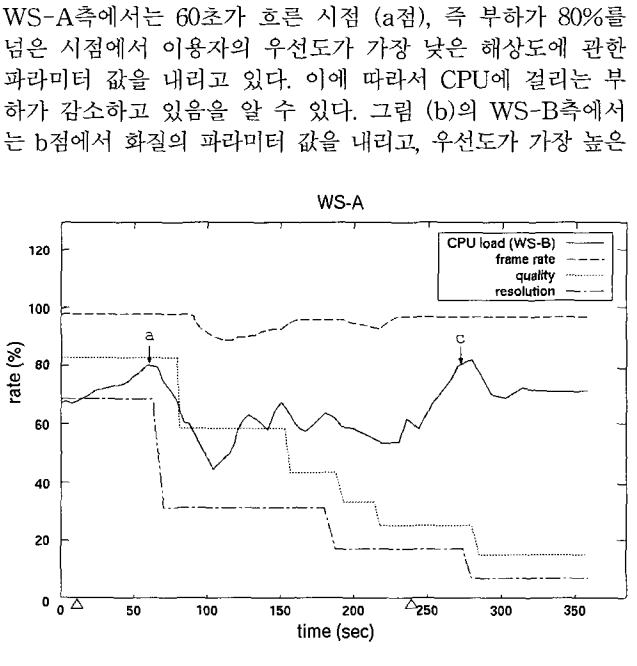


그림 6. 실험환경
Fig. 6. Experiment environment

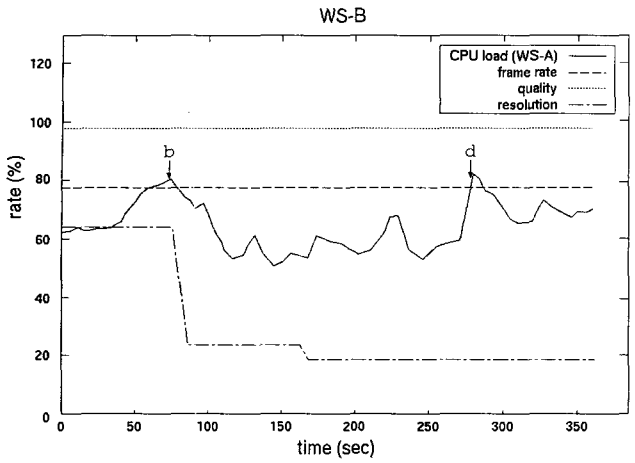
4.2 자원변동 및 이용자 요구변화에 대한 시스템의 유연성 평가

화상회의 중에 발생하는 CPU 부하 및 이용자 요구가 동적으로 변화하는 상황에서 제안 아키텍처를 기반으로 구축한 FVCS의 유연성 동작을 관찰하였다. 그림 6의 실험환경에서와 같이 먼저, 시스템 동작 중에 이용자 B(User-B)측 단말(WS-B)의 CPU에 대하여 부하 조정용 프로세스(CPU Load Generator)를 기동하여 부하를 주었을 경우 이용자 A(또는 이용자 B)에 제공되는 동화상에 대한 QoS 파라미터 동작 상황을 관찰하였다. 여기에서, WS-A의 User-A는 동화상에 대한 QoS 요구의 우선도는 프레임 비율(frame rate)이 가장 높고, 다음에 화질(quality) 그리고 해상도(resolution)순으로 하였다. 또한, WS-B측 이용자의 동화상에 대한 우선도는 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 임으로 정한 값인 부하가 80%를 넘었을 때에 감지하는 것으로 하였다. 목표로 설정한 CPU 부하는 60% 이하로 하였다.

그림 7에서는 기존의 룰 기반 지식을 적용한 FVCS의 실험 결과를 나타내고 있다[4-7]. 화상회의 중에 10초가 지난 시점(첫 번째 △점)에서 WS-B측의 CPU 부하 발생기(CPU Load Generator)에 의해서 부하를 걸었을 경우, 그림 (a)의 WS-A측에서는 60초가 흐른 시점 (a점), 즉 부하가 80%를 넘은 시점에서 이용자의 우선도가 가장 낮은 해상도에 관한 파라미터 값을 내리고 있다. 이에 따라서 CPU에 걸리는 부하가 감소하고 있음을 알 수 있다. 그림 (b)의 WS-B측에서는 b점에서 화질의 파라미터 값을 내리고, 우선도가 가장 높은



(a) WS-A에서의 QoS 파라미터 변동
(a) Change of QoS parameters at WS-A

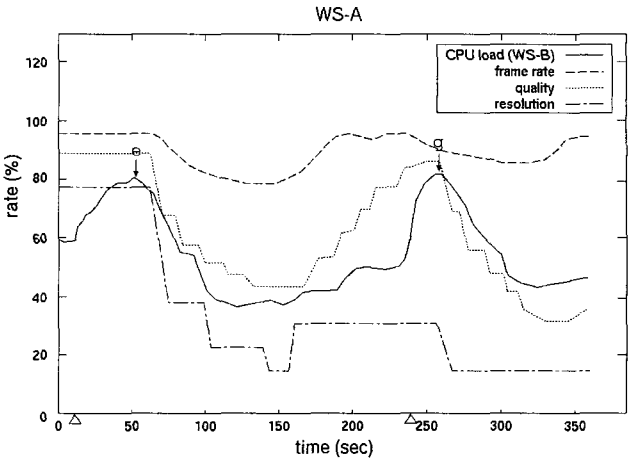


(b) WS-B에서의 QoS파라미터 변동
(b) Change of QoS parameters at WS-B
그림 7. CPU 변동에 대한 기존 FVCS의 대응
Fig. 7. Behavior of existing FVCS against CPU variation

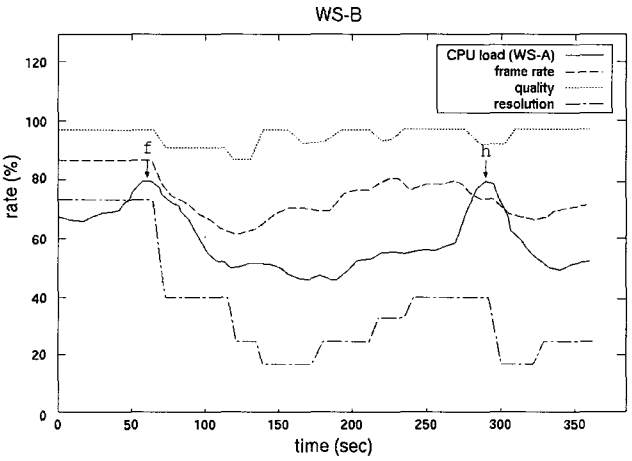
프레임 비율도 점차로 내리고 있다. 그림 (b)의 WS-B측에서도 우선도가 낮은 해상도를 먼저 최대한 내리고 프레임 비율 파라미터 값을 내려서 CPU에 걸리는 부하를 감소시키고 있다. CPU 부하가 어느 정도 내려지고 있을 때 240초 지점(두 번째 △점)에서 WS-B 측의 CPU에 부하를 걸었다. 부하가80%를 넘은 시점(그림 (a): c점, 그림 (b): d점)에서 각 파라미터 값들의 변화가 많지 않음을 나타내고 있다. 또한 CPU부하가 목표 설정치(60%) 아래로 내려 지지 않고 있음을 알 수 있다.

이와 같은 원인은 첫째, 기존의 전문가의 경험에 의해 정해진 룰 지식에 의해서 구성된 화상회의 시스템에서는 그 지식의 선택에 의해서CPU의 부하를 어느 정도 감소시킬 수는 있지만 지식 선택의 폭이 정해져 있어서 더 이상의 조정이 어려움을 알 수 있다. 둘째로는 급격한 자원상황 변화에 대한 대응 방안이 부족하였음을 알 수 있다. 셋째로는 에이전트 간 협조에 사용되는 프로토콜 지식이 적절하지 못하였음에 기인한다.

한편, 그림 8에서는 제안 QoS 파라미터 자동 조정 알고리즘 및 아키텍처를 적용한 FVCS의 CPU 자원 변동에 대하여 대응 실험 결과를 나타내고 있다. 실험 조건은 그림 6과 같은 상황에서 실시하였다. 각 화상회의 이용자의 동화상에 대한 우선도에 순위에 따라서 QPTM 내의 QoS 파라미터 값을 조정하는 조정자의 초기 값은 WS-A측은 a_1= 0.196(frame rate), a_2=0.324(quality), 그리고 a_3= 0.525(resolution)로 정하였다. WS-B측은 b_1=0.182(quality), b_2=0.351(frame rate), 그리고 b_3=0.463(resolution)으로 정하였다. 이 초기 조정자 값들은 반복된 실험에 의해서 적절한 값으로 정하였다. 화상회의 중에 10초가 흐른 시점(첫 번째 △점)에 WS-B측의 CPU 부하 발생기(CPU Load Generator)로 부하를 걸었을 경우, 그림 (a)의 WS-A측에서는 55초가 흐른 시점(e점), 즉 CPU 부하가 80%를 넘은 시점에서 각각의 서로 다르게 조정된 파라미터 값을 내리면서 동시에 부하를 내리고 있다. 그림 (b)의 WS-B측에서는 65초가 흐른 시점(f점)에서 각각 조정된 파라미터 값을 내리면서 동시에 부하를 감소시키고 있다. CPU 부하가 어느 정도 내려지고 있을 때 240초 지점(두 번째 △점)에서 WS-B 측의 CPU에 부하를 걸었다. 부하가 80%를 넘은 시점(그림 (a): g점, 그림 (b): h점)에서 각 파라미터들은 서로 다르게 조정된 값에 의해서 내려지면서 CPU 부하를 아래로 내리고 있음을 알 수 있다.



(a) WS-A에서의 QoS 파라미터 변동
(a) Change of QoS parameters at WS-A

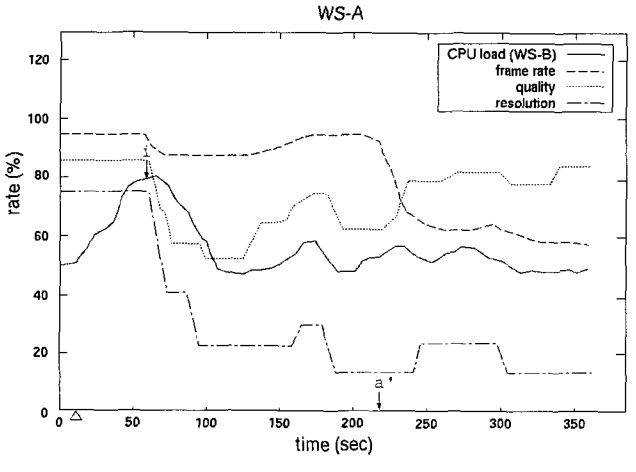


(b) WS-B에서의 QoS파라미터 변동
(b) Change of QoS parameters at WS-B
그림 8. CPU 변동에 대한 FVCS의 대응
Fig. 8. Behavior of FVCS against CPU variation

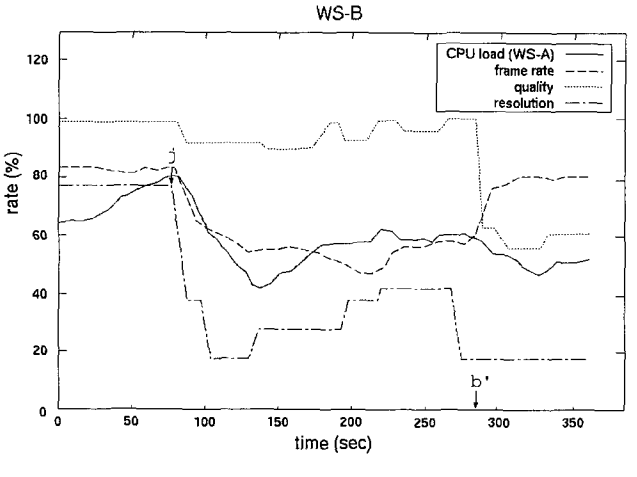
그림 8의 결과에서 보여 지는 바와 같이 CPU 부하를 감소시키기 위한 CSM의 에이전트 간 협조 및 QPTM에서 QoS 조정 기법이 반복적으로 잘 동작하여 결국 CPU 부하를 지속적으로 목표 설정치(60%) 이하로 조정하고 있음을 알 수 있다.

그림 9 에서도 그림 8과 같이 제안한 아키텍처 및 QoS 파라미터 조정 기법을 적용하여 CPU 자원 변동에 대한 대응 실험 결과를 나타내고 있다. 실험 조건은 처음에는 그림 7 및 8과 같은 상황에서 실시하였으며 이용자의 동화상에 대한 우선도가 변화 했을 경우를 추가적으로 실험하였다. 각 화상회의 이용자의 동화상에 대한 우선도에 순위에 따라서 QPTM 내의 QoS 파라미터 값을 조정하는 조정자의 초기 값은 WS-A측은 a_1=0.159(frame rate), a_2=0.347(quality), 그리고 a_3=0.475(resolution)로 정하였다. WS-B측은 b_1=

0.146(quality), b_2=0.383(frame rate), 그리고 b_3=0.556(resolution)으로 정하였다. 화상회의 중 10초가 흐른 시점(△점)에 WS-B측의 CPU 부하 발생기(CPU Load Generator)에 의해서 부하를 걸었을 경우, 그림 (a)의 WS-A측에서는 60초가 흐른 시점(i점), 즉 CPU 부하가 80%를 넘은 시점에서 각각의 서로 다르게 조정된 파라미터 값을 내리면서 동시에 부하를 내리고 있다. 그림 (b)의 WS-B측에서는 75초가 흐른 시점(j점)에서 각각 조정된 파라미터 값을 내리면서 동시에 부하를 감소시키고 있다. 한편, CPU 부하가 어느 정도 목표 설정치 이하로 조정되고 있을 때 WS-A 측의 이용자인 User-A가 동화상에 대한 QoS 파라미터의 우선도를 a′ 시점(215초 지점)에서 화질, 프레임 비율 그리고 해상도 순으로 변경하고 싶다는 메시지를 VCM-A에 보내면 CPU부하를 목표 설정치 이하로 유지하면서, 이용자의 우선도에 따른 파라미터 변경을 시도하고 있음을 알 수 있다. 또한, WS-B 측의 이용자 (User-B)도 파라미터 우선도를 b′ 시점(285초 지점)에서 프레임 비율, 화질 그리고 해상도 순으로 변경하고 싶다는 메시지를 VCM-B에 보내면 VCM-B 또한 CPU부하를 목표치 이하로 유지하면서 파라미터 변경을 시도하고 있다. 그림 8의 결과에서 보여 지는 바와 같이 CPU 부하를 감소시키기 위한 CSM의 에이전트 간 협조 및 QPTM에서 자동 QoS 조정 기법이 반복적으로 잘 동작하여 결국 CPU 부하를 지속적으로 목표 설정치 이하로 조정하고 있다. 그리고 부하가 변동하고 있는 상황에서 이용자의 동화상에 대한 QoS 우선도 변경요구가 있을 경우에도 QoS 파라미터 우선순위를 바꾸면서, CPU 자원 확보를 하고 있다.



(a) WS-A에서의 QoS 파라미터 변동
(a) Change of QoS parameters at WS-A



(b) WS-B에서의 QoS파라미터 변동
(b) Change of QoS parameters at WS-B

그림 9. CPU 변동 및 이용자 요구 변화에 대한 대응
Fig. 9. Behavior of FVCS against CPU variation and user request change

4.3 고찰

기존의 룰 지식 기반 에이전트로 구성된 FVCS에서는 에이전트가 미리 정해진 룰에 한정된 지식의 선택으로 CPU의 부하를 어느 정도 유연하게 감소를 할 수 있지만 다양하게 변하는 자원상황에 대하여 동적으로 선택하는 알고리즘이 부족하였기 때문에 결국 문제해결의 복잡성에 따라서 에이전트 그룹이 유연하게 협조하는 것은 곤란한 경우가 있었다. 또한, 에이전트 간 협조에 사용되는 프로토콜 지식이 적절하지 못하였음을 알 수 있다.

그러나, 제안 아키텍처에서는 협조문제 해결의 진행상황에 대응할 수 있도록 개선된 프로토콜을 이용하고 이와 함께 각 CM간의 원활한 협조가 이루어 졌다. 그리고 CSM에서 원하는 QoS 요구 값을 자동으로 조정하는 기법을 QPTM에 적용하고, 모듈 간의 메시지 교환에 의해 문제 발생에 적절하게 대응하는 것이 확인 되었다. 이들의 동작은 화상 회의 시스템 운용 전문가의 한정된 지식을 기초로 동작한다는 기존의 제어전략의 문제점 해결하는 방법에 있어서 유용한 한 방법으로 생각된다.

본 논문에서는 "유연성"을 제한된 자원상황이나 까다로운 이용자 요구 등의 다양한 제약조건이 주어질 때 시스템의 자율적 또한 동적인 조정능력으로 취급하였다. 결국 제안 아키텍처의 적용에 의한 FVCS는 다양하고, 엄격한 QoS 요구가 있을 경우에도 대응이 가능한 것이 확인되어서 기존의 룰 기반 FVCS와 비교하여 시스템의 고도한 유연성이 실현하였다고 생각된다. 더불어 제안 자동 QoS 파라미터 조정기법을 적용한 매니저 에이전트 아키텍처 내 각 기구를 기능적으로 독립한 모듈로서 구축하는 수법은 다양한 환경에 특화한 화상 회의 시스템에 VCM 에이전트를 시작으로 VoD(Video on Demand)와 같은 다른 멀티미디어 통신 시스템을 에이전트 기반으로 구축할 때 적용 가능하다고 생각된다. 또한 이와 관련된 시스템 개발에 있어서 효율성의 향상을 기대할 수 있다.

5. 결 론

본 논문에서는 에이전트 기식기반 화상회의 시스템의 서비스 조정 능력 및 시스템의 유연성을 실현하기 위한 에이전트의 아키텍처를 제안하였다. 또한, 발생한 문제의 성질이나 문제해결의 진행상황을 고려하면서 전략에 따라서 QoS를 자동으로 조정하여 유연하게 문제 상황에 대처 가능하게 하는 QoS 파라미터 조정기법을 에이전트의 지식부분 구성에 적용하였다. 또한 제안한 아키텍처를 이용하여 FVCS에서 실험을 실행하였다. 그 결과 FVCS에 있어서 다양하고, 긴급을 요하는 조건의 경우에도 에이전트 간 협조가 효과적으로 실행되었으며 기존의 룰 지식 선택 방법에 의한 FVCS 보다

더욱 더 유연하게 QoS 제어가 가능했음을 보여주었다.

결론적으로 기존의 VCS를 이용한 화상회의를 할 경우 발생하는 시스템의 내·외적 변동에 의해서 시스템 서로 간 조정이 불가능하여 극단적으로 동화상이 정지될 수 있다. 그러나 본 논문에서는 시스템 간 협조를 할 수 있는 에이전트 기술을 적용하여 시스템이 상황에 따라서 유연하게 대처할 수 있고, 동화상이 정지되는 것을 방지하면서 시스템이 유연하게 동작하게 되었음을 증명하였다. 또한, 일반 이용자를 지향한 화상회의 시스템을 구축하는데 있어서 본 시스템의 유용함을 알 수 있다.

향후 과제로서는 CPU 및 네트워크 자원상황을 동시에 고려한 시스템의 구성이 필요하다. 또한 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.

참 고 문 헌

[1] T. Turletti and C. Huitema, "Videoconferencing on the Internet", IEEE/ACM Trans. on Networking, Vol. 4, No. 3, pp. 340-351, 1996.

[2] J. Bolliger and T. Gross, "A Framework-Based Approach to the Development of Network-Aware Applications", IEEE Trans. on Software Engineering, Vol. 24, No. 5, 1998.

[3] S. MaCanne and V. Jacobson, "vic: a flexible framework for packet video", ACM Multimedia, pp. 511-522, Nov. 1995.

[4] T. Suganuma, T. Kinoshita, K. Sugawara, and N. Shiratori, "Flexible Videoconference System based on ADIPS Framework", Proc. of the 3rd International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM98), pp. 83-100, 1998.

[5] S. D. Lee, T. Karahashi, T. Suganuma, T. Kinoshita, and N. Shiratori, "Construction and Evaluation of Agent Domain Knowledge for Flexible Videoconference System", IEICEJ, vol. J83-B, pp. 195-206, 2000.

[6] T. Suganuma, S. Imai, T. Kinoshita, and N. Shiratori, "A QoS Control Mechanism Using Knowledge-Based Multiagent Framework", IEICE Trans. Information and Systems, Vol. E86-D, No. 8, pp. 1344-1355, 2003.

[7] S. D. Lee and D. S. Han, "Multiagent based Adaptive QoS Control Mechanism in Flexible Videoconference System", ICACT 2004, Vol. II, pp. 745-750, 2004.

[8] S. G. Kang and C. ISik, "Partially connected neural network structured by input types for mapping problems", IEEE Trans. on Neural Networks, Vol. 16, No. 1, pp. 175-184, 2005.

[9] S. Fujita, H. Hara, K. Sugawara, T. Kinoshita, and N. Shiratori, "Agent-based design model of adaptive distributed systems", Applied Intelligence, Vol. 9, No. 1, pp. 57-70, July/Aug. 1998.

[10] T. Kinoshita and K. Sugawara, "ADIPS Framework for Flexible Distributed Systems", Springer-Verlag Lecture Notes in AI, 1599, pp. 18-32, 1998.

[11] DASH-Distributed Agent System Based on Hybrid Architecture. [On-line] http://www.agent-town.com/dash/index.html

[12] J. K. Outsterhout, "Tcl and the Tk Toolkit", Addison-Wesley, 1994.

저 자 소 개

이성독(Sung-Doke Lee)

1988 : 전북대학교 전자공학과 (학사)
1991 : 전북대학교 전자공학과 (석사)
2002 : 일본 도호쿠대학교 정보과학연구과 (박사)
1991.5~1993.7 : 군산대학교 전기공학과 조교
2002.4~2003.3 : 일본 도호쿠대학교 전기통신연구소 연구원
2003.8~2005.7 : 한국정보통신대학교 공학부 계약교수
2005.8~현재 : 한국정보통신대학교 공학부 연구조교수

관심분야 : 지식표현, 에이전트공학, 멀티미디어시스템, 웹서비스, 바이오인포메틱스
E-mail : sdlee@icu.ac.kr

강상길(Sang-Gil Kang)

1989 : 성균관대학교 전자공학과 (학사)
1995 : Columbia University 전자공학과 (석사)
2002 : Syracuse University 전자공학과 (박사)
2003.4~2004.8 : 한국정보통신대학교 공학부 계약교수
2004.9~현재 : 수원대학교 정보공학대학 컴퓨터학과 전임강사

관심분야 : 인공지능, 멀티미디어, 데이터마이닝, 신호처리, 타임시리즈, 제어공학
E-mail : sgkang@mail.suwon.ac.kr