M-083

Knowledge Circulation Framework for Ubiquitous Communication Services : Development and Evaluation

Shintaro Imai† Atsushi Takeda‡ Takuo Suganuma§ Norio Shiratori§

1. Introduction

In ubiquitous information environment, service provisioning ability of a system is significantly limited due to lack of computational power of small devices/sensors, and limitations in availability and stability of wireless networks. In this environment, it is a big hurdle to give mechanisms to strategically control the ubiquitous services according to node statuses and network conditions in appropriate manner. Effective use of many kinds of the knowledge concerning diverse services and intricately changing environment is a challenging research topic.

To resolve such kind of the issue, it is essential to effective acquisition, maintenance, placement and reuse of the operational heuristics through the network. To do this, first of all, we have proposed a Knowledge Circulation Framework (KCF) [1], a framework that enables sharing the operational knowledge for effective service provisioning. In [1], we achieved an adequate QoS control by KCF in the situation that computational resource is greatly degraded. In this paper, we propose a design and an implementation of KCF, targeting ubiquitous information environment where computational and network resources are extremely unstable and limited. To overcome the hurdle in service provisioning in such kind of environment, selection of the knowledge acquired in the operation on similar environment is a core issue. In the proposed design, we developed KCF with a concept of knowledge-based multiagent system, and introduce a similarity metric to compare characteristics of different ubiquitous environments.

We applied KCF to the ubiquitous videoconference system (VCS), and evaluated it by some experiments. We verified our proposed system from a viewpoint of QoS to the user. As a result, frame-rate of the video increased from 15.5 fps to 18.2 fps, on average. On the other hand, variance of the fps decreased from 11.3 to 2.5. We describe details of this experimental result in section 4. From the experimental results, we confirmed that our system can provide better and more stable QoS, improving the adaptability to the various environments.

2. Knowledge Circulation Framework

2.1 Related works

There are several previous works dedicated to adaptive service control in application level on best effort networks and systems [2, 3]. In addition, more advanced QoS control mechanisms specialized to the unstable network environment such as wireless networks are emerging in recent years. In these works, the service control mechanisms, in range from simple algorithms to more intelligent functions, are basically hard-corded in the application. Since the service control mechanisms are tightly coupled with the service elements, they can not be separated off and reused, in spite of their advanced control ability.

†Research Fellow of the Japan Society for the Promotion of Science / Research Institute of Electrical Communication, Tohoku University

‡Department of Computer Science, Tohoku Bunka Gakuen University

§Research Institute of Electrical Communication / Graduate School of Information Science, Tohoku University

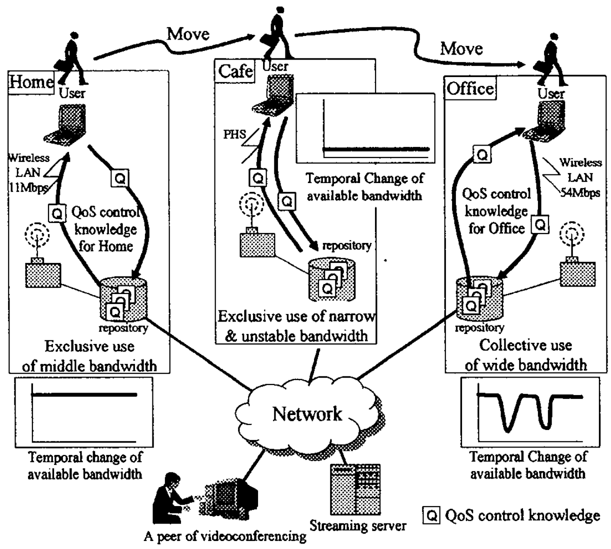


Fig. 1: QoS control knowledge circulated in ubiquitous information environments

2.2 Knowledge-based service control

One of the possible solutions to realize more adaptive service control is to acquire and reuse operational knowledge of human operators/designers in the system to control the quality of the services. To scale up the previous works described in the section 2.1, we have proposed Flexible Multimedia Communication Service (FMCS) with knowledge-based QoS control scheme [4]. In this scheme, QoS control knowledge is described in rule-type representation and is separated from the service elements. In addition, knowledge repository is newly introduced in this scheme. Using FMCS, designers of QoS control knowledge describe the knowledge and register it in the repository. Then the knowledge is instantiated and reused in run-time to control the QoS. Compare to the previous works, the range of QoS control ability was improved by FMCS.

2.3 Concept of KCF

Considering the problems described above, in this paper, we propose and develop Knowledge Circulation Framework (KCF) which enables dynamic circulation of the knowledge in ubiquitous information environments. Fig.1 shows the basic idea of our framework.

In KCF, the QoS control knowledge is circulated in the network via the repository. Using this framework, the knowledge, created during operation, such as sequence of change on parameters against the typical fluctuation of network resources, can be stored and reused in other similar situations. There are mainly two distinguished ideas in KCF as follows:

(1) Knowledge acquisition through operation: In related works described in previous sections, useful operational history and heuristics on service control, created during operation time, are basically put away. Using KCF, such heuristics are actively acquired and tailored for reuse.

(2) Reuse of acquired knowledge: In related works, service control mechanisms are statically embedded in application. While in FMCS, service control knowledge can be easily replaced, thus adaptability of the system is refined. However, there is no way to circulate the knowledge automatically. Using KCF, the acquired heuristics are effectively reused in many situations.

Employing this framework, service control ability would be greatly improved without any hard maintenances of knowledge repository against the drastic change of user requirement and environment, such as movement of the user among different environments.

3. Agent-based Design of KCF

3.1 Overview of KCF

KCF consists of three phases of circulation, i.e., knowledge acquisition, management, and placement phases.

Knowledge acquisition: In this phase, system acquires service control knowledge dynamically when some distinctive phenomenon occurs during service provisioning. These include knowledge concerning characteristics of the phenomenon, way to deal with the problem, history of actions to cope with the problem, effectiveness of the actions, etc.

Knowledge management: The service control knowledge acquired by Knowledge acquisition phase is simply stored in user terminals in disorganized manner. In this phase, the system performs accumulation and arrangement of the acquired knowledge based on the characteristics and the reusability of it.

Knowledge placement: In this phase, the system decides which and where service control knowledge should be places to provide service effectively, when the service is required under some conditions. The selection of the knowledge is important because ubiquitous information environment is very diverse. The service control is actually performed with the placed knowledge after this phase.

3.2 Design of agent architecture for KCF

In order to circulate the knowledge efficiently, KCF employs mobile agent technology to carry the knowledge. The agents play a role of carrier of the knowledge on the network. Moreover, we introduce agent repository as a warehouse and dispatcher of the knowledge.

We applied an agent architecture in [5] in KCF. The agent holds the knowledge in form of "Fact" and "Rule". Fig.2 shows an example of the rule type knowledge which an resource-monitoring agent has.

A rule is represented in the following form:

```
(rule Rule-name
   Condition-part (If-part)
   --> Action-part (Then-part))
```

When a conditional part is TRUE, then actions described in the action part are performed. Knowledge shown in Fig.2 is to inform a warning, when the agent detects degradation of computer resource.

3.3 Design of agent behavior for KCF

In this section we illustrate behavior of agents according to the phases described in section 3.1. Here, we circulate the following types of knowledge: knowledge about the environment, history of usage of QoS control knowledge, and heuristics on effects of the applied QoS control knowledge. The design of agent behavior for KCF is summarized in Fig.3. KCF is constructed based on multiagent framework with Agent Repository and Agent Workplace. The Repository is a server to store the agents, and the Workplace is an execution environment of agents on each workstation (WS).

```
(rule warning_service_down
    (record :record1 ?record1 :record2 ?record2 :record3 ?record3
            :record4 ?record4 :record5 ?record5)
    (threshold :threshold ?thre)
    (> ?record1 ?thre)
    (> ?record2 ?thre)
    (> ?record3 ?thre)
    (> ?record4 ?thre)
    (> ?record5 ?thre)
    (count :count ?cou) = ?cha
    (>= ?cou 10)
    (name :vcm ?vcm)
    -->
    (send :performative warning :to ?vcm
          :content (request_action :action service_down))
    (modify ?cha:count 0)
)
```

Fig. 2: Example of knowledge representation in an agent

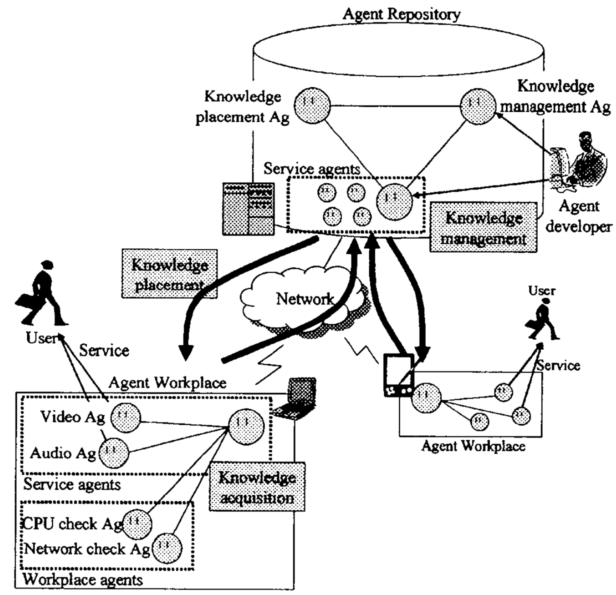


Fig. 3: Design of agent behavior for KCF

(1) Knowledge acquisition phase:

(1-a) Knowledge acquisition during service provisioning: Workplace agent permanently resides in Agent Workplace to monitor the status of computational resources, network resources and users. They acquire four kinds of information, they are, type and status of the resource, average of resource usage rate, standard deviation of the resource usage rate, and user requirement. On the other hand, Service agents observe achievement level of the user requirement. Here, the user requirement achievement level is derived from three metrics: a proportion of provided service level to user requirement level, stability of QoS, and a proportion of availability of the resource to its constraint. As a result, above three kinds of knowledge is acquired for circulation.

(1-b) Feed back to agent repository: When the service provisioning ends, Service manager agent, that is one of the Service agents, sends the knowledge acquired by Service/Workplace agents to

Knowledge management agent. Knowledge management agent resides in the Agent Repository.

(2) Knowledge management phase:
Knowledge management agent in the Repository classifies and accumulates knowledge that is fed back by Service manager agent. This operation in this phase includes the following operations:
(2-a) Knowledge association: Knowledge management agent associates the knowledge about user requirement achievement level and the knowledge about environment with the used QoS control knowledge.
(2-b) Knowledge integration: Knowledge management agent compares the knowledge that is fed back by Service manager agent with existing knowledge about knowledge about the environment and history of usage of QoS control knowledge. When the type and status of the resource match, and average of the resource usage rate, and standard deviation of the resource usage rate and user requirement are similar in some measure, history of usage of QoS control knowledge match, Knowledge management agent integrates these knowledge.

(3) Knowledge placement phase:
(3-a) Selection of Service agents: When a service is required, the Knowledge placement agent determines Service agents to be placed to the Agent Workplace using contract-net protocol [6].
(3-b) Knowledge hand-over: When Service agents to be placed to Agent Workplace are determined, the Knowledge management agent compares the knowledge about environment that has been accumulated, with the information about environment where the new service will start, in a similar way to (2-b) of Knowledge management phase. Knowledge management agent selects the knowledge that has the highest user requirement achievement level and that meets the following conditions necessary: (i) the type and status of the resource match, (ii) average of the resource usage rate, standard deviation of the resource usage rate and user requirement are similar in some measure. Knowledge management agent gives QoS control knowledge that included selected knowledge to Service agents.
(3-c) Service initiation: Service agents which are given knowledge and placed at an Agent Workplace start to offers the service, and the process returns to (1-a) of Knowledge acquisition phase.

4. Experiments

4.1 Implementation

We implemented KCF based on the design described in section 3.3 using DASH framework [7], one of the multiagent frameworks.

We developed seven agents: Knowledge management agent, Knowledge placement agent, VideoConference Manager (VCM) agent, Video agent, Netcheck agent, NetcheckVCS agent and UserVCS agent. Each agent has a computational base process (BP) to realize the function described in section 3.3. As for the video transmission/receiving software, we employed Java Media Framework (JMF), and embedded it as a BP in the Video agent.

4.2 Experiment environment and conditions

The configuration of experiment is as shown in Fig.4. This configuration assumes ubiquitous information environment. We installed the developed agents based on KCF on WS-{A, B} whose specifications are shown in the figure. User-{A, B} receive services on WS-{A, B}. WS-A is connected with 100Mbps Ethernet. WS-B is connected with 11Mbps wireless LAN. In this experiment, we assume the general environment that is configured by usual spec computers and networks with resource limitations, such as home or outside location. We also assume that videoconferencing is performed with other applications at the same time in such kind of environment. Under this condition, VCS has the limitation concerning the resource usage rate for the user to use other applications. WS-{C, D} add load to a wireless LAN by the data communication based on the traffic pattern that is actually observed in real networks. By this intended load, resource availability limitation becomes not achieved, and then the agents control QoS to improve this undesirable situation. In these experiments, VCS had acquired 16 kinds of knowledge by preliminary videoconference operation based on KCF. Here, we discuss the effectiveness of KCF by observing temporal changes of QoS and resource availability of VCS, comparing the VCS with KCF to the VCS with existing scheme (FMCS [4]).

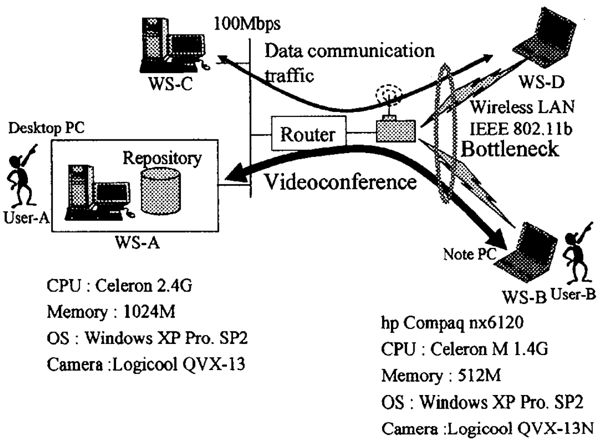


Fig. 4: Experimental environment

The following items are conditions of the experiments.

- The value of user requirement is fps = 20(f/s)
- Network resource availability limitation is available bandwidth rate ≥ 50 %

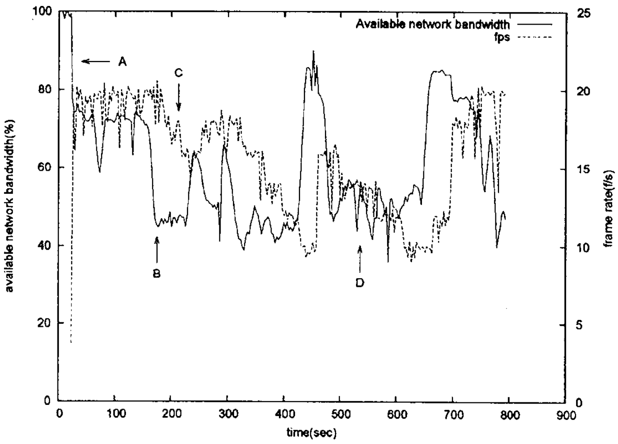
4.3 Experimental results

We added two network load patterns to VCS with KCF and with existing scheme. Results can be found in Fig.5 and Fig.6

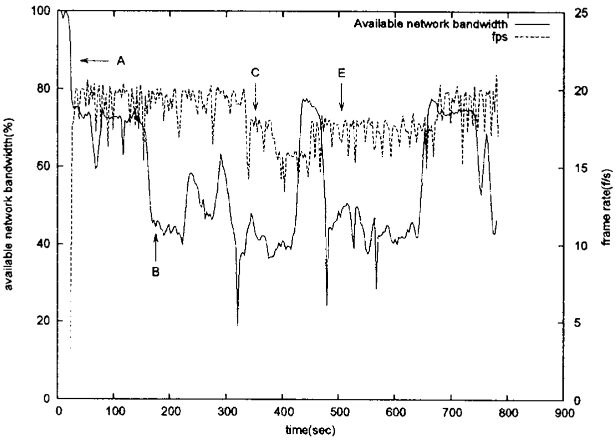
After the experiments started, we initiated videoconference and added the network load based on the load pattern actually observed (point A). As a result, the available network bandwidth was decreased and network resource limitation was not achieved (point B). VCS corresponded by adjustment of the QoS (point C). VCS with existing scheme in Fig.5(a) and Fig.6(a), frequently adjusted QoS, directly according to the change of the network resource (point D). As a result, QoS became unstable. On the other hand, VCS with KCF in Fig.5(b) and Fig.6(b), controlled QoS that did not adjust frequent corresponding to change of network resource situation by using the knowledge that accumulated beforehand (point E). As a result, KCF was able to avoid a frequent QoS fluctuation, and provided service with stable quality.

4.4 Evaluation

From the experimental results of section 4.3, we compare operation of VCS with existing scheme and VCS with KCF. VCS with

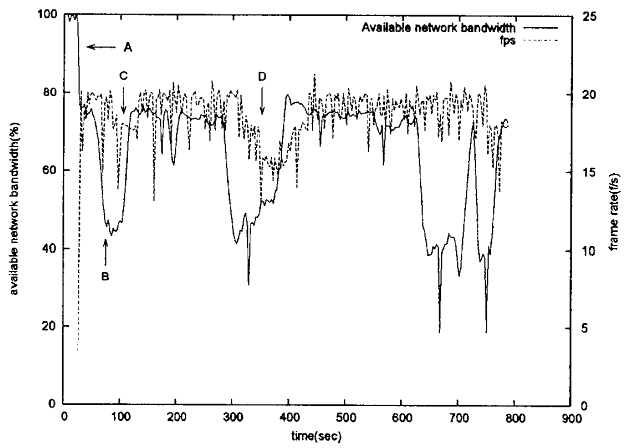


(a) VCS with existing scheme (FMCS [4])

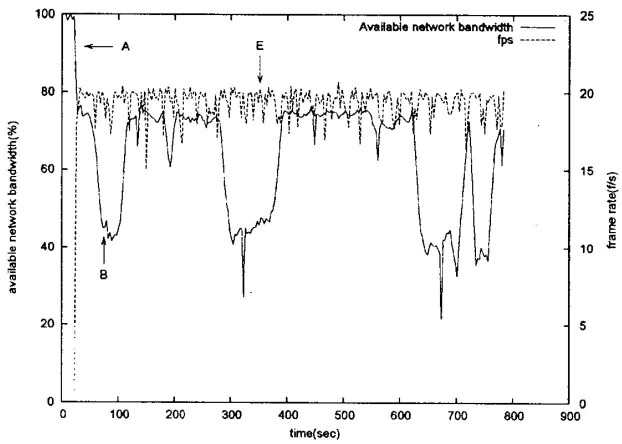


(b) VCS with proposed scheme (KCF)

Fig. 5: VCS against network traffic pattern (1)



(a) VCS with existing scheme (FMCS [4])



(b) VCS with proposed scheme (KCF)

Fig. 6: VCS against network traffic pattern (2)

existing scheme frequently adjusted QoS because VCS was not given the knowledge to correspond resource environment. On the other hand, VCS with KCF provided service with stable QoS by suitable QoS control. This is because VCS was able to effectively control the QoS, because it has knowledge about the environment, i.e., the network resource in that environment is repeatedly changing with a short time interval. This knowledge has been acquired by prior operation based on KCF. From these experiments, we concluded that the adaptability of VCS to ubiquitous information environments has improved by applying the KCF.

5. Conclusion

In this paper, we proposed a Knowledge Circulation Framework (KCF) that enables sharing knowledge on the ubiquitous information environments. This framework supports circulation of knowledge in the network, by a sequence of acquisition, management, and placement of the knowledge to reuse it effectively. We proposed a design of KCF with knowledge-based multiagent system, and then we performed experiments by applying the KCF to VCS. From the results of experiments, we concluded that the adaptability of VCS to ubiquitous information environments has improved by applying our proposed framework. In our future work, we will evaluate the KCF by the experiment in various actual environments.

Acknowledgement

This work has been supported by the Grant-in-Aid for JSPS Fellows No.18-5190 of Japan Society for the Promotion of Science (JSPS).

References

[1] S. Imai, T. Suganuma and N. Shiratori, "Knowledge Circulation Framework for Flexible Multimedia Communication Services," *IEICE Trans. Inf. & Syst.*, Vol. E88-D, No. 9, pp. 2059-2066, Sep. 2005.

[2] T. Turletti and C. Huitema, "Videoconferencing on the Internet," *IEEE/ACM Trans. on Networking*, vol.4, pp.340-351, Mar. 1996.

[3] J. Bolliger, and T. Gross, "A framework-based approach to the development of network-aware applications," *IEEE Trans. Software Eng.*, vol.24, pp. 376-390, May. 1998.

[4] T. Suganuma, S. Imai, T. Kinoshita, K. Sugawara and N. Shiratori, "A Flexible Videoconference System Based on Multiagent Framework," *IEEE Trans. on Systems, Man, Cybernetics, partA*, vol. 33, no.5, pp. 633-641, Sep. 2003.

[5] S. Fujita, H. Hara, K. Sugawara, T. Kinoshita and N. Shiratori, "Agent-based design model of adaptive distributed systems," *Applied Intelligence*, vol.9, pp. 57-70, Jul/Aug, 1998.

[6] R. G. Smith, "The Contract net protocol: High-level communication and control in a distributed Problem solver," *IEEE Trans. Comput.*, vol. 29, no. 12, pp. 1104-1113, 1980.

[7] DASH -Distributed Agent System based on Hybrid Architecture- Webpage,
http://www.agent-town.com/dash/index.html