

マルチエージェントに基づく QoS アウェアな コンテキスト情報管理方式の提案

高橋 秀幸, 東海林 祥一, 山中 一宏, 北形 元, 菅沼 拓夫, 白鳥 則郎

東北大学電気通信研究所/情報科学研究科

{hideyuki, tokairin, yamanaka, minatsu, suganuma, norio}@shiratori.riec.tohoku.ac.jp

あらまし

本稿では、ユビキタス環境におけるハードウェア、ソフトウェア、ネットワークなどの多種多様なサービス構成要素を、高度な自律性と協調性に基づくエージェントとして構成することにより、QoS アウェアなサービスを提供する、コンテキスト情報の効果的な管理手法を提案する。具体的には、コンテキスト情報の過度な供給によって利用可能な計算機・ネットワーク資源が減少し、サービス品質が低下する問題を回避する手法を提案する。

A QoS-aware Context Information Management Scheme for Ubiquitous Information Environment based on Multiagent

Hideyuki Takahashi, Yoshikazu Tokairin, Kazuhiro Yamanaka,
Gen Kitagata, Takuo Suganuma and Norio Shiratori

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

Abstract In this paper we propose an effective context information management scheme for QoS-aware ubiquitous service provisioning. To do this we compose entities of hardware, software, network and user in ubiquitous environment as autonomous and cooperative agents. By employing this scheme, we can avoid degradation of available network bandwidth and computational resource due to supplying excessive context information.

1 はじめに

近年のユビキタスコンピューティングの進展に伴い、コンテキストアウェアサービスへの期待が高まっている [1, 2, 3]. コンテキストアウェアサービスとは、ユビキタス情報環境上に存在するハードウェア、ソフトウェア、ネットワーク、ユーザなどの構成要素 (エンティティ) の状況を示すコンテキスト情報に基づき提供されるサービスである。一般に、ユーザコンテキスト (位置情報や嗜好) やリソース状況などからなるコンテキスト情報 (以下 CI と略) は刻々と変化し、その情報量も膨大である。したがって、それらを全て流通・処理するだけの計算機・ネットワーク資源がユビキタス情報環境において常に利用可能であるとは限らない。このため、特にマルチメディア通信サービスのような、資源を大量に消費するアプリケーションを提供する際、過度な CI の流通・処理の発生により、本来のサービスが利用する資源が減少し、結果としてサービス全体の QoS (Quality of Service) の大幅な低下を招く問題がある。

本研究は、ユビキタス情報環境でのサービス提供において計算機・ネットワーク資源が十分でない場合に、効果的な CI の管理により、適切な QoS を実現することを目的とする。本稿では、ユビキタス情報環境上に存在する各種エンティティを、高度な自律性と協調性に基づく

エージェントとして構成することによる効果的なコンテキスト情報管理方式を提案する。提案方式により、過度な CI の供給によって使用可能な通信帯域や計算処理能力が減少することを回避し、サービス全体の品質や機能を維持する効果が実現できる。

本手法をユビキタスライブストリーミングシステムに適用し、実験を行った。その結果、既存方式では、PHS などの帯域の狭いネットワーク環境下で、常時毎秒 20 回という過度な回数のユーザ位置情報の供給によってフレームレートが 10fps まで低下するのに対し、提案方式では平均 18fps 程度に維持することが可能であることを示した。また、計算機資源に関しては、既存手法では、使用可能な CPU 資源が 6% まで減少し、その影響で動画像のフレームレートが平均 19fps まで低下するのに対し、提案方式では、位置情報の更新頻度を状況に応じて毎秒 1~10 回程度に制御して提供しつつ、使用可能な CPU 資源を 69% まで回復した。その結果、フレームレートを平均 28fps 程度に維持することが可能であることを示した。

2 既存研究と課題

2.1 既存研究

ユビキタス情報環境における、QoS を考慮したコンテキストアウェアなフレームワークに関する研究として、[4, 5] などがある。これらは、ネットワーク資源・計算機

資源などからなる CI を判断基準として、QoS を考慮したサービスの実現を目指している。これらの研究では、CI の取得方法や管理方法は静的であり、CI の情報量が膨大になった際のシステム全体の安定的な振る舞いに関する研究が期待されている。

一方、コンテキスト情報管理方式に関する研究としては、[6, 7] などがある。CHANSE [6] では、種々の CI を管理サーバで集中管理し、センサの故障などにより所望の CI が得られない場合は、センサの代替により継続的な CI の提供を実現する。藤波らは、各種 CI における更新頻度の多様性の抽象化を目的としたフレームワークを提案している [7]。CI の更新の際に、更新頻度の低い CI に対しても最新状態を確認することにより、最新の CI を常に獲得し、CI の更新状態の遅延から生じる誤動作の軽減に大きく貢献している。これらの研究では、CI の選択処理を静的に記述する必要があり、サービス提供中のセンサの追加やアプリケーションの状態を考慮した動的な CI の提供への期待が高まっている。

また、コンテキスト情報の品質 (Quality of Context) に関する研究が行われている [8, 9]。Rocha らは、ヘテロな環境上で CI を管理するためのミドルウェアを提案し、QoC のパラメータとして、CI の新しさ (freshness)、精度 (precision) を定義している [8]。Buchholz らは、[9] で、QoC パラメータとして、CI の精度 (Precision)、正確さ (Probability of Correctness)、信頼性 (Trustworthiness)、粒度 (Resolution)、そして更新頻度 (Up-to-dateness) を定義している、さらに、QoC と QoS の関連性および、QoC が与える QoS への影響について述べているが、実システムにおける具体的な QoC の利用法については今後の課題としている。

2.2 技術的課題

QoS アウェアなユビキタスサービスを実現するためのコンテキスト情報管理において、以下の 2 点が課題となっている。

(P1) サービス提供のための CI の動的選択

ユビキタス情報環境では、サービスを動的に構成して提供する際、同一の目的で使用される CI の提供が可能なエンティティが複数存在することがある。また、各エンティティは異なる特性を持ち、得られる QoC も多様であるため、それらを十分考慮しないと、所望の QoS でのサービス提供が困難となる。例えば、ユーザの位置情報を提供する CI を選択する場合、RFID、GPS、超音波センサなどの測定可能な精度や使用環境 (室内外) などを考慮する必要があり、また、ユーザの移動やサービスに応じて、利用可能な CI を動的に切り替える必要がある。

(P2) システム全体の挙動を意識した効果的な CI の流通

ユビキタス情報環境の特徴として、エンティティの状態は刻々と変化し、計算機・ネットワーク資源が常に十分利用できるとは限らない。そのため、エンティティ間で交換される CI の量も膨大であり、その CI を全て流通させ、処理することが可能であるとは限らない。したがって、連携するエンティティの状態によっては、過度な CI の流通・処理が発生し、エンティティが本来のサービス提供に必要な資源が消費され、QoS の大幅な低下を招く問題がある。そのため、サービスに関わるシステム全体の挙動を意識した CI の流通が必要である。

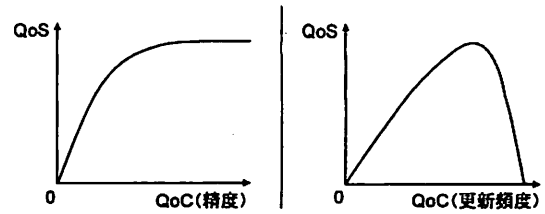


図 1: 各 QoC パラメータと QoS の関係の例

3 エージェント間協調に基づくコンテキスト情報管理方式

3.1 コンテキスト情報管理方式の概要

2.2 節の 2 つの技術的課題は、各エンティティの自律性・協調性の欠如と、サービスを支える柔軟なシステム構成のための基盤の欠如に起因すると考えられる。そこで、サービスに関わる各種エンティティを、高度な自律性と協調性に基づくエージェントとして構成することによる、効果的なコンテキスト情報管理方式を提案する。本方式によって以下の効果が期待される。

- エンティティの動作状況に応じた最新の CI の自律的獲得と効果的な分散管理
- エージェント間協調による CI の効果的な流通
- CI の動的な選択と合成によるサービスのアドホックな構成・提供
- システム構築における拡張性・柔軟性の向上

本方式は以下の 2 ステップからなる。

(S1) 動的なコンテキスト選択

(S2) QoC 調整

ここで、(S1) はサービス提供開始時に実行され、(P1) の解を与える。また (S2) はサービス提供中に実行され、(P2) の解を与える。スペースの関係から (S1) は他稿に譲ることとし、本稿では (S2) に焦点を当てる。

3.2 QoC と QoS の関係

本方式における QoC の基本概念とパラメータ定義は [8] に基づくものとする。ここで、あるサービスの CI としてユーザの位置情報を利用する場合を想定し、位置情報の特定精度と更新頻度を QoC パラメータとする。位置情報の特定精度は、高くなるに従って QoS も向上するが、ある一定の精度からは飽和状態となる。例えば、美術館などの展示ガイドサービスでは、展示物の前にユーザがいることを把握できれば十分であり、ミリ単位の特定精度で位置情報を通知しても QoS は向上しない。一方、更新頻度については、低くなるに従ってユーザの移動をリアルタイムで捉えることが困難となり、逆に、更新頻度が過度になると CI の通知や獲得・処理に追われ、QoS の低下や、システム全体の挙動が不安定になる。

以上をまとめると図 1 のようになる。これは QoC とシステム全体の QoS の関連をグラフ化したものである。すなわち、システム全体の QoS を安定かつ高く維持するためには、QoC のパラメータの特性とシステム内の CI のセマンティクスに基づき、QoC を適正なレベルで制御する、動的な QoC 調整の仕組みが必要である。

3.3 エージェントアーキテクチャ

効果的な CI の管理を実現するため、各種エンティティを、高度な自律性と協調性に基づくエージェントとして構成する。エージェント化とは、エンティティおよびエンティティの CI の管理に関する知識を付加し、エンティ

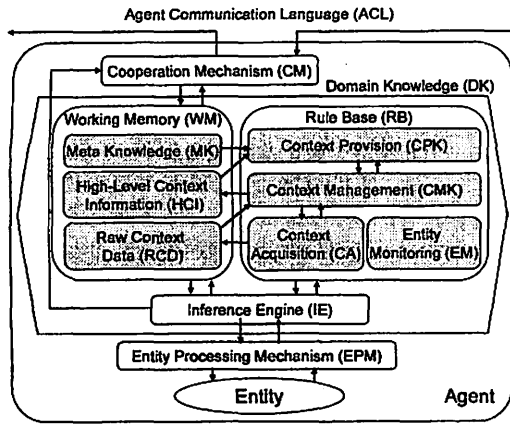


図 2: エージェントアーキテクチャ

ティが自律的かつ協調的に動作する機能を与えることである。

エージェントアーキテクチャを図 2 に示す。CM はエージェント間でメッセージを交換するための機構である。DK は各エンティティに関する様々なドメイン知識を蓄積し、それらを活性化する推論型知識ベースシステムである。EPM は DK とエンティティ間のインタフェースであり、エンティティの直接制御、および状態監視を行う。DK は WM, IE, RB のサブシステムからなるプロダクションシステムである。WM には事実、RB にはルールの集合が保存される。IE はルールと事実から推論を行う機構である。

以上の基本構造に基づき、DK 内の知識モジュールとして CI 管理機能を構成する。CI は、EPM を介して CA により獲得され、RCD として蓄積される。RCD は CMK により解析され、QoC、フォーマットなどのメタ情報が付加される。さらに、高次の CI として HCI に格納される。CPK は、他のエージェントからの要求・情報に基づいて、HCI から得た CI を、適切なタイミング、間隔、フォーマットで他のエージェントへ通知する。EM はエンティティの動作状況の監視・制御によって、効率的なデータ獲得を行う。MK は、エンティティの機能や性能を静的に記述した知識である。これらの知識モジュールが相互に連携して、エンティティの自律性と協調性の向上を実現する。

3.4 エージェント間協調による QoC 調整

システム全体の挙動を安定に保つために、各エンティティの状態、CI の利用形態などに基づいて QoC を動的に変更し、CI の流通を制御する仕組みが必要である。本手法では、特に更新頻度に着目し、状況に応じた QoC の動的調整および CI の集約により QoS アウェアなサービスを実現する機能を提供する。本手法は以下の 4 つのステップから構成される。QoC 調整の動作例を図 3 に示す。

(1) イベント検出: QoC の調整の必要が生じるイベントの検出。例えば、以下の 2 つがある。

(1-1) QoC 変更要求: システムの QoS の向上を目的として、CI の更新頻度の変更を要求するイベント。例えば、位置情報と携帯端末を利用した鬼ごっこなどのアプリケーションの場合、鬼の接近に応じて位置情報の更新頻度を高くする要求など。

(1-2) 資源状況の変化: エンティティの資源情報から、更新頻度を抑える必要がある場合のイベント。例えば、CI

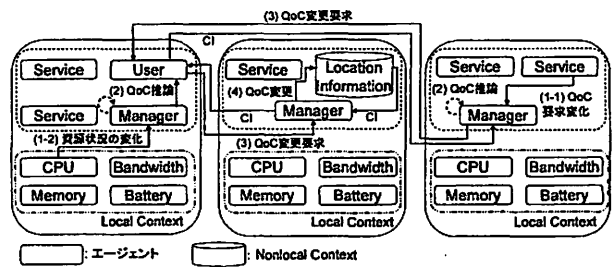


図 3: QoC 調整の動作例

による CPU 使用率、ネットワーク使用率が閾値を超えた旨の警告が発生した場合など。

(2) 適切な QoC の導出: 適切な更新頻度を決定するための推論を実行する。また、(1-2) において、ネットワーク資源の低下などによる CI の遅延が閾値以上に達した場合に CI の集約レベルを決定するための推論も同時に実行する。

(3) QoC 変更要求: 更新頻度の変更要求、あるいは CI の集約要求を CI 通知元のエージェントへ送信する。

(4) QoC 変更: (3) を受信したエージェントが更新頻度、CI 集約レベルを変更する。

以上から CI の QoC と集約レベルを決定する。

4 実装

4.1 ユビキタスライブストリーミングサービスの概要

本手法の有効性を検証するため、ユビキタス情報環境を想定したライブストリーミングシステムを実装した。本システムは、固定したカメラから撮影された動画を、移動するユーザの要求と位置に応じて最寄の表示装置 (PC ディスプレイ、テレビ等) へ配信するサービスを提供する。応用例として、在宅高齢者の見守り支援やペットの監視などを想定している。

4.2 実験システムの設計

本実験に用いたエージェントの構成を図 4 に示す。

User Ag: サービスに関わるユーザの要求・位置情報などの CI を管理し、他のエージェントへ通知する。

UserReq Ag: ユーザ要求を獲得し、*User Ag* へ通知する。

Location Ag: 位置情報の CI を獲得し、提供する機能を持つ。他のエージェントからの要求に応じて、動的に更新頻度を変更し、CI を通知する。

Manager Ag: 各 PC 内のエージェントを管理し、他のエンティティの *Manager Ag* と通信やサービス入力元・出力先切り替えの判断および命令を行う機能を持つ。

CPU Ag: CPU の利用率を CI として、他のエージェントから要求された間隔で監視し通知する。

JMF Send Ag, JMF Recv Ag: JMF [10] を利用し、*JMF Send Ag* が動画の送信を、*JMF Recv Ag* が動画の受信を、*Manager Ag* の要求に応じて行う。提供可能な画質、フレームレート等の調整を行う。

本実験の実験環境を図 5 に示す。エージェントは、ルールの DASH [11] と、開発支援環境 IDEA [12] を使用した。位置情報データを用いるため、毎秒約 20 回の更新頻度で位置情報を取得する超音波センサ (ZPS [13]) を利用した。OS は Windows XP、システムの実装言語には Java と C++ を使用した。通信方式は制御部分に TCP を、メディア転送には UDP を使用した。現在、エージェント間の通信方式 (TCP, UDP) をネットワークの状況に応じて選択す

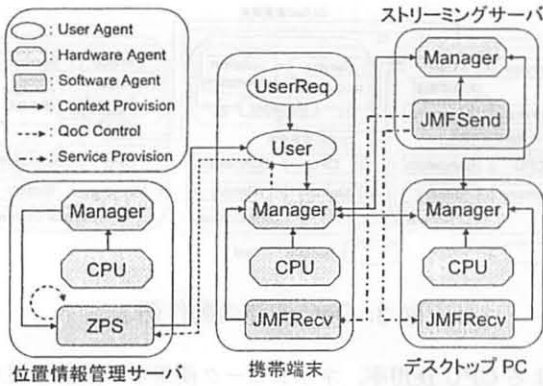


図 4: 実験環境におけるエージェント構成

る機構の開発を行っているが、本実験では実験シナリオによって静的に決定した通信方式で位置情報の送受信を行う。ネットワーク環境は、有線 LAN (Ethernet 100Mbps), 無線 LAN (IEEE802.11b/54Mbps), PHS 回線 (128kbps) が混在する環境を構築し、動画像の送受信には Web カメラと PC ディスプレイを利用した。初期設定では、Web カメラからの動画像をユーザが持つ携帯端末へ出力する。ユーザが移動し、デスクトップ PC に近づいた場合は、デスクトップ PC へサービスが移転する。User Ag は携帯端末上に配置し、ZPS 上の Location Ag から受けたユーザの位置情報を管理する。

5 実験と評価

5.1 実験方法

本実験の目的は、エージェント間協調による QoC 調整の有効性を検証することである。具体的には、本手法により、状況に応じた QoC の調整によって適切な QoS の実現が可能であることを確認する。

実験方法は、ユーザ携帯端末のネットワークとして PHS および無線 LAN を利用し、動画像を受信しながら位置情報 (CI) を受信したときの動画像のフレームレート (fps) の変化を測定する。本実験では、QoC として更新頻度を、QoS として fps をパラメータとする。

実験シナリオを、図 6 に示す。シナリオ 1 はサービス移転を行わない場合、シナリオ 2 はサービス移転を行う場合である。エリア B からエリア D は ZPS によるユーザ位置情報の監視範囲である。シナリオ 1 では、ユーザは PHS で接続した携帯端末を持ち、エリア A からエリア B に移動後、再びエリア A に移動する。シナリオ 2 では、ユーザは無線 LAN に接続した携帯端末を持ち、ユー

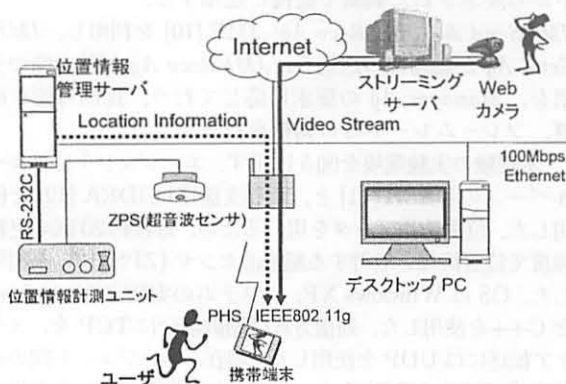


図 5: 実験環境

ザはエリア B からエリア D へと移動する。サービス移転はエリア D のデスクトップ PC に近づいた際に発生する。本実験では、位置情報に応じた QoC 変更要求はデスクトップ PC が行うものとした。各シナリオにおいて、既存手法 (更新頻度の動的な調整を行わない場合) と本手法 (エージェント間協調による更新頻度の動的な調整を行う場合) を比較する。なお、ネットワーク帯域に応じたエージェント間の位置情報の通信方式として、シナリオ 1 では UDP を、シナリオ 2 では TCP を用いた。

本実験ではエージェント間協調による QoC 調整の効果を測定する。そのため、予備実験の経験を基に、更新頻度を簡略的な 3 段階の調整 (10, 4, 1 [回/s]) に設定した。既存手法では位置情報管理サーバは常に毎秒 20 回の更新頻度で取得した位置情報をそのまま提供する。本手法では、User, CPU から CI を受信した各 Manager は以下の QoC 調整ルールに基づき位置情報の更新頻度 u [回/s] の変更要求を行う。c は携帯端末、デスクトップ PC, 位置情報管理サーバの CPU 使用率を表す。

1. if $c \geq 80\%$ then $u = 1$
2. if エリア B でユーザを検出 then $u = 1$
3. if エリア C でユーザを検出 then $u = 4$
4. if エリア D でユーザを検出 then $u = 10$

5.2 実験結果

(1) シナリオ 1 の実験結果

シナリオ 1 の実験結果を図 7 と図 8 に示す。シナリオ 1 では、ユーザはエリア A で動画像を受信後、約 30 秒後にエリア B へ移動し、約 90 秒後に再びエリア A へ移動する。図 7 は既存手法、図 8 は本手法の結果である。動画像の要求フレームレート (ユーザ要求) は、PHS の使用可能なネットワーク帯域を考慮し 20fps とした。図 7, 図 8 の上段には、携帯端末が受信した動画像の fps と位置情報サーバが携帯端末に送信した位置情報の送信トラフィック量を示し、下段には、携帯端末の QoC を示している。

既存手法の場合 (図 7), ユーザがエリア B に移動後、fps が 10fps 程度まで低下した。これは動画像の受信中に、過度の QoC で位置情報を受信し、携帯端末のネットワークが圧迫されたため、fps が低下したと考えられる。そのため、位置情報を受信しないエリア A に再度移動した後、fps が 17fps 程度まで回復している。なお、携帯端末の CPU 使用率は 70% 程度であり、CPU 負荷による fps への影響はない。一方、本手法の場合 (図 8), 既存手法に比べて fps の低下はなく、17fps 程度を維持している。また、ユーザの位置に応じた QoC の調整を行っており、QoC による fps への影響もないことがわかる。

(2) シナリオ 2 の実験結果

シナリオ 2 の実験結果を図 9 と図 10 に示す。ここでは、ユーザはエリア B から約 15 秒ごとにエリア C, D へと

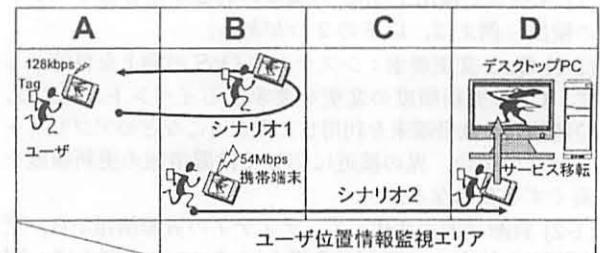


図 6: 実験シナリオ

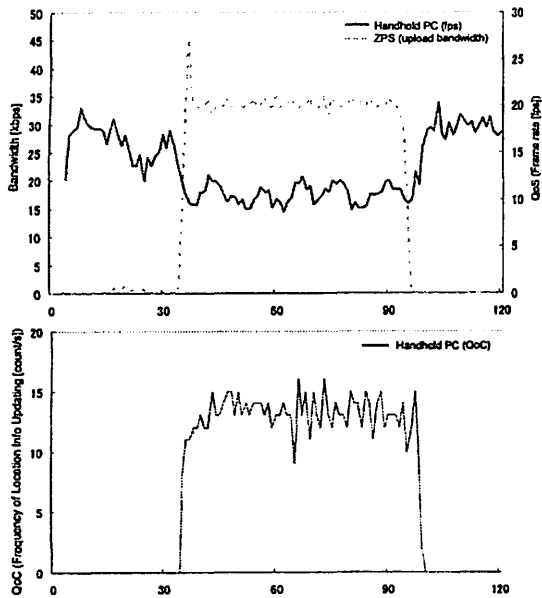


図 7: シナリオ 1 の実験結果 (既存手法)

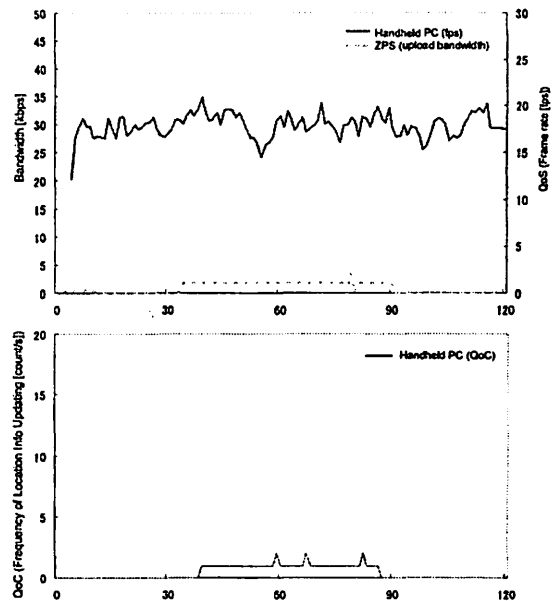


図 8: シナリオ 1 の実験結果 (提案手法)

移動し、デスクトップ PC へサービス移転を行う。ユーザ要求は 30fps である。図 9 は既存手法、図 10 は本手法の結果である。また、上段には携帯端末、中段にはデスクトップ PC の CPU 利用率と fps、下段には携帯端末の QoC を示す。既存手法の場合(図 9)、位置情報を常に毎秒 20 回の更新頻度で処理するため、携帯端末の使用可能な平均 CPU 資源は 6%まで低下した。また、ユーザ要求 (30fps) に対し、平均フレームレートは 19fps であった。一方、位置情報管理サーバの平均 CPU 使用率は 37%と低負荷であるのに対して、携帯端末の QoC が徐々に低下している。これは携帯端末の CPU 負荷が増加し、徐々に位置情報の処理回数が低下したためと考えられる。さらに、エリア D で行われるサービス移転処理も、実験時間内に実行できなかった。一方、本手法の場合(図 10)、ユーザがエリア C に移動するまで CPU 利用率の急激な増加はなく、エリア D ではサービス移転処理が行われた。サービス移転に要した時間は約 2 秒であり、サービス移転後のデスクトップ PC はユーザ要求の 30fps を維持している。QoC は、実験開始から 15 秒ごとに变化している。これはユーザの位置情報に応じたエージェント間協調によって QoC 調整が行われた結果である。しかし、75 秒付近では QoC が低下し、携帯端末の CPU 使用率も大きく低下した。これは、携帯端末の CPU 使用率が 80%を超えたために、Manager が QoC の変更要求を行った効果と考えられる。また、サービス移転が完了し、他のエージェントが QoC の変更要求を行う必要がないため、QoC が 1 [回/s] という低い QoC で安定している。なお、本手法の場合、携帯端末の使用可能な平均 CPU 資源は 69% であり、既存手法に比べ 63% の増加、平均フレームレートは 28fps であり、既存方式と比較して 9fps 高い結果であった。

5.3 評価

実験結果より、2.2 節の (P2) で述べたとおり、過度の QoC によりネットワーク資源や計算機資源が枯渇し、QoS 低下の問題が発生することを示した。さらに、その

際、QoC 自体も低下する場合があることがわかった。このような場合に、本手法のエージェント間協調の効果によって、動的な QoC 調整が行われ、QoS の向上が可能であることを確認した。今後、QoC 調整の段階の細分化や調整パターンの変更を行うアルゴリズムを導入することで、より効果的な QoC 調整が可能であると考えられる。

また、3.3 節で述べたアーキテクチャにより、本提案のエージェントは、エンティティの動作状況に応じた最新の CI の自律的獲得と効果的な分散管理、および CI の動的な選択と合成によるサービスのアドホックな構成・提供の機能を有している。これらについては今後評価を行っていく。さらに、システム構築時において、新しいセンサ、ハードウェア等の導入の際に、該当エージェントを新たに追加するのみで容易に対応が可能であった。これはエージェント型システムの拡張性・柔軟性による効果である。この性質はユビキタスアプリケーションにおいて特に有用であると考えられる。

6 おわりに

ユビキタス環境上に存在する各種エンティティを、高度な自律性と協調性に基づくエージェントとして構成することにより、個々のエージェントが CI を管理しつつ、その流通を効果的に制御する、コンテキスト情報管理方式を提案した。実験結果より、既存手法では、ネットワーク資源が不十分である場合、過度な位置情報の提供のため、映像のフレームレートが 10fps まで低下するのに対して、本手法は 17fps 程度を維持することが可能であった。また、計算機資源に関して、既存手法では使用可能な平均 CPU 資源が 6% まで減少し、その影響からフレームレートは平均 19fps まで低下した。一方、状況に応じて位置情報の更新頻度を調整する本手法の効果によって、使用可能な CPU 資源は 20% まで低下した後、平均して 69%まで回復し、フレームレートは平均 28fps 程度を維持することが可能であることを示した。今後は、過去の履歴や平均値などを考慮した動的な QoC 調整アルゴリズムの評価、および、QoS と QoC のパラメータの拡張などに取り組む予定である。

謝辞 本研究の一部は、科学研究費補助金 (16300011) の

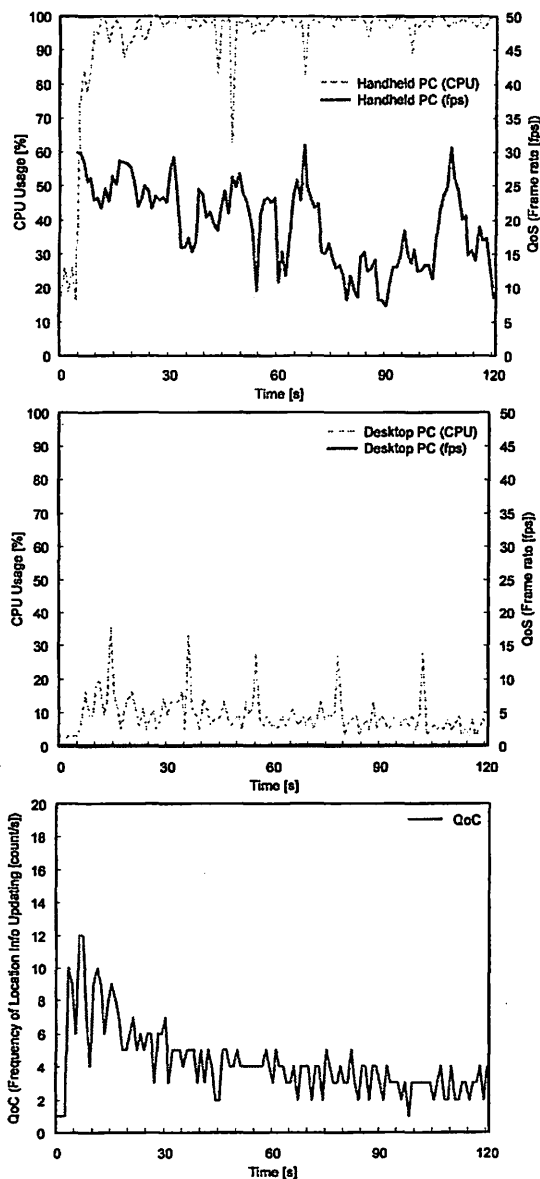


図 9: シナリオ 2 の実験結果 (既存手法)

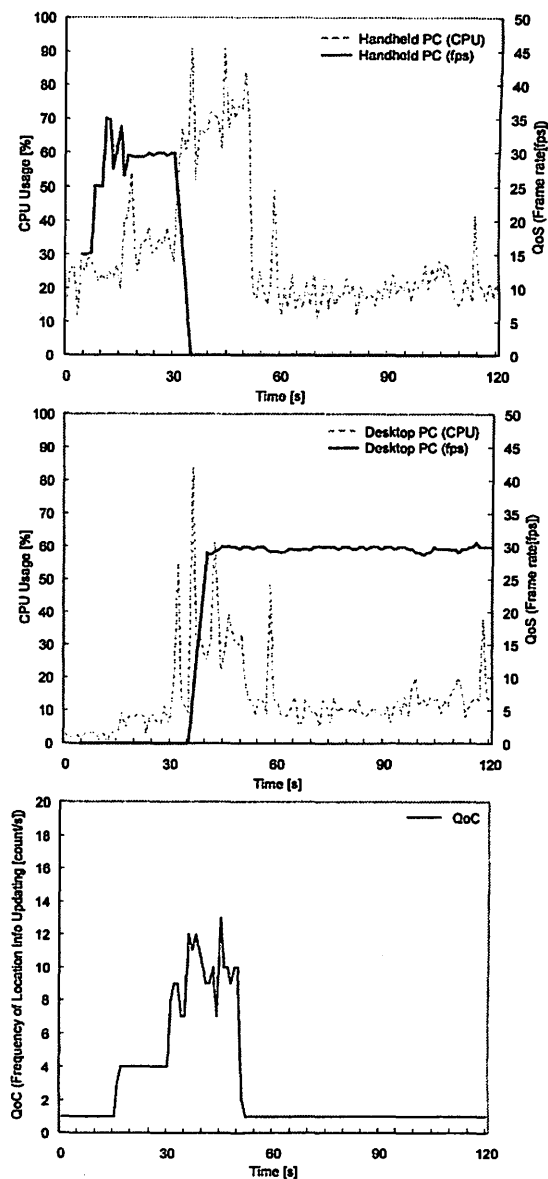


図 10: シナリオ 2 の実験結果 (提案手法)

援助を受けて実施した。

参考文献

- [1] N. Davis, et al., "Using and determining location in a context-sensitive tour guide," *IEEE Comput.*, Vol. 34, No. 8, pp. 35-41, Aug. 2001.
- [2] E. D. Mynatt, et al., "Aware technologies for aging in place: understanding user needs and attitudes," *IEEE Pervasive Comput.*, Vol. 3, No. 2, pp. 36-41, Apr.-Jun. 2004.
- [3] 長 健太, 他, "エージェントフレームワークを用いたコンテキストウェアなテレマティクスサービスの構築," 信学論, D-I, Vol. J88-D-I, No. 9, Sep. 2005.
- [4] J. Liu, et al., "QoS-aware Service Location in Mobile Ad-Hoc Networks," *Proc. of 2004 IEEE International Conference on Mobile Data Management(MDM'04)*, Jan. 2004.
- [5] M. Inoue, et al., "Context-Based Network and Application Management on Seamless Networking Platform," *Wireless Personal Communications*, Vol. 35, No. 1-2, pp. 53-70, Oct. 2005.
- [6] 中村 哲也, 他, "適応型通信サービスにおけるコンテキスト把握機構 CHANSE," 情処学論, Vol. 43, No. 2, pp. 593-604, Feb. 2002.

- [7] 藤波 香織, 他, "コンテキストウェアなアプリケーションフレームワークにおけるメタコンテキスト情報の利用方法の提案とその応用," 日本ソフトウェア科学会コンピュータソフトウェア誌, Vol. 21, No. 1, pp. 46-59, Jan. 2004.
- [8] T. Buchholz, et al., "Quality of context: what it is and why we need it," in *Proc. of the Workshop of the HP OpenView University Association 2003 (HPOVUA2003)*, 2003.
- [9] R. Cout Antunes da Rocha, et al., "Middleware: Context Management in Heterogeneous, Evolving Ubiquitous Environments," *IEEE Distributed Systems Online*, Vol. 7, No. 4, pp. 1-13, ISSN 1541-4922, Apr. 2006.
- [10] Sun Microsystems, Java Media Framework (JMF), <http://java.sun.com/products/java-media/jmf/>.
- [11] S. Fujita, et al., "Agent-based design model of adaptive distributed systems," *The International Journal of Artificial Intelligence, Neural Networks and Complex Problem-Solving Technologies*, Vol. 9, No. 1, pp. 57-70, 1998.
- [12] 打矢 隆弘, 他, "エージェントシステムのインタラクティブ開発環境," 信学論, D-I, Vol. J88-D-I, No. 9, pp. 1344-1355, 2005.
- [13] 古河機械金属株式会社. ZPS (Zone Positioning System), <http://www.furukawakk.jp/products/>.