

## 広帯域大規模分散環境における時間の共有 - イベント同期 -

岡村耕二 山口政博 荒木啓二郎  
奈良先端科学技術大学院大学 情報科学研究科

音声や動画といった連続メディアを扱うアプリケーションソフトウェアが、それらの連続メディアに対して停止や再開の操作を繰り返し施すと、連続メディアの品質が低下してしまう。そのような場合のために、本稿では処理の再開時に関連する複数の連続メディアの同期処理を行なうイベント同期の提案を行ない、イベント同期を実現するための通信プロトコルアーキテクチャを示す。また、遠隔の対人コミュニケーションにイベント同期を適用した例も示す。

## SyncWare on Global Distributed Broadband Environment - Event Synchronization -

Koji OKAMURA, Masahiro YAMAGUCHI and Keijiro ARAKI  
Graduate School of Information Science  
Nara Institute of Science and Technology

The quality of continuous media, such as sound and video, decreases when an application software which uses these continuous media handles them. In this paper, we propose event synchronization which offers synchronization among related continuous media when the application restarts and show the architecture of communication protocols which supports the event synchronization. Then we adapt event synchronization to remote human communication as a example.

## 1 はじめに

近年の計算機の処理能力の向上やネットワークの高速化および普及によって、遠隔にある音声や動画をネットワーク経由で取り寄せることが可能になってきた。音声や動画のネットワーク経由での伝送は、送信側の計算機で符号化、ネットワークを用いた通信、受信側の計算機で復号化という手順で行なわれる [1]。このように計算機やネットワーク上で処理することのできる符号化された音声や動画は連続メディアと呼ばれ、その連続メディア処理には時間的な制約があるという特徴がある。

しかしながら、連続メディアは応用ソフトウェアによって様々な形態で利用されており、必ずしも常に同じ品質の時間的制約を保証しながら処理される必要はない。例えば、音声や動画を実時間的に伝送する必要のある遠隔会議では、発言者の意図を伝達している時と、伝達していない時では保証すべき時間的制約の品質は変化する。また、例えば、連続メディアを利用した情報発信／情報獲得などの応用ソフトウェアでは、利用者が処理を途中で停止することもあるので、連続メディア処理の停止、再開が繰り返し行なわれることになる。このように応用ソフトウェアによって連続メディア処理に対して操作が施された場合、連続メディア処理の時間的な制約が失われてしまうため、再生が再開される時に関連する複数の連続メディア間の再同期処理を行なう必要がある。本稿はその同期処理を行なうイベント同期の提案を行なう。

本稿は 2 章で応用ソフトウェアが連続メディアに操作を施す場合にイベント同期が必要であることを述べ、3 章でイベント同期の説明を行ない、その実現のための通信プロトコルアーキテクチャを示す。4 章では遠隔地の対人コミュニケーションにイベント同期を適用した例を説明し、5 章ではまとめを述べる。

## 2 連続メディアに施される操作

2 章ではまず、連続メディアを用いる典型的な応用の種類をあげ、それぞれの応用が連続メディアに施す操作について説明する。そして、それらの連続メディアに対する操作を行なう時に同期処理が必要であることを述べる。

### 2.1 連続メディアを用いる応用の種類

分散型連続メディア応用は、その連続メディアの処理方法によって、大きく二種類に分けることができる。一つは送信側において符号化された連続メディアを途中で

蓄積せずに、実時間的に送信して、受信側で復号化を行なう処理である。もう一つは送信側において符号化済みの蓄積された連続メディアを予め準備しておき、受信側の要求によって送信し、受信側で復号化を行なう処理である。本稿では前者を非蓄積型連続メディア処理、後者を蓄積型連続メディア処理と呼ぶ。

これら二種類の連続メディア処理の本質的な相違点は、論理時間 [2] の進行方法である。非蓄積型連続メディア処理における論理時間は停止しないが、蓄積型連続メディアの論理時間は必ずしも常に進行しているとは限らない。

図 1 は非蓄積型連続メディア処理の論理時間の進行を表し、図 2 は蓄積型連続メディアの論理時間の進行を表す。図 1、図 2 ともに、左側が送信側、右側が受信側を表している。それぞれの図において横軸は実時間の進みを表し、縦軸は論理時間の進みを表している。

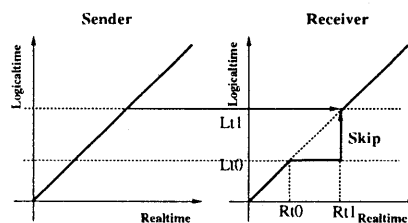


図 1: 非蓄積型連続メディアの論理時間の進行

図 1 の非蓄積型連続メディア処理では、実時間時刻  $Rt1$  において、受信側の再生処理に遅延が生じた場合、論理時間は停止することがないので、受信側の論理時間は  $Lt0$  から  $Lt1$  にスキップする。スキップを行なう時、連続メディア処理は一旦停止し、論理時間をスキップした後、再開する。

一方、図 2 の蓄積型連続メディア処理では、先ほどの非蓄積型連続メディア処理と同様に受信側で再生時間に遅延が生じた場合、送信側の論理時間も停止するので、論理時間時刻は  $Lt0$  のままである。実時間時刻  $Rt1$  では論理時間時刻  $Lt0$  から再生処理が再開する。

以上のように、非蓄積型、蓄積型いずれの連続メディアの処理方法においても、共通的に連続メディア処理の停止や再開という操作が行なわれることがわかる。例えば、図 1 および図 2 において、連続メディア処理の再開の操作は実時間時刻  $Rt1$  に、また、処理の停止の操作は、

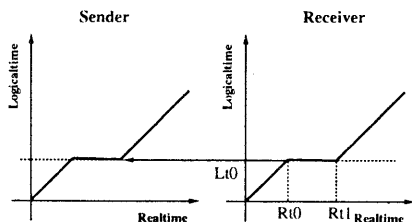


図 2: 蓄積型連続メディアの論理時間の進行

それぞれ連続メディア処理が応用ソフトウェアによって停止させられる実時間時刻  $Rt1$  および  $Rt0$  において行なわれている。

## 2.2 連続メディアに対する操作

以上のように応用ソフトウェアからの操作によって連続メディア処理の品質である、再生のなめらかさや、関連する複数の連続メディア間の同期が失われてしまう。ところが従来の連続メディアを保証する方式では、外部からの影響を排除することによって品質を保証しているため、このような避けることのできない外部からの影響による品質の低下に対処することができない。

本稿では、連続メディア処理に操作が施された時にイベント同期を導入することによって、この問題を解決する。イベント同期では、連続メディア処理の停止の時は特別な処理を行っていないが、連続メディア処理の開始の時は以下のことに注意して処理を行なっている。

- ・ 連続メディア処理の開始時に、もし、関連する複数の連続メディアを同期させる必要がある場合、処理の開始点を合わせる必要がある。
- ・ 連続メディア処理の開始時に、バッファにまだ再生されていない連続メディアが残っていた場合、それらの処理を行なう必要がある。ただし、その連続メディアを再生するか、破棄するかは応用ソフトウェアによって決められる。

イベント同期の詳細については次の 3 章で説明する。

## 3 イベント同期

イベント同期は連続メディア処理の再生および停止によって引き起こされる、関連する複数の連続メディア間

の同期のずれを修正することをその目的としている。そのため、まず、連続メディア間の同期のずれを定量的に解析し、それに基づいて、イベント同期の説明を行ない、その実装のための通信プロトコルアーキテクチャを示す。

### 3.1 連続メディア処理間の同期のずれ

図 3 に、典型的な連続メディア同期機構を示す。同期機構に必要な全ての機能は再生側にのみ存在する。

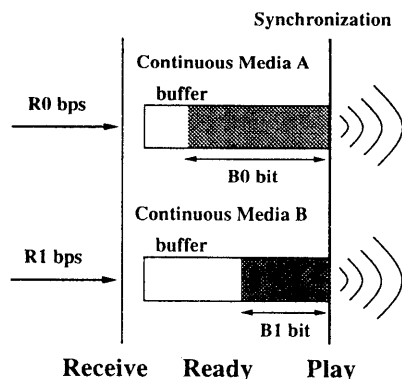


図 3: 連続メディア同期機構

図 3 は、2 つの連続メディア処理 A, B において、連続メディアの受信、バッファリングおよび同期しながら再生が行なわれている様子を表している。連続メディア A, B はそれぞれ、平均レート  $R0(bps)$ 、 $R1(bps)$  で伝送され、それぞれのバッファには連続メディアデータが  $B0(bit)$ 、 $B1(bit)$  溜っているとす。この時の、連続メディア A, B 間の同期のずれの時間は、以下の (1) 式で表すことができる。

$$\left| \frac{B0}{R0} - \frac{B1}{R1} \right| \text{ (sec)} \quad \dots (1)$$

それぞれの連続メディアの伝送レートは常に一定であると仮定すると、複数の関連する連続メディア間の同期のずれは (1) 式から、バッファに溜っているデータ量から算出することが可能であることがわかる。バッファにデータが溜る原因としては、なんらかの理由により連続メディアの再生が再生予定時刻に行なわなかった場合を想定することができる。

同期機構はこの (1) で表されている同期のずれの時間が QoS で指定した時間を越えた時、溜っているデータ

の破棄や再生タイミングを調整することによって同期の品質を保証する。しかし、この時、いかにして同期機構に(1)式が同期のQoSを越えたことを認識させるかという問題がある。

同期の品質を保証する最も簡単な方式として、一定時間間隔で(1)式が常に指定された同期のQoSを満たしているかどうかチェックする方法が考えられるが、この方式のポーリングのオーバーヘッドは無視できない。そこで、本稿では、2章で述べた連続メディアに対する操作を施す時に連続メディアの連続性が特に著しく損なわれることに注目して、連続メディアの操作時に関する連続メディア間の同期の品質のチェックを行なう方式を提案する。連続メディア処理に対する操作をイベントとしてとらえ、本稿で提案する同期方式をイベント同期と呼ぶ。

### 3.2 イベント同期処理

イベント同期処理は、連続メディア処理に対して施される操作のうち、連続メディア処理を開始する時にだけ行なわれる。連続メディアを開始する時の状態が図3であった時、図3において、連続メディアAは連続メディアBに対して遅延していたとする。この時、AのBに対する遅延時間は以下の(2)式で表すことができる。

$$\frac{B_0}{R_0} - \frac{B_1}{R_1} \text{ (sec)} \quad \dots (2)$$

連続メディアAの平均レートは $R_0(\text{bps})$ なので、バッファには(3)式で表される連続メディアBに対して遅延しているデータが存在する。

$$R_0 \times \left( \frac{B_0}{R_0} - \frac{B_1}{R_1} \right) \text{ (bit)} \quad \dots (3)$$

連続メディア処理が開始される時、イベント同期処理によってこの同期のずれが調整される。この時、連続メディア処理を扱う応用の仕様や連続メディアAおよびB間の相対的な優先度によって以下に示されるいずれかのモードで処理が行なわれる。

#### 全バッファをクリアする

連続メディア処理を停止した後、次に再開するまでに受信した連続メディアを再生する必要がない場合は、全バッファをクリアし、再生を再開してから受信した連続メディアを再生できるようにする。図3において連続メディアA、Bそれぞれのバッファに溜っている $B_0(\text{bit})$ 、 $B_1(\text{bit})$ は破棄される。この処理は、再生の開始時に関する複数の連続メディア間での同期のずれの有無に関係なく行なわれる。

#### 遅延している連続メディアに合わせる

遅延している連続メディアAを破棄することができない場合、連続メディアAの再生が開始から、(2)式で示される時間後に、連続メディアBの再生を開始する。

#### 先行している連続メディアに合わせる

先行している連続メディアBの再生をすぐに開始したい場合、遅延している連続メディアAのバッファから(3)で示される量のデータを破棄して、再生を開始する。

### 3.3 連続メディア同期機構のための通信プロトコル

3.3章では、イベント同期を利用した連続メディア同期機構を支援するための通信プロトコルについて説明する。通信プロトコルのアーキテクチャは図4に示されるように同期レイヤ、連続メディア転送レイヤおよびデータ転送レイヤという3つのレイヤから構成される。

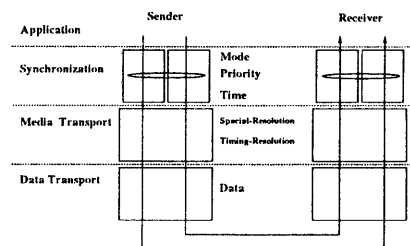


図4: アーキテクチャ

以下にそれぞれのレイヤの機能を説明する。

#### 同期レイヤ

同期レイヤではイベント同期を行なうために必要な情報を送信する。それには、各連続メディアの再生予定時刻、関連する複数メディア間の優先度、およびイベント同期処理のモード情報が含まれる。

#### 連続メディア転送レイヤ

連続メディア転送レイヤは、受信先でメディアが連続メディアとして再生できるように必要な情報を送信する。それには、単位フレームあたりの情報量を

示す空間的解像度と、単位時間あたりのフレーム数を示す時間の解像度が含まれる。

#### データ転送レイヤ

データ・トランスポートレイヤは、連続メディアデータを指定したアドレスに転送する。例えば UDP プロトコルをそのまま利用して構わない。

## 4 対人コミュニケーションへの適用

本稿で提案するイベント同期を、連続メディアを用いた遠隔対人コミュニケーションに適用した例を説明する。

### 4.1 対人コミュニケーション

対人コミュニケーションでは、相手に自分の意図を正確に伝えることが重要である。遠隔地の人間同士が互いに自分の意図を相手に正確に伝えるために、連続メディアを有効に利用することができる。また音声や動画といった複数の連続メディアを同時に利用することによって、さらに意図の伝達に正確さを増すことができる。

図5に、人間Aと人間Bの間で、連続メディアを用いて意図の交換を行なっている様子を示す。

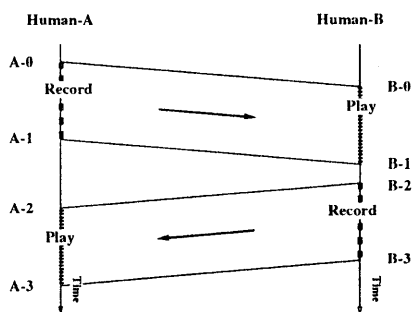


図5: 対人コミュニケーション

図5では、時間は図の上から下に進み、矢印が意図の伝達を表している。各時刻におけるイベントを以下に説明する。

A-0 参加者Aが意図Aの伝達を開始する。

B-0 参加者Bは受け取った意図Aの再生を開始する。

A-1 参加者Aの意図Aの伝達が終了する。

B-1 意図Aの再生が終了する。

B-2 参加者Bは参加者Aの意図Aに反応して、意図Bの伝達を開始する。

A-2 参加者Aは受け取った意図Bの再生を開始する。

B-3 参加者Bの意図Bの伝達が終了する。

A-3 意図Bの再生が終了する。

図5における連続メディア処理を詳しく見てみることにする。まず、この応用では非審積型連続メディアを用いており、常に相手に自分の画像や音声を送っている。ただし、伝送路や端末が受ける予測が困難な負荷の影響によって、必ずしも、常に高品質で再生されていないかったり、音声と動画の同期のずれも発生する可能性があるとする。

ところが、人間Aの時刻A-2～A-3や、人間Bの時刻B-0～B-1の時は、相手の意図の再生を行なっているため、高品質にかつ、音声と動画も同期させる必要がある。そのため、意図が含まれた連続メディアの再生が開始される人間A側では、時刻A-2において、また人間Bにおいては時刻B-0において、それぞれイベント同期を行なう。もし、遅延によってバッファに未再生の連続メディアが残留していた場合は、全てクリアするモードを選ぶのが適切である。

これによって、対人コミュニケーションにおいて重要な意図の伝達がイベント同期によって、正確に行なうことが可能になる。

### 4.2 グループ同期への適用

我々は、遅延時間を無視することのできないような大規模なネットワークで、対人コミュニケーションを円滑に行なう研究をしている[3]。この研究で用いられているグループ同期は、ノード間の遅延時間の差異によってノードごとに再生される内容が異なることを防ぐために、全ノードで受信した連続メディアの再生を一定時間保留させて再生するアルゴリズムを提案している。

グループ同期を図6に示す。グループ同期では、受信した連続メディアの再生は全ノード間の遅延の最大遅延時間 $d$ の間、保留し、絶対時間で先に送信された連続メディアを再生する必要がないことが確認してから再生を始める。人間Aは時刻 $T-0$ で送信を始め、人間Bは時刻 $T-1$ で送信を始める。人間Bは、送信を開始して最大遅延時間分保留している間に人間Aが自分よりも先に

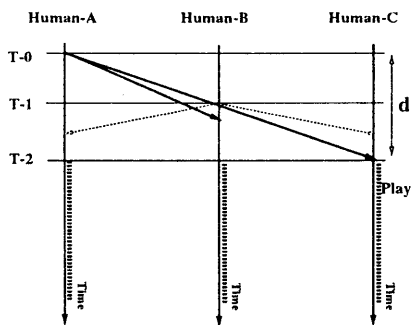


図 6: グループ同期

発言していたことを知り、送信をキャンセルする。そして、全ノードにおいて再生は時刻  $T-2$  から開始される。

このようにグループ同期では発言者からの意図を含んだ連続メディア処理は一定時間保留された後、再生が開始されるという操作が施されているため、連続メディアの開始においては本稿で提案しているイベント同期を有効に利用することができる。

### 4.3 グループウェアへの適用

グループウェアにおいても連続メディアは有効に利用されている。石井はグループウェアにおける対人コミュニケーションを、アウェアネス、コミュニケーション、コラボレーションという3つの階層に分類している[4]。アウェアネスとは連続メディアに制御を加えることなしに送受信することによってお互いの状態を認識可能にする程度のレベルであり、コミュニケーションは連続メディアに適切な制御を加えることによってお互いに意図の交換が可能にするレベルである。本稿で提案するイベント同期は、通常はアウェアネスのレベルを提供している連続メディアを利用した対人コミュニケーションソフトウェアの機能を、必要に応じて動的にコミュニケーションのレベルを提供する機能まで引き上げることができる。

## 5 おわりに

本稿では応用ソフトウェアが関連する複数の連続メディアを操作する時に必要となるイベント同期の提案を行なった。今後はネットワークバンド幅が数百 Mbps、各ノード間の遅延時間が数百 msec 程度の広帯域大規模

分散環境上で遠隔会議円滑に行なうためにグループ同期とイベント同期の機能を持った会議システムを試作し、評価する予定である。

## 参考文献

- [1] 安田, “マルチメディア符号化の国際標準”, 丸善株式会社, 1991.
- [2] D. P. Anderson and G. Homsy, “A Continuous Media I/O Server and Its Synchronization Mechanism”, *IEEE Computer*, vol.24, no.10, pp.51-57, Oct. 1991.
- [3] 山口、岡村、荒木, “広帯域大規模分散環境における時間の共有 - グループ同期 -”, 情報処理学会 マルチメディア通信と分散処理研究会, 1995.
- [4] 石井, “CSCW とグループウェア”, オーム社, 1994.