マルチメディア通信と分散処理　94－17
分散システム/インターネット運用技術　14－17
（1999．7．16）

時間拡張LOTOSの処理系を用いたSMIL記述の実行とQoS制御

寺島芳樹[1]　安本慶一[2]　東野輝夫[1]　安倍広多[3]　松浦敏雄[3]　谷口健一[1]

[1] 大阪大学基礎工学部情報科学科　[2] 滋賀大学経済学部情報管理学科　[3] 大阪市立大学学術情報総合センター

{terasima,higashino,taniguchi}@ics.es.osaka-u.ac.jp

yasumoto@biwako.shiga-u.ac.jp, {k-abe,matsuura}@media.osaka-cu.ac.jp

あらまし　本稿では，マルチメディア記述言語SMILのあるサブクラスにQoS制御機能を追加した「拡張SMIL」とその実行方式を提案する．拡張SMILでは，複数マルチメディアオブジェクト(動画，音声など)の再生開始・終了時刻の指定とそれらの再生順序(並行，逐次など)などに加え，優先度やシステムの負荷変動に応じた動的QoS制御，並行オブジェクトにおける再生のずれの補正が指定可能である．拡張SMIL記述は，時間拡張LOTOS記述(形式記述言語E-LOTOSのあるサブクラス)に変換され，我々が開発した時間拡張LOTOSコンパイラを用いて実現される．提案手法により，システム資源の動的変動に対し各ユーザの嗜好に合ったQoS制御を行うシナリオの記述・実行が行えること，また，今回提案したようなQoS機能をSMILへ拡張する際に実行系の構築が簡便であること，などを確認した．

Implementation of SMIL with QoS control using Real-time LOTOS

Yoshiki Terashima[1], Keiichi Yasumoto[2], Teruo Higashino[1],
Kota Abe[3], Toshio Matsuura[3] and Kenichi Taniguchi[1]

[1] Dept. of Information and Computer Sciences, Osaka University

[2] Dept. of Information Processing and Management, Shiga University

[3] Media Center, Osaka City University

Abstract In this paper, we add some QoS control mechanisms to a subclass of SMIL (called SMIL-QoS), and propose its implementation technique. In SMIL-QoS, in addition to standard facilities of SMIL (e.g., parallel/sequential play-back of multi-media objects with specifying their starting and ending time), (1) dynamic QoS control of each multi-media object, and (2) explicit inter-media synchronization among objects, can be specified. In our implementation technique, we use a subclass of E-LOTOS (called real-time LOTOS) as an intermediate language since it can nicely treat timing aspect and parallel processes with synchronization. Thus, we convert a given SMIL-QoS description to the corresponding real-time LOTOS specification, and implement it with our compiler. We confirmed that the proposed SMIL-QoS and its implementation technique enable user-oriented multi-media presentation scenarios which match user's own preferences and can tolerate dynamic variation of system load. We also confirmed that our implementation technique is applicable to processor prototyping in SMIL extension.

1　はじめに

近年，複数マルチメディアオブジェクト(動画，静止画，音声，テキスト等)を用いた効果的なプレゼンテーション作成のため，それらの再生順序・時刻をシナリオとして記述可能な，シナリオ記述言語およびその処理系が注目されている[1, 4]．シナリオ記述言語の本命として規格化されたSMIL 1.0 [4]では，個々のマルチメディアオブジェクトの表示位置，再生範囲，再生開始・終了時刻の指定や，複数オブジェクトの並行・連続再生などの強力な制御機構，さらには，表示系/ネットワークの性能，機能の制限に応じた代替メディアの指定などが，簡潔に記述できるという特徴を持つ．

SMIL1.0では，複数マルチメディアオブジェクトを並行再生させた時のオブジェクト間の同期の精度や，計算機/ネットワーク資源不足時の代替メディアへの切替えの条件などは，全て表示系の実装依存とみなし，プレゼンテーション記述に陽に指定することはできない．しかし，計算機/ネットワーク資源が変動する環境では，各ユーザサイトにおいて，(各ユーザにとって)相対的に重要でないオブジェクトの再生品質を落してでも重要なオブジェクトの再生品質を保ちたいといった要求や，あるビデオを字幕あるいは音声吹替えで見たい場合に，別々に送られてくるデータをメディアの特性に合わせた精度(あるいは各ユーザが許容可能な精度)で同期させたい，などの要求が存在する．従って，ユーザやアプリケーションごとにサービスの品質に対する要求が異なる場合に対応するには，シナリオ記述言語にQoS制御を指定する機能が必要である．

SMILでは，各オブジェクトへの再生開始・終了時刻の指定や，複数オブジェクトの並列，逐次再生や同期などが指定可能である．これらに加え，動的なQoS制御やメディア間同期を実現する際には，時間を扱える機構，また並行動作とその間の同期

を効率良く扱える機構が必要となる．形式記述言語E-LOTOS[10]は，システムの各動作（イベント）／動作間に実行可能な時刻の範囲を指定する機能と，複数並行プロセス間の選択，同期，割込などの強力な構文を有し，SMILおよびその拡張言語を実現するのに必要十分な機能を持つと思われる．

本稿では，SMILのあるサブクラスを対象に，QoS制御およびメディア間同期を陽に指定するための構文を追加した「拡張SMIL」と，拡張SMIL記述をE-LOTOSのサブクラスである時間拡張LOTOSのコンパイラ[2, 3]を用いて実行する方式を提案する．

拡張SMILでは，(1) 並行再生される各オブジェクトに対する優先度の指定，(2) システムの現在の負荷および優先度に応じた，オブジェクトの再生品質の変更，および代替メディアへの切替え，(3) 複数オブジェクトの並行実行における同期時間間隔の指定，などが可能になる．

拡張SMIL記述は，対応する時間拡張LOTOS仕様に変換することにより実現する．変換は，与えられた拡張SMIL記述におけるマルチメディアオブジェクトの再生動作を，各オブジェクトを指定の品質で再生するプロセス[2, 3]の呼び出しに置き換え，複数オブジェクトの並列，逐次再生を，対応するプロセスの並列実行，逐次実行に置き換えることで実現する．また，拡張SMIL記述におけるQoS制御は，それを満たすためにオブジェクト再生プロセス間で成り立つべき制約を加えることで実現する．

作成した変換系および時間拡張LOTOSコンパイラを用いて，幾つかの拡張SMIL記述から実行可能コードを生成し実行した結果，提案するQoS制御機能により，システム資源の動的変動に対し，各ユーザの嗜好に合った形でQoS制御を行えることを確認した．処理系構築の効率から，時間拡張LOTOSに変換し実行する本方式は，SMILへのQoS制御機能などの拡張に有用であると思われる．

2 SMILとそのQoS拡張

本節では，マルチメディアシナリオ記述言語SMILの概要と，本稿で拡張するQoS制御機能について述べる (SMILの詳細は文献[4]参照)．

2.1 SMIL

SMILドキュメントはXMLエレメント[8]により記述する．エレメント (<element> ... </element>) により機能を指定し，その性質などはエレメント属性 (<element attr_name = attr_value>...</element>) に記述する．

SMILドキュメントでは，再生される連続メディア (動画，音声) や静止メディア (静止画，テキスト) のメディアデータ (以後**オブジェクト**と呼ぶ) について表示されるエリアの幅，高さ，座標などのレイアウトの情報と，再生されるオブジェクトの種類やその再生タイミング (開始時刻，終了時刻，再生期間など) で表されるオブジェクトの挙動とが分けて記述される．

オブジェクト間の制御機構としてpar，seqエレメントがあり，オブジェクトを並行，逐次に再生することができる．これらを入れ子状に使用することで，複雑な順序関係も記述することができる．

この他に，代替のコンテンツを複数指定しておくために使われるswitchエレメントと，どのコンテンツを再生するかを決定するために主に使われるエレメント属性testなどがある．また，ハイパーリンクのためのエレメントが定義されている．

2.2 SMILへのQoS制御文の拡張

本稿では，SMILのサブセットとして次のエレメントと属性を対象とする．

エレメント	エレメント属性
region	id,left,top,width,heigh,title
video,audio,img,text	id,begin,end,dur,clip-begin, clip-end,src,region,fill,repeat
par	id,begin,end,dur,endsync,repeat
seq	id,begin,end,dur,repeat

ここでは，オブジェクトの場所を示すsrcはオブジェクトのファイル名で指定する (ハイパーリンクは対象としない)．オブジェクトの再生開始，終了時刻や再生期間，再生範囲を決めるbegin, dur, end, clip-begin, clip-endの時刻はクロック値として秒で指定する．regionによる表示位置の指定や，再生終了の挙動を指定するfill，繰り返しの回数を指定するrepeat，並列再生の終了タイミングを決めるendsyncなどの指定方法は標準SMILと同じである．

標準SMILには，この他にルートウィンドウの性質を決めるroot-layout，表示エリアとオブジェクトの大きさが合わない時の表示方法を決めるfit，何のメディアか分からないオブジェクトの指定に使用するref，特定メディアが表示できない場合の代替テキストを指定するaltなどがあるが，これらは表示系の問題であるので本稿では扱わない．

本研究では，上記のSMILのサブセットに対し，以下のような3つのQoS制御機能を拡張した．

(1) 複数オブジェクト間の優先度と品質許容範囲の指定　全体のバランスを考慮したQoS制御を記述するため，並行再生される各オブジェクトの動作に対し優先度を指定する構文，および，各オブジェクトに対して許容品質範囲 (どの程度まで品質を下げても良いか) を指定する以下の構文を追加する．

構文	意味
priority = "p"	オブジェクトの優先度 p

pは1〜k (kは任意の整数) の値をとり，1に近

いほどより優先度が高い (重要である) ことを示す.

構文	意味
maxtemporalcrop = "n%"	最大 n%まで間引く

ここで, n はオブジェクトの時間解像度 (動画の場合, フレームレート) を, この値まで間引くことができることを表す.

これらを用いた記述例を以下に与える.

```
<par>
  <video src="video1.mpg"
    maxtemporalcrop="50%" priority="1" />
  <video src="video2.mpg"
    maxtemporalcrop="66%" priority="50" />
  <video src="video3.mpg"
    maxtemporalcrop="66%" priority="100" />
</par>
```

上記において動画 video1.mpg, video2.mpg, video3.mpg が 30fps で再生されることが要求されているとすると, 優先度が高く設定された video1.mpg が 30fps で再生できなくなった時, 優先度が低く設定されている video3.mpg のフレームレートを段階的に最高 66% まで間引く, すなわち最高 10fps まで落すことで, video1.mpg の 30fps での再生動作を保つようにする. video3.mpg がそれ以上落すことが許されていないならば, 次に優先度の低い video2.mpg が最高 10fps まで落される.

(2) 代替メディアへの切替え条件の指定 代替メディアへの動的な切り替えを可能にするため, 本研究では, メディアの再生動作について動的な変化に基づいた代替メディアへの切替え条件及びその代替メディアの指定を行うための次の構文を追加する.

構文	意味
<switch> ··· </switch> system-load = "選択条件"	コンテンツに代替メディア記述条件を満たしたコンテンツが選択

標準 SMIL の switch エレメントではシステムの既知の性能等を判断する test 属性により静的に複数コンテンツから一つを選択するが, ここでは system-load の条件により動的に選択する代替メディアをコンテンツとして記述する.

その条件として, ここでは動的に変化する情報をオブジェクトの再生動作にかかる負荷として考え, 低負荷 (0) から高負荷 (1) に切り替わったことを system-load="0->1" などで表す.

これを用いた記述例を以下に与える.

```
<switch>
  <video src="video.mpg"
    system-load="0 or 1->0" />
  <video src="video_smaller.mpg"
    system-load="0->1 or 2->1" />
  <text  src="text.txt"
    system-load="1->2" />
</switch>
```

上記では, 動画 video.mpg が最初に再生されるオブジェクトで, その代替メディアとして video.mpg と同じ内容だが画像サイズが小さい動画 video_smaller.mpg, 動画の概要を示すようなテキスト text.txt が指定されている.

video.mpg の再生途中で負荷が高くなった場合, video_smaller.mpg に切り替えられるが, 内容は連続して再生され続ける. さらに負荷が高くなった時には, 対応する場面を表すテキストが表示する. 負荷が軽減された時には, 動画の再生に戻る.

(3) メディア間同期の精度指定 一定の精度で同期 (再生のずれの補正) を必要とする複数オブジェクトの再生ため, 本研究では, par エレメントの属性として, 並行に再生されるオブジェクト間で同期させる時間間隔を指定するための次の構文を追加する.

構文	意味
syncrate = "t s"	時間間隔 t 秒で同期

これを用いた記述例を以下に与える.

```
<par syncrate="1.0s">
  <video src="video.mpg" region="a" >
  <audio src="audio.wav" />
</par>
```

この例では video.mpg と audio.wav の再生中に, 再生のずれが 1 秒ごとに調整される.

3 時間拡張 LOTOS への変換

拡張 SMIL 記述は, 時間拡張 LOTOS 記述 [3] に変換し, 我々が開発している時間拡張 LOTOS コンパイラ [2] を用いて実行可能コードに変換する.

3.1 時間拡張 LOTOS

形式記述言語 LOTOS[9] では, システムの仕様をいくつかのプロセスからなる並行プロセスとして記述する. 各プロセスの動作は**動作式**と呼ばれ, プロセス外部 (環境) から観測可能なアクションである**イベント**, 観測不可能な**内部イベント**, あるいはプロセス呼び出しの実行系列として定義される. ここでイベントは, 環境との相互作用 (データの入出力) であり, ゲートと呼ばれる作用点で生起する. イベント間の実行順序を指定するため, 連接 $(a;B)$, 選択 $(B1[]B2)$, 同期並列 $(B1|[gl]|B2)$, 非同期並列 $(B1|||B2)$, 割込み $(B1[>B2)$, 逐次 $(B1>>B2)$ などのオペレータが任意の部分動作式間に指定される. 特に, 同期並列オペレータを使用することにより, 複数のプロセスが指定されたゲート上のイベントを同時に実行しデータ交換を行なう, といった動作を記述することができる (**マルチランデブ**と呼ばれる). 我々は文献 [2] において, LOTOS に E-LOTOS[10] が持つ以下の以下の構文を追加した時間拡張 LOTOS を定義している.

構文	意味
loop B endloop	B を繰り返し実行する
var $V(,V)^*$ in B endvar	B で使用する変数群 (書換可能) の宣言
a@?t	イベント a が実行された時刻を変数 t に取得する
a@?$t[p(t)]$	イベント a は $p(t)$ を満たす時刻 t に実行できる
a@!T	イベント a は時刻 T に実行できる
wait(d)	d 単位時間待つ

(ここで，@オペレータが省略されたイベントは任意の時刻に実行可能である．時刻は各イベントが実行可能になってからの経過時刻であり，1 単位時間は 1ms に相当する)．

3.2 変換の概要

与えられた拡張 SMIL 記述の変換においては，(1) オブジェクトの再生順序の制御，(2) オブジェクトの再生動作，(3) オブジェクトに対する時間制御，をどのように行うかがポイントとなる．以下，それぞれについて説明する．

(1) オブジェクトの再生順序の制御

あるメディアの再生動作は，時間拡張 LOTOS により一つのプロセス (以降**再生プロセス**と呼ぶ) として記述される [2, 3]．

SMIL 記述における一つのオブジェクト obj-i の再生動作を一つの再生プロセス Play(obj-i) に対応させる．その再生プロセス Play(obj-i) がどのような順番で行われるかを簡単な変換例を用いて説明する．

par，seq エレメントによる複数オブジェクトの並列，逐次再生においては，各オブジェクトに対応した各再生プロセスの並列，逐次実行を行う一つのプロセスとして記述する．

```
( SMIL 記述 )
<par>                    <seq>
  <obj-1 … />            <obj-1 … />
  <obj-2 … />            <obj-2 … />
  <obj-3 … />            <obj-3 … />
</par>                   </seq>

( 対応する時間拡張 LOTOS 記述 )
Play_par:= Play(obj-1) ||| Play(obj-2) ||| Play(obj-3)
Play_seq:= Play(obj-1) >> Play(obj-2) >> Play(obj-3)
```

obj-i の再生の完了により再生プロセス Play(obj-i) が正常終了するならば，Play_par は obj-1, 2, 3 の並行再生が全て完了して終了し，また Play_seq は obj-1, 2, 3 が順番に再生されて終了する．

par，seq の入れ子構造でも同様に以下のように変換できる．

```
( SMIL 記述 )
<par>                    <seq>
  <obj-1 … />            <obj-1 … />
  <obj-2 … />            <obj-2 … />
  <seq>                    <par>
    <obj-3 … />            <obj-3 … />
    <obj-3 … />            <obj-4 … />
  <seq>                    <par>
</par>                   </seq>

( 対応する時間拡張 LOTOS 記述 )
Play_par':= Play(obj-1)1 ||| Play(obj-2) ||| Play_seq
Play_seq':= Play(obj-1)1 >> Play(obj-2) >> Play_par
```

Play_par' は obj-1, 2 の再生完了，及び obj-3, 4 の逐次再生が完了 (Play_seq が終了) することにより終了し，Play_seq' は obj-1, 2 が順番に再生された後，obj-3, 4 の並行再生が完了 (Play_par が終了) して終了する．

(2) オブジェクトの再生動作

再生プロセスでは，(1) 実際の再生操作を行う主動作 Main，(2) 連続メディアに対し指定した品質 (フレームレートなど) で再生するための品質制約 Restriction，を独立したプロセスとして時間拡張 LOTOS により記述し，これらをマルチランデブ機構を用いて同期実行させることで，オブジェクトを指定した品質で再生する [2, 3]．

動画の場合，主動作は 1 フレーム分のデータの読み込み，デコード，表示の繰り返しという動作，品質制約はそれらを指定した時間周期 (1/フレームレート) で行うという動作になる．音声の場合も同様に，適当な周期 (0.5 秒など) を設定し，主動作をその周期分の音声データの読み込み，再生の繰り返しとすることで実現される．

静止画，テキストなどの静止メディアに対しては，描画を実行する主動作のみで再生プロセスとなる．

動画や音声オブジェクトに対し，エレメント属性 clip-begin, clip-end が指定されている場合，それらにより有効になるデータのみが再生される必要がある．このため，再生プロセスに再生開始フレームおよび終了フレームを指定できるよう拡張した．これらの値は，オブジェクトのエンコードレート (動画の場合，フレーム／秒) と clip-begin, clip-end で指定された時間から算出される．

(3) オブジェクト再生における時間制御

SMIL 記述のオブジェクト再生においては，再生開始時刻と終了時刻が指定可能である．以下の例のように，再生終了時刻の代わりに再生期間が指定された場合，再生終了時刻は再生開始時刻＋再生期間となる．

```
<video src="video.mpg" region="a"
  begin="10s" dur="20s" />
```

この例では，video.mpg の再生を，現在から 10 秒後に開始し，その 20 秒後に再生を打ち切ることを指定している．

しかし，SMIL では，(i) 終了時刻が明示されていない場合や，(ii) 並行オブジェクトにおいて想定される終了時刻が複数ある場合，も存在するため，それらに場合についての制御を以下に述べる．

(i) 再生期間が明示されていない時

SMILの定義より，静止メディアの場合は再生期間0とみなし全く表示しない．連続メディアの場合，オブジェクト自体に含まれるデータをエンコードされた時のレートで全て再生し終るまでにかかる時間が再生期間となる．

(ii) 想定される再生期間が複数ある時

parエレメントに対し，エレメント属性endsync = first (最初に再生が終了する子エレメントにあわせてparエレメントが終了) が指定されている時，並列実行されている再生プロセスの内一つでも再生が完了したら全ての再生は途中でも終了させる．

このような場合の再生打ち切りの制御は，期間満了，又は他の並行再生プロセスの終了に伴う強制終了の複数条件を判断して行う．3つの再生プロセスをendsync=firstで制御する場合の記述例を以下に示す．

```
   Sync_First
|[vend,intr]
   (Play(obj1) ||| Play(obj2) ||| Play(obj3))
where
  Sync_First:= vend; intr; exit
  Play:= ( (再生動作) >> vend;exit) [> intr; exit
```

ここで，イベントvendは再生期間の満了を表すイベントであり，イベントintrは，再生中に強制終了するためのイベントである．

4 QoS制御の実現

本節では，SMIL記述に付加したQoS制御の時間拡張による実現方法について述べる．

4.1 優先度が指定された複数オブジェクトの品質制御

優先度，品質が指定された複数オブジェクトに対するQoS制御では，各オブジェクトが要求品質通りに再生できているかどうか，それをオブジェクトにかかる負荷という形で捉え，各オブジェクトの品質，優先度，負荷の情報を元に，どのオブジェクトの品質をどのように調整するかを決定する．

ここでは，(1) どのようにして品質を下げるか (maxtemporalcrop)，(2) オブジェクトにかかる負荷をどのように取得するか，(3) 優先度による制御をどのように行うか (priority)，が問題となる．以下，それぞれについて述べる．

(1) 品質の調整

30fpsの動画オブジェクトに対しmaxtemporal-crop="66%"と指定されているならば，フレームレートを10fpsまで落すことが許される．ここではこの動画の品質の情報として，30，15，10fpsの三段階があると考える．

15,10fpsへのフレームレートの変更は，それぞれ2フレームあたり1，3フレームあたり2のフレーム処理 (デコード，表示) を省くことにより行う．

(2) オブジェクトにかかるの負荷の取得

オブジェクトを指定した品質 (フレームレート) で再生できないという状態がある程度続いた時，ここではオブジェクトの負荷がより高くなったと考える．逆に指定したフレームレートで再生できる状態がある程度続いた時，より負荷が低くなったと考える．

ここでは，負荷の高さの情報を低，中，高の三段階で考え，一定期間にフレーム処理を周期 (1/フレームレート) 時間内に行えなかった割合がある値を超えたら負荷が高くなったと判断し，負荷の情報を一段回上げる．一定期間にフレーム処理を周期時間内に行えたという割合がある値を超えたら一段回下げるとする．

オブジェクトの再生プロセスから上記の負荷を取得するプロセスが文献[2]で与えられている．

(3) 優先度による制御

ある二つのオブジェクトA，Bに対しpriorityによりそれぞれ優先度が指定されている時，ここでAの負荷が高くなった場合次のようなポリシーで品質制御を行う．

1. A，Bが同じ優先度又はAがBよりも低いならば，Aの品質を下げる．これ以上下げられないならば，Bの品質を下げる．
2. AがBよりも優先度が高いならば，Bの品質を下げる．これ以上下げられないならば，Aの品質を下げる．

これを，AとBの負荷情報を低，中，高とした場合において，設定される品質30，15，10fpsを対応させると次の表のようになる．最初はA，B共に負荷低，30fpsで再生されているものとする．

1. AとBの優先度同じ

	A負荷 低	中	高
B負荷 低	A=30, B=30	A=15, B=30	A=10, B=30
中	A=30, B=15	A=15, B=15	A=10, B=15
高	A=30, B=10	A=15, B=10	A=10, B=10

2. AがBよりも優先度高い

	A負荷 低	中	高
B負荷 低	A=30, B=30	A=30, B=15	A=30, B=10
中	A=30, B=15	A=30, B=10	A=15, B=10
高	A=30, B=10	A=15, B=10	A=10, B=10

オブジェクトが三つ以上の場合も同様に，あるオブジェクトの負荷が高くなった場合，その時点で品質を下げられるオブジェクトの内，最も優先度の低いもの (同じ優先度が複数ある時はその中でかかっている負荷が低いものから選ぶ) の品質を下げる．この場合でも，上述のポリシーに基づき各オブジェクトの負荷と設定する品質の対応表が作れる．

各再生プロセスから負荷の情報を受け取り，その結果に応じて表のように各再生プロセスの品質を決定する制約プロセスを記述し，再生プロセス群と同期させることで，優先度と品質許容範囲に基づいたQoS制御を実現した．

4.2 代替メディアへの動的切り替え

オブジェクトの動的切り替えでは，動的に変化する情報を前節と同様オブジェクトにかかる負荷として考え，その負荷情報に応じ再生されるオブジェクトを決定するが，代替メディア切り替え時に内容の連続性の維持が問題となる．

ここでは，代替メディアに切り替える時，適切な量のデータをスキップすることにより内容の連続性を維持することにする．

2.2 節の例のような代替メディアが記述されている場合，負荷情報によりどのファイルを読み込み表示するか選択する．例えば video.mpg の再生中，再生したフレーム数をカウントしておき，video_smaller.mpg に切り替わった時には，その分だけ video_smaller.mpg のフレームを読み飛ばして，次のフレームを読み込み表示することにより場面を連続させる (切り替えをスムーズにするため，先読みバッファを使っている)．

テキストから動画への切り替えでは，テキスト表示時間分の動画をスキップする．

4.3 指定した時間間隔での同期

一定時間間隔の同期では，その時間ごとに各オブジェクトが待ち合わせを行うことにより実現する．

ここでは，動画，音声の連続メディアに対し，一フレームを動画では1コマ，音声では便宜的にある一定時間分のデータであるとする．

指定された時間間隔 (`s_interval`) と各オブジェクトのフレームレート `frate0`, `frate1` から，フレームの再生が指定された時間に何回行われるかが求められる (`cnt0`, `cnt1`)．

各再生プロセスにおいて一フレームの再生で一回行われるイベント (vin0!any, vin1!any) が求めた回数分行われたら，他の再生プロセスが同様に処理し終るまで待ち合わせるよう指定する．以上のメディア間同期に対する制約は，例えば，次のようなプロセスとして記述できる．

```
Mediasynch[vin0,vin1](frate0, frate1, s_interval):=
  loop
    ?cnt0:=frate0*s_interval;
    ?cnt1:=frate1*s_interval;
    while ((cnt1>0) or (cnt2>0)) do
      [cnt0 > 0]-> vin0!any; ?cnt0:=cnt0-1; exit
   [] [cnt1 > 0]-> vin1!any; ?cnt1:=cnt1-1; exit
    endwhile
  endloop
```

上記の制約プロセスを，2つの並行な再生プロセスとの間に，vin0, vin1 に対して同期指定することで，メディア間同期が達成される．同期すべき再生プロセスが3つ以上の場合も，同様の方法で制約プロセスを記述できる．

5 おわりに

本稿では，SMIL のサブセットに QoS 制御の機能を付加した拡張 SMIL を定義し，それを時間拡張 LOTOS の処理系を用いて実行することを提案した．

提案手法では，拡張 SMIL 記述を時間拡張 LOTOS 記述に変換する際，QoS 制御機能の拡張を制約指向記述スタイル [7] を用いて比較的容易に実現できるため，本方式は SMIL を拡張する際の，処理系のプロトタイピングに有用であると思われる．

より高度な QoS 制御機構やネットワークを介する動作の QoS 制御の拡張及びその実装が今後の課題である．

参考文献

[1] Fujikawa, Shimamura, Teranishi, Shimojo, Matsuura, Miyahara : "Application Level QoS Modeling for A Distributed Multimedia System", Proc. of 1995 Pacific Workshop on Distributed Multimedia Systems Honolulu, HI, pp.44-51 (March 1995).

[2] 東野 輝夫 他: "分散システムの設計開発のための LOTOS コンパイラの研究開発", 情報処理振興事業協会「独創的先進的情報技術に係わる研究開発」研究成果報告書, http://www-tani.ics.es.osaka-u.ac.jp/IPA/ , 71 page, 1999-03.

[3] 辰本 比呂記, 安本 慶一, 東野 輝夫, 安倍 広多, 松浦 敏雄, 谷口 健一 : "時間拡張 LOTOS を用いたマルチメディアシステムの記述とその実現", 情報処理学会マルチメディア通信と分散処理 (DPS) ワークショップ論文集, pp. 133 – 138 (1998).

[4] Synchronized Multimedia Integration Language (SMIL) 1.0 Specification, http://www.w3c.org/TR/REC-smil/

[5] W3C Synchronized Multimedia Activity Home Page, http://www.w3c.org/AudioVideo/

[6] ISO-QoS: "Quality of Service Basic Framework - Outline", *ISO/IEC JTC1/SC21/WG1 N1145* (1994).

[7] Vissers, C. A., Scollo, G. and Sinderen, M. v.: "Architecture and Specification Style in Formal Descriptions of Distributed Systems", *Proc. of 8th Int. Conf. on Protocol Specification, Testing, and Verification (PSTV'88)*: 189 – 204 (1988).

[8] Extensible Markup Language (XML) 1.0, W3C Recommendation, (SMIL) 1.0 Specification, http://www.w3c.org/TR/REC-xml/

[9] ISO : "Information Processing System, Open Systems Interconnection, LOTOS - A Formal Description Technique Based on the Temporal Ordering of Observational Behaviour", *ISO 8807* (1989).

[10] ISO: "Final Committee Draft 15437 on Enhancements to LOTOS", *ISO/IEC JTC1/SC21/WG7* (1998).