

Eclair:同期化コンテンツの放送型配信とその版管理方式の提案

野田 玲子[†] 角谷 和俊^{††} 田中 克己^{††}

本稿では、同期化コンテンツの配信方式と、そのバージョン管理方式を提案する。同期化コンテンツの配信方式として、サーバ側で各コンポーネントを同期化し、パッケージ化して配信するパッケージ型配信方式と、サーバ側でコンポーネントのみを配信し、クライアント側で動的に同期化するコンポーネント型配信の2つの方式を提案し、様々な情報に対して適用可能であることを述べる。

また、上記のどちらの配信方式についても、コンポーネントのバージョン管理をサーバ側とクライアント側で行うことにより、コンポーネント及びコンポーネントから成る同期化コンテンツの時間的一貫性を保つことが可能であることを示す。

Eclair:Broadcasting System and Version Control Mechanism for Synchronized Contents

REIKO NODA[†], KAZUTOSHI SUMIYA^{††} and KATSUMI TANAKA^{††}

In this paper, we describe a broadcasting system for synchronized contents, and we also describe a version control method for the system. We propose two broadcasting models for synchronized contents. One is a package delivery model, in which a server synchronizes components and delivers synchronized contents to its client. The other is a component delivery model, in which a server delivers components, and its client receives and synchronizes these components dynamically.

Furthermore, we propose a version control method by which the temporal consistency is kept at both the server and its client.

1. はじめに

近年、インターネットやデジタル放送を用いた情報配信サービスが注目を集めている。特に、ユーザがあらかじめ興味のあるチャンネルやソースを指定しておく、自動的に最新の情報が配信される PointCast Network¹⁾、Castanet2.0²⁾ などのプッシュ型サービスが普及している。

一方、1セットの独立したマルチメディアオブジェクトを1つの同期マルチメディア表現に統合することを可能にする同期マルチメディア統合言語 (the Synchronized Multimedia Integration Language) (SMIL1.0)³⁾⁴⁾ による同期化コンテンツによる情報提供が注目を集めている。また、SMIL コンテンツは複数のマルチメディアファイルから構成されているため、ファイルの削除などによりアクセスできない場合や、

時間の経過とともに情報の価値がなくなってしまうコンテンツにアクセスしてしまう場合がある。また、実際の同期化コンテンツ配信においては、情報が頻繁に更新されるため、情報の時間的な一貫性に矛盾を生じる可能性がある。

本研究では、これらの問題点を解決するために、同期化コンテンツの放送型配信の2つの方式を提案する。一つは複数のマルチメディアファイルを SMIL により同期化し、パッケージ化して配信を行うパッケージ型配信方式であり、もう一つは複数のマルチメディアファイルを配信し、クライアント側で動的に同期化を行うコンポーネント型配信方式である。また、情報の更新の際の時間的一貫性を保つためのバージョン管理方式についても議論する。

以下、本論文の構成を示す。まず、2 で本研究の動機である同期化コンテンツ配信における問題点について、及び本研究のアプローチについて述べる。3 では、同期化配信の2つの方式について述べる。4 では同期化コンテンツの時間的有効性の定義について述べる。5 では配信情報の時間的一貫性を保つためのバージョン管理方式について述べる。6 ではまとめと今後の課題について述べる。

[†] 神戸大学大学院自然科学研究科情報知能工学専攻
Division of Computer and Systems Engineering, Graduate School of Science and Technology, Kobe University
^{††} 神戸大学大学院自然科学研究科情報メディア科学専攻
Division of Information and Media Sciences, Graduate School of Science and Technology, Kobe University

2. 本研究の動機

2.1 同期化コンテンツの配信

近年、1セットの独立したマルチメディアオブジェクトを1つの同期マルチメディア表現に統合することを可能にする同期マルチメディア統合言語 (SMIL1.0)³⁾⁴⁾ によるテレビ番組レベルのマルチメディアプレゼンテーションを作成する枠組みが提案されている。

SMIL は、2つの主要な要素 seq と par からなる。要素 seq は2つのマルチメディアファイルを順に再生する。要素 par は2つのメディアを同時に再生する。この2つを組み合わせることで、複雑なメディアの同期を記述することができる⁴⁾。

ニュースや天気予報、コンサートのライブ中継、名所の紹介など、さまざまな種類のコンテンツがこの SMIL により作成され、インターネット上で公開されており、RealSystemG2⁵⁾ などのブラウザで閲覧可能である。これらの同期化コンテンツによる情報提供における問題点は以下のとおりである。

- 同期化コンテンツ (SMIL コンテンツ)⁶⁾ は、自動的に配信されるわけではなく、あらかじめブラウザに設定されているチャンネルを指定するか、Web 上でリンクをたどったり、コンテンツの場所を指定することでアクセスしなければ閲覧することができない。
- 同期化コンテンツは複数のマルチメディアファイルから構成されているため、ファイルの削除などによりアクセスできない場合や、時間の経過とともに情報の価値がなくなってしまったコンテンツにアクセスしてしまう場合がある。
- ニュースや天気予報のようなプレゼンテーション形式のコンテンツはあらかじめサーバ側で同期の記述を行い、配信することが好ましい。それに対して、スポーツ中継などのリアルタイムな情報を同期化して配信するときは、サーバ側であらかじめ同期の記述を行うことが難しい。

したがって、様々な情報に対応した同期化コンテンツの配信方式、及び一貫性管理の機構が必要となる。

我々は、これまで、放送型配信システム *Mille-feuille* の研究を行ってきた^{6)~9)}。放送型配信システム *Mille-feuille* は、ドキュメントの配信を対象とし、配信情報の関連性、及び時間的一貫性を保つための枠組みを提案した。しかし、同期化コンテンツの配信については対応していない。本研究では、同期化コンテンツの配信方式及び、一貫性の管理の方式を提案する。

2.2 同期化コンテンツの一貫性管理

同期化コンテンツの放送型配信では、複数のマルチメディアファイル (以下コンポーネントと呼ぶ) を同期化して配信を行う。それぞれのコンポーネントは時間経過に伴い更新される。また、時間の経過とともに情報の価値がなくなることもある。したがって、同期化コンテンツ配信では、各コンポーネントに有効時間を付加することで、クライアントが価値のない配信情報にアクセスできないようにすることが必要である。

ここで、情報の時間的な有効性は、内容が更新された場合に変更される可能性がある。例えば、内容が更新され、新しいバージョンが作成された際に、古いバージョンの有効時間が修正される場合や、情報が削除された場合などである。クライアントでは、このような有効時間の変更は、新しいバージョンを受信することによってのみ修正可能である。しかし、放送型配信ではサーバからクライアントに対して一方的にコンテンツが配信されるため、クライアントが受信状態ではない場合、配信された情報を受信できない可能性がある。

したがって、クライアントがどのような状態の場合でも、蓄積されている情報の有効時間に矛盾が生じないようにするための機構が必要である。すなわち、有効時間の一貫性を保つためのバージョン管理を行う機構が必要である。本論文では、同期化コンテンツの放送型配信のためのバージョン管理方式の提案を行う。

3. 同期化コンテンツの放送型配信方式

3.1 パッケージ型配信

サーバ側で、複数のマルチメディアファイルを SMIL により同期化し、一つのプレゼンテーションとしてクライアントに配信する。各コンポーネントは同期化コンテンツとしてパッケージ化して配信されるため、コンポーネントが更新されると、それを含む同期化コンテンツも更新され、更新されたコンポーネントとそれを含む SMIL ソースのバージョンがクライアント側にパッケージとして配信される。図1は、パッケージ型配信を示している。

各コンポーネントには有効時間が付加されており、配信された同期化コンテンツが再生される際にこの有効時間のチェックを行うことで、価値のなくなったコンポーネントを含む同期化コンテンツや、サーバ側で削除されたコンポーネントを含む同期化コンテンツが再生されることを防ぐことができる。

パッケージ型配信の例として、天気予報やニュース番組の配信があげられる。SMIL を用いたニュースの配信を考えると、ニュースはいくつかの動画やテキス

☆ 以下、SMIL コンテンツのことを同期化コンテンツと呼ぶ。

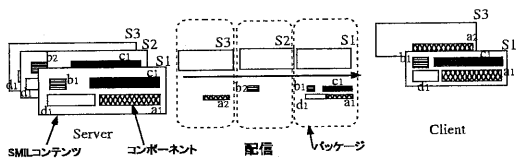


図 1 パッケージ型配信

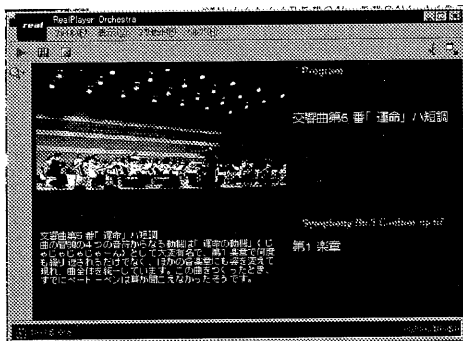


図 2 パッケージ型配信の表示例

トなどのコンポーネントから構成されており、それぞれのコンポーネントの内容ごとに、情報の有効時間を付与することで、ユーザに対して適切な時刻にその内容を提示することができる。また、コンポーネント単位で管理を行うことで、内容の更新や、有効時間の変更などの管理が容易になる。

例えば、天気予報は「天気図」（画像）や「明日の天気」（テキスト）「降水確率」（テキスト）などから構成された同期化コンテンツと考えることができるが、「天気図」は6時間に1回更新、明日の天気は1日1回更新など、更新間隔が異なるため、コンポーネントごとの版管理を行う必要がある。また、6月1日の「明日の天気」という情報は6月1日中に見ないと意味がない情報である。したがって、「明日の天気」というコンポーネントには、[6/1 0:00-6/1 23:59]という有効時間が指定されていることになり、この時間外に再生されることのないようにしなければならない。

図2はオーケストラのコンサートの模様を同期化コンテンツとしてパッケージ化して配信した例である。

3.2 コンポーネント型配信

コンポーネント型配信では、サーバ側で作成/更新され、配信されるコンポーネントを、クライアント側で動的に同期化を行い、同期化コンテンツに合成する。更新単位はコンポーネントとなる。クライアント側で合成される同期化コンテンツはクライアント側に蓄積され、参照することが可能である。図3はコンポーネント配信を示している。

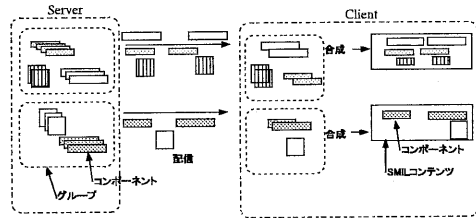


図 3 コンポーネント型配信

コンポーネント型配信では、ある程度の同期情報をクライアントが保持しているか、コンポーネント自身が別のコンポーネントとの同期情報を持っていると仮定し、その情報を元にサーバから配信されてくるコンポーネントを同期化する。

クライアントにおける動的な同期化の一方式として、以下の方式が考えられる。サーバから配信される各コンポーネントのうち、相互に関連したデータが存在するとする。コンポーネントは更新され、バージョンアップが行われるごとに配信される。サーバ側で各コンポーネントは id で管理されており、ある id を持つコンテンツがバージョンアップされた場合、新しいバージョンと古いバージョンは同じ id を付与されて送信される。従って、あるコンポーネントのバージョンは、コンポーネント id とバージョン番号 ver の組で一意に決まる。

ここで、サーバは関連するコンポーネントを一つのグループとして配信を行う。クライアント側では一つのグループから一つの同期化コンテンツが動的に生成される。

グループ内の相互に関連するコンポーネント間には以下の3つの関連づけが行われており、その関連づけが一つの同時再生区間 (par) に合成される。

- (1) (id_1, id_2)
- (2) $(id_1, (id_2, ver))$
- (3) $((id_1, ver_1), (id_2, ver_2))$

(1) はコンポーネント同士を同時再生し、そのコンポーネントのバージョンを順次再生する。(図4(1))

(2) はコンポーネント1とコンポーネント2のあるバージョンを同時再生する。コンポーネント1のバージョンは順次再生される。(図4(2))

(3) はコンポーネント1のあるバージョンと、コンポーネント2のあるバージョンを同時再生する。(図4(3))

コンポーネント型配信の例として、F1中継の配信が考えられる。サーバ側で配信されるデータは、レース映像、タイムなどのデータ、クラッシュの映像とそ

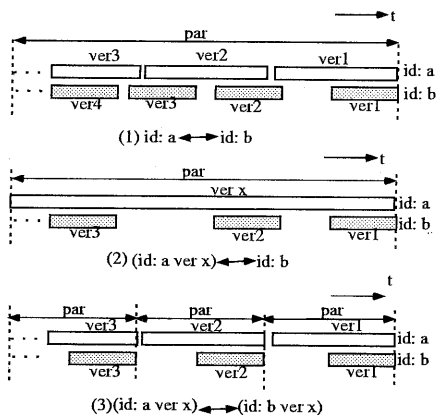


図4 コンポーネントの動的同期化

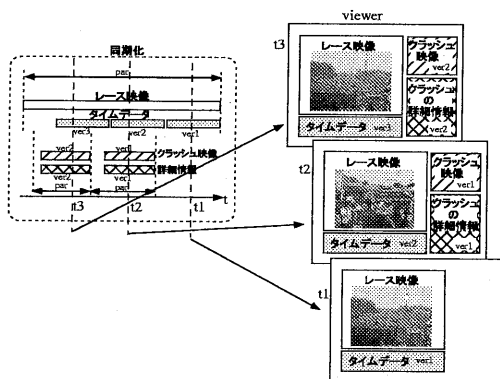


図5 コンポーネント型配信の表示例

の音声解説などである。例えば、レース映像とタイムのデータは上記の(2)で対応づけし、クラッシュの映像と音声解説は上記の(3)で対応づけると、クライアント側では図5のような同期化コンテンツが生成される。また、タイムデータやクラッシュ映像などは頻繁に更新されるため、バージョンが到着する毎に、クライアントは同期化コンテンツを再合成する。

4. 配信コンテンツの時間的有效性

4.1 同期化コンテンツの定義

同期化コンテンツ S はいくつかのコンポーネントの c_i の集合である。したがって、同期化コンテンツ S およびコンポーネント c_i は以下のように定義される。

$$S = \{c_i | i = 1, 2, \dots, n\}$$

$$c_i = \{p_i, d_i, [v_{i1}, v_{i2}]\}$$

ここで、 p_i は c_i の再生時間であり、テキストや画像などの静止メディアの再生時間は SMIL ソースにより与えられる。

d_i は c_i の再生開始時間であり、 S が再生されてから c_i が再生を開始するまでの時間である。これは再生時にクライアント側で SMIL ソースより与えられる値である。

$[v_{i1}, v_{i2}]$ は c_i の有効時間である。有効時間は各コンポーネントの有効期限であり、コンポーネントが再生を開始することが可能な時区間を表す。有効時間は絶対時間で指定する。

4.2 同期化コンテンツの再生可能区間

時刻 t において S を再生するとき、 S の全てのコンポーネント c_i について、 c_i が再生される時刻、つまり時刻 $t + d_i$ が c_i の有効時間 $[v_{i1}, v_{i2}]$ を含んでいなければならない。言い換えると、時刻 t に c_i が再生可能か否かは、区間 $[v_{i1} - d_i, v_{i2} - d_i]$ が時刻 t を含んでいるかどうかによって決まる。したがって、 S が再生可能となる時刻 V_1 は、 S の全てのコンポーネント c_i が時刻 t に再生可能とならなければならない。つまり、

$$V_1 = \max\{v_{i1} - d_i, v_{i2} - d_i, \dots, v_{n1} - d_n\}$$

となる。

また、 S が再生可能でなくなる時刻 V_2 は、 S のいずれかのコンポーネントが再生可能でなくなる時刻と考えることができる。つまり、

$$V_2 = \min\{v_{i2} - d_i, v_{i2} - d_i, \dots, v_{n2} - d_n\}$$

となる。

したがって、同期化コンテンツ S の再生可能時間は、 $[V_1, V_2]$ となる。

5. 同期化コンテンツ配信におけるバージョン管理

配信されるコンテンツは、サーバ側で時々刻々と変更される可能性がある。サーバ側でコンテンツが更新される毎に、新しいバージョンを配信し、クライアント側ではそのバージョンを蓄積する。このとき、何らかの理由によりクライアントが受信できない可能性がある。例えば、クライアント側が受信状態ではない場合などである。従って、本研究では以下の環境を前提としている。

- コンテンツは時々刻々と更新され、配信される
- 更新情報はすべてサーバに蓄積される。
- 配信された更新情報は各クライアントですべて受信されるとはかぎらないが、受信した情報は各クライアントに蓄積される。

この環境において、クライアント側でユーザーに時間的に矛盾のない情報を提示するためには、それぞれのコンポーネントのすべてのバージョンに関して、有効時間や再生時間、再生開始時間が正しくなければいけ

ない。

同期化コンテンツの配信では、パッケージ型配信、コンポーネント型配信の双方において、クライアント側に蓄積されているコンポーネントの有効時間と再生開始時間から同期化コンテンツの有効時間が計算される。ここで、コンポーネントの各バージョンの有効時間は、サーバ側で更新されるため、クライアントがその変更を受信しないと、蓄積しているバージョンの有効時間に矛盾を生じる。

すなわち、パッケージ型配信、コンポーネント型配信のどちらの場合においても、コンポーネントの各バージョンの有効時間の一貫性を保つことが重要である。

ここで、Mille-feuilleにおける、有効時間に矛盾を生じないためのバージョン管理方式⁹⁾をSMILの各コンポーネントのバージョン管理に適用することで、各コンポーネントの時間的一貫性を保つことができる。

5.1 サーバ側におけるバージョン管理

サーバ側でのコンテンツの更新は、2つの種類にわけに行く必要がある。一つが内容の更新、もう一つが有効時間の更新である。本研究ではこの二つの更新を、2分木を用いることで制御する。

5.1.1 バージョン木

バージョン木は2分木であり、ノードが各バージョンを示す。また、あるノードの左の子節点はそのノードの内容が更新された次のバージョンであり、右の子節点は有効時間の変更が行われたバージョンである。

バージョン木 T は以下のように定義される。

$$T = (V, E)$$

$$V = \{c_1, c_2, \dots, c_n\}$$

$$E \subseteq \{(c, c') | c \in V, c' \in V\}$$

$$c = (id, version, tr, p, [t_1, t_2]), c \in V$$

V はバージョン木のノードであり、配信コンテンツの更新管理の対象となる同期化コンテンツを構成するコンポーネントの各バージョンの集合を表す。

E はバージョン木の枝である。左の枝が内容の更新を表し、右の枝が有効時間の変更を示す。

c はバージョン木 T のノードであり、コンポーネントの各バージョンを示す。ここで、 id は識別子、 $version$ はバージョン番号、 tr はコンポーネントのトランザクション時間、 p はコンポーネントの再生時間、 $[t_1, t_2]$ はコンポーネントの有効時間である。バージョン番号は、メジャー番号とマイナー番号の2つの組で表される。メジャー番号が内容の更新、マイナー番号が有効時間の更新を表す。バージョン番号が (i, j) のコンポーネントを c_{ij} と表す。

コンポーネントが更新や削除された際、必要な場合には、過去のバージョンの有効時間を変更し、右の子節点として追加する。そのノードを配信することで、

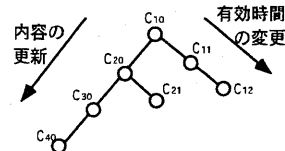


図6 バージョン木

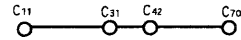


図7 バージョンリスト

クライアント側にバージョンの削除や有効時間の変更を通知する⁹⁾。

5.1.2 バージョンの配信

サーバは各クライアントにコンポーネントのバージョン木のノードのうち、クライアントに配信されているノードと、サーバ側でのバージョン木の葉節点との差分のノードを配信する。例えば、図8において、サーバは、3番目におけるクライアントに配信済のノード $N_C(t) = \{c_{11}, c_{30}\}$ と、4番目のサーバ側における葉のノード $N_S(t) = \{c_{11}, c_{21}, c_{31}, c_{40}\}$ との差分である $N_S(t) - N_C(t) = \{c_{21}, c_{31}, c_{40}\}$ をクライアントに配信する。

5.2 クライアント側におけるバージョン管理

5.2.1 バージョンリスト

クライアント側でのバージョン管理は、各コンポーネントのバージョンリストを生成することで行う。バージョンリストは線形リストであり、ノードが各バージョンを表す(図7)。クライアントは起動中に、サーバ側でのコンポーネントのバージョン木の葉のバージョンを1回だけ受信することができる。

コンポーネントのバージョンリスト l は以下のように定義される。

$$l = [c_1, c_2, \dots, c_n]$$

$$c = (id, version, tr, p, [t_1, t_2], ar)$$

ここで、 $id, ID, version$, トランザクション時間 tr , 有効時間 $[t_1, t_2]$, 再生時間 p はサーバ側のノードと同じものである。 ar はそのノードの到着時間である。バージョンリストでは、バージョン番号 (i, j) の i は必ずしも連続しているとは限らない。なぜなら、クライアントがシステムを起動していない間に、受信できないノードがあるからである。

クライアント側では、サーバから配信された新しいコンポーネントのバージョンに対して、追加、挿入、交換を行うことで、常に有効時間の正しいノードのみを蓄積する⁹⁾。

5.2.2 クライアントにおけるコンテンツの一貫性

クライアント側では、システムを起動していないと

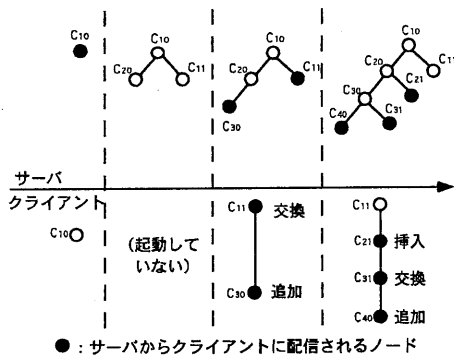


図8 クライアントにおける処理

受信できない。しかし、サーバ側において有効時間が変更されたノードは、クライアントがシステムを起動したときに、必ずクライアント側に配信される。なぜならば、サーバは、クライアント側のノードとサーバ側での葉のノードとの差分を配信するためである。クライアント側では、そのノードと配信済みのノードを交換することによって、有効時間が正しいノードのみを蓄積する。したがって、受信できないノードはあるが、蓄積されるノードの有効時間は正しい。

また、各コンポーネントの各バージョンの有効時間が正しいので、そのコンポーネントのバージョンから成る同期化コンテンツの有効時間も正しい。同期化コンテンツの有効時間は、それぞれのコンポーネントがもっている有効時間や再生時間、再生開始時間が正しくなければいけないが、再生時間は、各バージョンに固有で、変化しない値である。再生開始時間は、SMILソースにより求められる値なので、再生時にクライアント側で計算可能である。また、コンポーネントの各バージョンの有効時間は、ここまで述べてきたバージョン管理方式により、常に一貫性が保たれている。よって、同期化コンテンツの有効時間の一貫性も保たれることになる。

したがって、クライアント側に蓄積されているすべての情報の時間的一貫性が保たれる。

6. おわりに

本稿では、同期化コンテンツのための放送型配信方式の提案と、配信情報の時間的一貫性を保つためのバージョン管理方式について述べた。今回提案した配信方式は、ニュースなどのパッケージ化された情報の配信、スポーツ中継などのリアルタイム情報の配信など、様々な情報配信に適用することができる。また、今回提案したバージョン管理方式により、クライアントがシステムを起動していない期間に受信できなかつ

た情報があつたとしても、情報の有効時間に対する一貫性が保たれることを示した。

今後の課題として、配信時間や到着時間の導入と、それによる一貫性の精密な制御があげられる。また、クライアント側で配信されるコンポーネントの内容に基づいた動的な同期化コンテンツの生成機能、及び、情報フィルタリング機能の検討が必要である。また、試作システム *Eclair* の実装が挙げられる。

謝辞 本研究の一部は、文部省科学研究費重点領域研究「高度データベース」(領域番号 275 (08244103)) の援助を受けています。また、本研究の一部は、日本学術振興会未来開拓学術研究推進事業における研究プロジェクト「マルチメディア・コンテンツの高次処理の研究」(プロジェクト番号 JSPS-RFTF97P00501) によっています。また、本研究の一部は、次世代情報放送システム研究所(株)の委託研究の援助を受けています。ここに記して謝意を表すものとします。

参考文献

- 1) PointCast: PointCast Network, <http://www.pointcast.com>.
- 2) Lemay, L.: *Official Marimba Guide to Castanet*, Sams.net(松田晃一ほか訳: Marimba オフィシャルガイド Castanet, トップラン, 1997) (1997).
- 3) SMIL: (Synchronized Multimedia Integration Language), <http://www.w3.mag.keio.ac.jp/AudioVideo/>.
- 4) Liu, P.: An Introduction to the Synchronized Multimedia Integration Language, *IEEE Multimedia*, Vol. 5, No. 4, pp. 84-88 (1998). <http://www.computer.org/multimedia/mu1998/u4toc.htm>.
- 5) G2, R.S.: Real Networks, <http://www.real.com/>.
- 6) 野田玲子, 角谷和俊, 田中克己: 時間依存リンクを用いた情報配信システム Mille-feuille の設計, 電子情報通信学会第9回データ工学ワークショップ (DEWS'98) (1998).
- 7) Sumiya, K., Noda, R. and Tanaka, K.: Hypermedia Broadcasting with Temporal Links, *Proc. of 9th International Conference on Database and Expert Systems Applications (DEXA'98)*, pp. 176-185 (1998).
- 8) 角谷和俊, 野田玲子, 田中克己: 放送型ハイパーメディアのための時間依存リンク機構, 電子情報通信学会論文誌 D-I, Vol. J82-D-I, No. 1, pp. 291-302 (1999).
- 9) 野田玲子, 馬強, 角谷和俊, 田中克己: 放送型情報提供システム Mille-feuille における時間依存リンク情報の配信, 情報処理学会研究会報告 99-DBS-116 (1998).