次世代コンテンツモデル MCOM

鈴木 正[†] 池田哲夫[††,☆]
青木輝勝[†] 安田浩[†]

本稿では，次世代のマルチメディアコンテンツに必要な要件として，1) コンテンツの再利用（Reuse），2) ユーザニーズに最適化されたコンテンツの提供（Adaptation），3) コンテンツの著作権管理の支援（Copyright Management）をあげ，これらの要件を同時に満たすことが可能な新たなコンテンツモデル MCOM（Multimedia Content Object Model）を提案する．著者らは，従来，データとして取り扱われていたコンテンツを，オブジェクト指向アプローチを用いて，メタ情報，コンテンツ（コンテンツデータ），および振舞いを備えたコンテンツオブジェクトとして再定義した．また，インタフェースと実装の分離，あるいはサブクラスによる拡張といったオブジェクト指向の特徴をコンテンツモデルに適用可能なように設計した．また，MCOM の設計とともに Java 言語を用いて実装した MCOM フレームワークについて述べる．さらに，具体的なコンテンツを作成することにより，上述した要件を満たすことを示す．

MCOM: A Next Generation Content Model

Tadashi Suzuki,[†] Tetsuo Ikeda,[††,☆] Terumasa Aoki[†]
and Hiroshi Yasuda[†]

In this paper, we focus on 1) Reuse, 2) Adaptation and 3) Copyright Management as requirements for next-generation content models and propose MCOM (Multimedia Content Object Model) for this purpose. By this new content model, we can redefine content based on object-oriented approaches, and we show MCOM makes it possible to support the above 3 requirements. Furthermore, MCOM has a lot of advantages such as separation of interface and implementation, and extension using sub classes etc. In this paper, we first propose MCOM architecture and after that we present the MCOM framework developed by Java as an implementation language. Finally we actually produce MCOM content and show that the MCOM framework satisfies all of the above requirements.

1. はじめに

インターネットの普及，パーソナルコンピュータの高性能化，ネットワークのブロードバンド化が進むにつれ，マルチメディアコンテンツが広く利用されるようになった．このような中で，コンテンツクリエータが既存のコンテンツを素材として再利用（Reuse）したいという要求が顕著化している．この要求に応えるものとして，既存のコンテンツを素材として再利用可能なコンテンツ記述方式[1] が提案されている．また，コンテンツをユーザニーズに最適化（Adaptation）したいという要求も顕著化している．この要求に応えるものとして，ユーザニーズに最適化可能なコンテンツ記述方式[2] が提案されている．

一方，マルチメディアコンテンツをビジネスでの用途を主としてインターネット上に流通させるコンテンツ流通が今後最も期待されるアプリケーションの 1 つとなりつつある．しかしながら，コンテンツ流通が広く普及していくにあたり，その促進を妨げる著作権管理上の問題が指摘されている．著作権管理（Copyright Management）のために，著作権情報記述方式[3),4)] および，カプセル化アプリケーション[5)~8)] が提案されている．

しかしながら，これらの既存のモデルおよび方式は，**再利用（Reuse）**，**ユーザニーズ最適化（Adaptation）**，**著作権管理（Copyright Management）**のすべての要求を同時に達成することが

† 東京大学先端科学技術研究センター
Research Center for Advanced Science and Technology, The University of Tokyo
†† 日本電信電話株式会社 NTT サイバースペース研究所
NTT Cyber Space Laboratories, NTT Corporation
☆ 現在，岩手県立大学大学院ソフトウェア情報学研究科
Presently with Graduate School of Software and Information Science, Iwate Prefectural University

できない．これら 3 つの要求はそれぞれ，コンテンツクリエータ，エンドユーザ，コンテンツホルダの要求であり，コンテンツのライフサイクルの中で同様に重要である．今後のコンテンツ流通を考えるうえでこれら 3 要求を同時に満たすことができる新たなコンテンツモデルの構築が必要不可欠である．

著者らは，新たなコンテンツモデルを設計するにあたり，最初に，コンテンツモデルの役割を明確化することから始めた．指摘されている著作権管理上の問題の中で，必ずしもすべての機能をマルチメディアコンテンツが解決するのではなく，他の仕組みが解決し，マルチメディアコンテンツはそれと協調することが可能であるようにすべきであるものが含まれているためである．これまで一般に著作権管理上の問題として以下のようなものが指摘されている[4]．

A) 「劣化なく複製が容易」というディジタルの特徴により，著作権者が安心してディジタルコンテンツをネットワークに流せない．

B) 課金・決済のしくみが乏しく，著作権者の支払い方法が困難．

C) コンテンツの権利関係などを調べる手段が少なく，特に要素片の組み込みを中心に，再利用（編集・加工）が不安でできない．

D) ディジタルコンテンツを大量にデータベース化して相互利用する際，アクセス・検索する共通的な識別体系がない．

そこで本稿では，コンテンツプラットフォームレイヤ，コンテンツモデルレイヤ，コンテンツアプリケーションレイヤという 3 つのレイヤを定義し，それぞれのレイヤが解決するべき問題を明確化することから設計を開始する．

コンテンツプラットフォームレイヤ このレイヤはマルチメディアコンテンツに特化しない多くの一般的なシステムを含む．たとえば，ネットワーク，パーソナルコンピュータ，スピーカやモニタなどが含まれる．このレイヤの責務の 1 つとしてマルチメディアコンテンツを含むディジタルデータがキャプチャされたり，コピーされたりすることを防ぐことがあげられる．この定義に従うと上記の問題 A) はこのレイヤで解決されるべきものである．

コンテンツモデルレイヤ 本稿で提案するマルチメディアコンテンツモデルはこのレイヤに属する．マルチメディアコンテンツモデルに従って作成されたマルチメディアコンテンツは，メタ情報へアクセスする方法を提供しなければならない．また，マルチメディアコンテンツ自身のデータを不正な操作からデータを保護しなければならない．この定義に従うと，問題 C) はこのレイヤで解決されるべきものである．また，再利用およびユーザニーズ最適化もコンテンツモデルが実現するものとし，このレイヤで解決されるべきものとする．

コンテンツアプリケーションレイヤ このレイヤはマルチメディアコンテンツのための様々なアプリケーションおよびシステムを含む．たとえば，コンテンツの ID を発行・管理するシステム，大量のマルチメディアコンテンツを保存・管理するシステム，課金システムなどが含まれる．この定義に従うと，問題 B) および D) はこのレイヤで解決するべきものである．

以上のレイヤ定義に基づき，本稿では，再利用，ユーザニーズ最適化に加え，著作権に関する問題の中から問題 C) をコンテンツモデル自身が解決すべき問題として取りあげることとする．

2. 次世代コンテンツの要求項目と既存方式の分析

2.1 次世代コンテンツの要求項目

新たなマルチメディアコンテンツモデルが達成すべきものとして，1 章で述べた 3 要求条件をあげる．また，これを以下のように 5 要求項目に細分化する．再利用を支援するコンテンツの要求項目として「再利用可能性」と「更新可能性」，ユーザニーズ最適化を支援するコンテンツの要求項目として「適応可能性」，著作権管理を支援するコンテンツの要求項目として「参照可能性」と「保護可能性」をあげる．

再利用可能性（Reusability） マルチメディアコンテンツは素材として様々な既存コンテンツを利用できなければならない．

更新可能性（Renewability） マルチメディアコンテンツは，利用している素材コンテンツの最新状態を人の手を介さずに自動的に反映できなければならない．このような特徴は，更新頻度が高く，また最新の情報の提供が要求される価格，地図，天気予報などを素材コンテンツに含むコンテンツに特に有用であると考えられる．

適応可能性（Adaptability） マルチメディアコンテンツはユーザニーズに最適化した形でコンテンツを提供可能でなければならない．ユーザニーズへの最適化として，ユーザ環境に対する最適化あるいは，ユーザ興味分野に対する最適化があげらている[2]が，さらに，コンテンツを素材として利用する際，そのまま利用することだけでなく，利用側コンテンツが必要な形に編集することも重要なユーザニーズへの最適化の 1 つと

表 1 既存研究の比較
Table 1 Comparison of existing researches.
（○ サポート，△ 部分的サポート，× サポートしていない）

	MPEG 1/2	cIDf	RightsShell	SMIL2.0	ZyX
Reusability	×	×	×	○	○
Renewability	×	×	×	○	○
Adaptability	×	×	×	△	△
Referenceability	×	○	○	△	△
Protectablility	×	×	○	×	×

してとらえ，これを満たすことも適応可能性の条件とする．

参照可能性（Referenceability） マルチメディアコンテンツはそれ自身のメタ情報はもちろん，素材として利用したマルチメディアコンテンツのメタ情報へアクセスする方法を提供しなければならない．これにより，コンテンツの利用者はこのメタ情報を用いて著作権情報などを調べることができる．また，メタ情報は，著作権者への連絡先，利用料金，利用可能期間など利用方法によって様々なものが考えられるため，項目を拡張することが可能でなければならない．

保護可能性（Protectability） マルチメディアコンテンツはコンテンツ 2 次利用者が編集をともなってコンテンツを再利用する際に著作権者の意図しない，誤った編集操作や不正な編集操作からコンテンツを保護できなければならない．

2.2 既存方式の分析

本節では 2.1 節であげた 5 要求項目に対する既存方式の対応状況について述べる．既存方式として，すでに広く利用されているコンテンツモデルとして MPEG1/2 を，再利用とユーザニーズへの適応を考慮したコンテンツモデルとして SMIL2.0[1] と ZyX[2] を，著作権保護を目的としたものとして cIDf[4] と RightsShell[6] を取りあげる．また，**表 1** にこれらの既存方式の要求項目への対応状況を示す．

2.2.1 MPEG1/2

すでに広く使われているマルチメディアコンテンツモデルであるが，再利用，ユーザニーズ最適化，著作権情報管理の機能は持っていない．たとえば，他のコンテンツを素材として利用する際には複製する必要があり，素材の最新の状態を反映させるためにはコンテンツを作り直す必要がある．ユーザニーズに最適化する仕組みや，不正な編集操作から自身を保護する仕組みもない．

2.2.2 SMIL2.0

テキスト，動画，音声などのマルチメディアコンテンツを同期させながら表示する方法を記述するための規格であり，マルチメディアドキュメントモデルもしくはマルチメディアコンテンツに分類できる．既存のマルチメディアコンテンツの再利用のほかに SMIL ファイル自身も素材として利用可能である．また，ユーザニーズの最適化のための機能として提供されているのはエンドユーザの端末や通信能力に応じてコンテンツを分岐条件にて切り替える単純な方式のみであり，コンテンツが素材として利用される際に，編集をともなって利用されることを考慮していないため，そのような場合には素材コンテンツを複製し，手作業で作り直す必要がある．この点で適応可能性の要求を満たしているとはいえない．著作権管理に関してはメタ情報記述に含まれるいくつかの項目を記述することだけであり，不正な編集操作などからコンテンツを保護する仕組みがなく，保護可能性を満たしていない．

2.2.3 ZyX

SMIL1.0[9] をベースとし，いくつかの拡張を行っている．SMIL1.0 よりも高度なユーザニーズへの適応が可能であるが SMIL 同様に，編集をともなって利用されることが考慮されておらず適応可能性は十分でない．また，保護可能性も満たしていない．

2.2.4 cIDf

コンテンツの著作権情報を管理するための XML[10] を用いた著作権情報の記述方式である．多くの著作権管理のための多くの項目をタグとして定義しており，また拡張も可能である．しかしながら，コンテンツの利用に際して，それらの記述内容を守るか守らないかは利用者の判断に依存してしまう．MEPG1/2 などのコンテンツとともに利用するが解決するのは参照可能性のみである．

2.2.5 RightsShell

コンテンツ配信の際，データの改竄やコピーからコンテンツを保護するためのアプリケーションである．コンテンツの保護と同時に課金も考慮している．既存コンテンツの利用を前提として著作権管理を主目的としているため，素材としての再利用およびユーザニーズ最適化は考慮されていない．

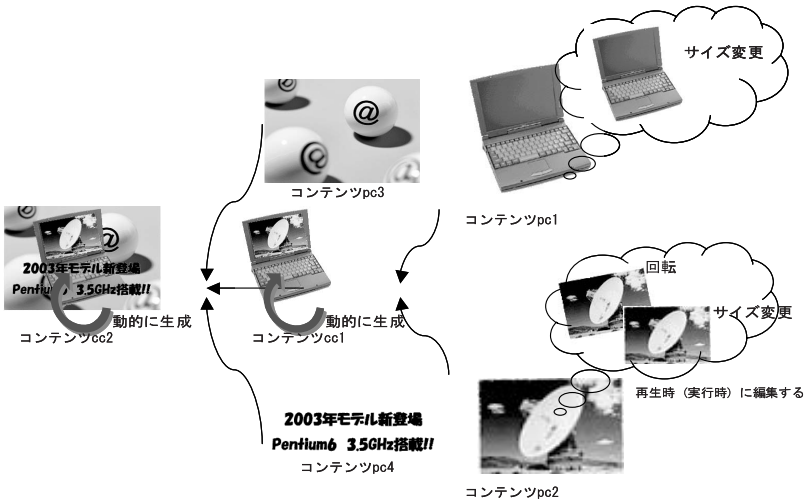

図 1 MCOM の特徴
Fig. 1 Characteristics of MCOM.

3. コンテンツモデル MCOM の提案

本稿では，再利用（Reuse），ユーザニーズ最適化（Adaptation），著作権管理（Copyright Management）のすべての要求を同時に達成するため，オブジェクト指向アプローチを用いて，新たなコンテンツモデル MCOM（Multimedia Content Object Model）を提案する．また，このコンテンツモデルを設計するにあたり，高速ネットワークおよび高速処理を前提とした．次の 3.1 節で MCOM の特徴を簡単に述べた後，3.2 節でコンテンツ記述方式について述べる．また，3.3 節以降では MCOM の詳細設計について述べる．

3.1 MCOM の特徴

MCOM とはこれまでデータとして扱われてきたコンテンツをオブジェクト指向のアプローチを用いて再定義したモデルであり，以下のような特徴がある．

コンテンツのオブジェクト化 MCOM では，従来，データとして取り扱われていたコンテンツを，メタ情報，コンテンツ，および振舞いを備えたコンテンツオブジェクトとして再定義する．コンテンツは自律的であり，クライアントの要求に応え，メタ情報をクライアントに渡す，あるいは，内部のコンテンツを編集するなどの作業を自ら行うことが可能となる．

階層構造 マルチメディアコンテンツが作成される際，多くの場合に，素材として，すでにあるコンテンツを利用すると考えられる．一般に，これらの作業はコンテンツ作成時に行われる．しかしながら MCOM では，素材コンテンツを静的に結合させず，利用側コンテンツでは素材として利用するコンテンツの参照先情報と素材を利用する方法の情報を保持する．この階層構造の特徴により，素材として利用しているコンテンツの参照を再帰的にたどることにより素材コンテンツのメタ情報を取得できる．

分散配置 コンテンツオブジェクトはインターネット上に分散配置される．分散配置することを可能にすることで，コンテンツプロバイダは提供するコンテンツをコピーすることなく一元管理することが可能になり，コンテンツをつねに最新の状態で提供することが可能になる．

サーバ側編集と編集操作のリスト 従来では，コンテンツクリエータが編集ツールを用いて素材コンテンツを編集していたが，MCOM では素材となるコンテンツオブジェクトが，クライアントの要求（編集操作のリスト）をもとに編集する．このサーバ側編集の特徴により，コンテンツオブジェクトは自らの規範に従ってコンテンツデータを編集することが可能になる．また，様々な編集操作を組み合わせることによりクライアントのニーズに適したコンテンツを提供することが可能になる．

実行時編集 MCOM では素材の編集作業をコンテンツ再生時（実行時）に行う．そして，編集されたコンテンツを収集してから再生を開始する．この特徴により，クライアントは素材として利用する際に編集が必要なコンテンツの場合においても，人の手を介することなく最新状態を反映したうえで利用することが可能になる．

図 1 の例を用いて，上述した MCOM の特徴を具体的に述べる．ここでは，インターネット上にコンテン

ツオブジェクト pc1，pc2，pc3，pc4，cc1，cc2 がそれぞれ分散配置されていると想定する．また，コンテンツ pc1 とコンテンツ pc2 を素材コンテンツとしてコンテンツ cc1 が作成され，コンテンツ pc3 とコンテンツ cc1 とコンテンツ pc4 を素材としてコンテンツ cc2 が作成されており，これらのコンテンツ間で階層構造を形成している．この階層構造の情報をもとに，コンテンツ cc2 を再生するとき，コンテンツ cc2 は素材として利用しているコンテンツ pc3，cc1，pc4 にそれぞれデータを要求する．cc2 からデータを渡すように要求された cc1 はさらに自身が素材として利用している pc1 と pc2 に対してデータを要求する．その際，pc1 は，サイズ変更の編集操作を行ったうえでデータを渡すように要求されているため，それらの処理を行ったうえで cc1 にデータを渡す．また，pc2 はサイズ変更と回転の編集操作を行ったうえで cc1 にデータを渡している．最終的に cc2 は各素材コンテンツを，必要な形に編集された状態で受け取り，再生する．

3.2 コンテンツ記述方式の選択

2.2 節で述べたように，コンテンツ記述方式としてすでに SMIL，ZyX などが存在するが，これらのモデルではクライアントツールがその記述内容を理解し，実行することを前提としているため，以下の問題が避けられない．

i) 記述した内容どおりに正しくコンテンツが扱われるかどうかがクライアントツールに依存してしまう．

ii) 記述した内容のすべてをクライアントツールが理解，実行できなければならない．

i) の問題は保護可能性の要求を解決することを困難にする．たとえば，編集の方法や制限などを記述した場合にも，その記述に正しく従うか否かがクライアントツールに依存してしまう．また，ii) の問題は機能の拡張にともないクライアントツールの機能を追加または変更しなければいけないという問題を引き起こす．これらの問題を避けるために，著者らはすでに広く利用されているコンテンツ記述方式であるデータ記述方式を用いず，オブジェクト指向アプローチを用いたコンテンツ記述方式を考案した．オブジェクト指向の特徴の 1 つであるカプセル化（情報隠蔽）[11] を利用するためである．MCOM では，サーバオブジェクトが，その実装を隠蔽し，自身が提供するサービスのみをクライアントオブジェクトに提示し，実際の処理の実行は自身が受け持つという，カプセル化の仕組みをコンテンツに導入することにより前述の問題 i)，ii) を解決する．

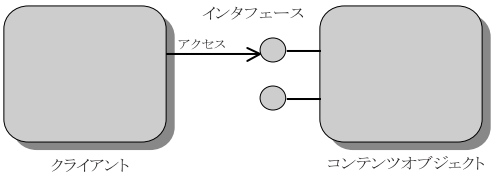

図 2 インタフェースを通じたコンテンツオブジェクトへのアクセス
Fig. 2 Access through an interface.

3.3 MCOM の基本設計

3.1 節で述べたように，MCOM では，従来，データとして取り扱われていたコンテンツを，メタ情報，コンテンツ，および振舞いを備えたコンテンツオブジェクトとして再定義する．それにともない，既存の MPEG ファイルや JPEG ファイルなどのコンテンツをコンテンツオブジェクトと区別するために，コンテンツデータと呼ぶことにする．コンテンツオブジェクトは，その実装を隠蔽し，インタフェースを通じてサービスを提供する．ここでインタフェースはオブジェクト指向アプローチで使われるインタフェース[12] を指すものとする．**図 2** にコンテンツオブジェクトにクライアントがアクセスする様子を示す．クライアントはインタフェースにアクセスし，インタフェースを通じてのみ内部データにアクセスすることができる．

3.4 階層構造

再利用可能性の要求条件を満たすために，MCOM に Primitive Content と Content Container の 2 つのオブジェクトを導入する．以下この 2 つのオブジェクトについて述べる．

3.4.1 Primitive Content オブジェクト

Primitive Content オブジェクトは MCOM における最も基本的な要素であり，内部にメタ情報とコンテンツデータ（たとえば，JPEG，GIF，MPEG など）から生成したオブジェクト（以下，コンテンツデータオブジェクトと略）を保持し，それを編集操作するためのメソッドを実装する．Primitive Content オブジェクトは自分自身が提供するコンテンツの種類に適した後述するコンテンツタイプインタフェース（たとえば，Image インタフェース，Sound インタフェース，Movie インタフェースなど）を持ち，その中で定義されたメソッドを実装する．**図 3** にそれぞれ適切なコンテンツタイプインタフェースを実装した Primitive Content オブジェクトを示す．ここで，Primitive Content pca は JPEG から生成されたコンテンツデータオブジェクトを保持し，Image インタフェースを実装している．Primitive Content pcb は GIF から生成されたコンテンツデータオブジェクトを保持し，Image インタ

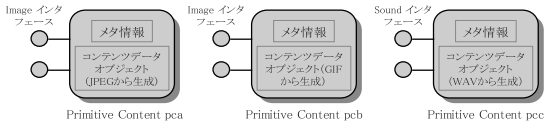

図 3 それぞれ適切なインタフェースを実装した Primitive Content オブジェクト
Fig. 3 Primitive Content Objects implemented suitable interfaces.

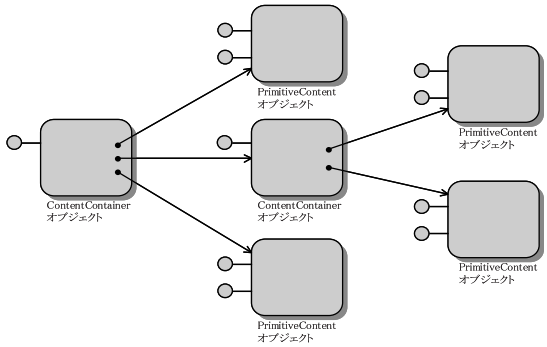

図 4 階層構造を形成するコンテンツオブジェクト
Fig. 4 Hierarchical structure of content objects.

フェースを実装している．また，Primitive Content pcc は WAV ファイルから生成されたコンテンツデータオブジェクトを保持し，Sound インタフェースを実装している．

3.4.2 Content Container オブジェクト

Content Container オブジェクトはメタ情報と，素材として利用するコンテンツオブジェクト（Primitive Content オブジェクトもしくは Content Container オブジェクト）への参照を 1 つ以上保持する．この Content Container オブジェクトの特徴によりコンテンツオブジェクトは**図 4** に示すような階層構造を形成することができ，既存のコンテンツオブジェクトを素材として再利用することができるようになる．

3.5 インタフェース

本節では，コンテンツオブジェクトが実装する 2 種類のインタフェースであるコンテンツタイプインタフェースとコンテンツインタフェースについて述べる．

3.5.1 コンテンツタイプインタフェース

コンテンツタイプインタフェースはコンテンツオブジェクトが提供するコンテンツの種類（たとえば，音声，画像，動画など）を表し，そのコンテンツの種類に適したサービスを提供するためのインタフェースである．このインタフェースを，後述する別のインタフェース（コンテンツインタフェース）と区別してコンテンツタイプインタフェースと呼ぶ．

静的コンテンツタイプインタフェース 3.4.1 項で述べたように，Primitive Content オブジェクトは自身が提供するコンテンツの種類に適した後述するコンテンツタイプインタフェースを提供する．この Primitive Content オブジェクトが提供するコンテンツタイプインタフェースはコンテンツオブジェクト生成時に決定され変化しないため，後述する動的コンテンツタイプインタフェースと区別して，静的コンテンツタイプインタフェースと呼ぶ．**図 5** の例では，Image インタフェースに translate メソッド，rotate メソッドおよび scale メソッドが定義されており，translate メソッドを通じて内部のコンテンツデータオブジェクトを編集している．

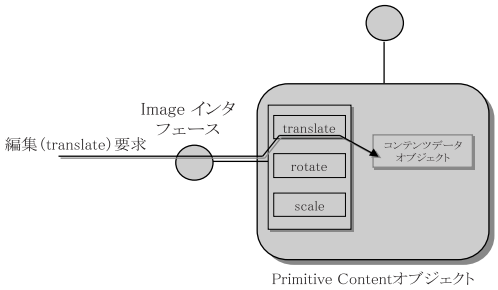

図 5 静的コンテンツインタフェース
Fig. 5 Static content interface.

動的コンテンツタイプインタフェース Primitive Content オブジェクトのコンテンツタイプインタフェースが静的に決定されるのに対し，Content Container オブジェクトのコンテンツタイプインタフェースは静的に決定されない．Content Container オブジェクトのコンテンツタイプインタフェースは Content Container オブジェクトが素材として利用するコンテンツオブジェクトが提供するコンテンツタイプインタフェースから導出される．これを Primitive Content オブジェクトが提供する静的コンテンツタイプインタフェースと区別して動的コンテンツタイプインタフェースと呼ぶ．**図 6** に動的コンテンツタイプインタフェースの例を示す．ここで，動的コンテンツタイプインタフェースをコンテンツコンテナに接続された破線のインタフェースとして示す．この例では Content Container cc1 は動的コンテンツインタフェースとして Image インタフェースを持ち，Content Container cc2 は動的インタフェースとして Sound，Image，Movie インタフェースを持っている．

3.5.2 コンテンツインタフェース

3.4.1 項で MCOM の基本要素である Primitive Content オブジェクトについて，また，3.4.2 項で，コンテンツオブジェクトを素材として利用するための要

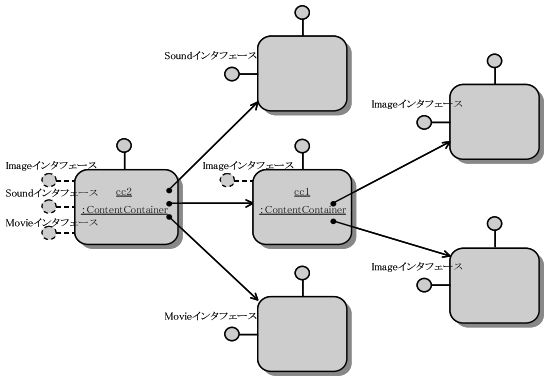

図 6 動的コンテンツインタフェース
Fig. 6 Dynamic content interface.

素として Content Container オブジェクトについて概説した．本項ではこの 2 種類のコンテンツオブジェクトが共通して持つコンテンツインタフェースについて述べる．コンテンツオブジェクトにアクセスするすべてのクライアントはこのコンテンツインタフェースにアクセスし，直接コンテンツタイプインタフェースにはアクセスしない．この仕組みにより，クライアントはコンテンツタイプインタフェースが静的なものか動的に導出されたものか区別する必要がなくなる．結果的に，クライアント側からは利用するコンテンツオブジェクトが Primitive Content オブジェクトなのか，Content Container オブジェクトなのか区別することなしに利用することが可能になる．コンテンツインタフェースが実装するメソッドのうち，特に重要なものとして以下のものがある．

- メタ情報を取得するメソッド
- コンテンツタイプインタフェースを取得するメソッド
- コンテンツタイプインタフェース内に定義されたメソッドを代理実行するメソッド
- コンテンツ（コンテンツデータオブジェクト）取得メソッド

図 7 にクライアントからコンテンツタイプインタフェースを取得する様子を示す．図 7 のコンテンツオブジェクトは動的コンテンツタイプインタフェースとして Image インタフェースと Sound インタフェースと Movie インタフェースを提供している．クライアントはこれらのコンテンツタイプインタフェースを静的コンテンツタイプインタフェースか動的コンテンツタイプインタフェースかを意識することなく取得している．

また，**図 8** にコンテンツインタフェースを通じて，コンテンツタイプインタフェースである Image インタフェースの中で定義された rotate メソッドを代理実行する例を示す．

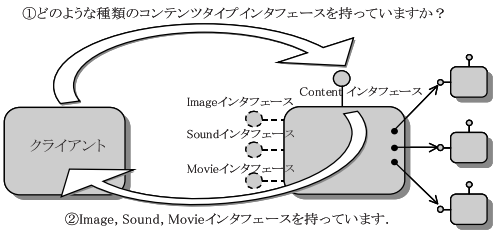

図 7 コンテンツタイプインタフェースの取得
Fig. 7 Query for content type interfaces.

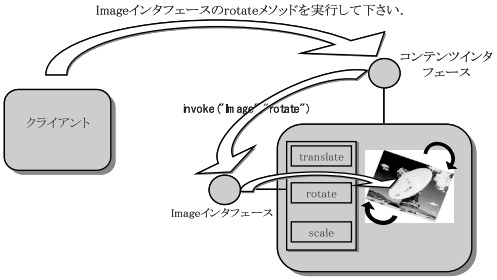

図 8 コンテンツタイプインタフェースで定義されるメソッドの代理実行の例
Fig. 8 An example of invoking a method defined in a content type interface.

3.6 編集操作の転送

図 8 において，コンテンツオブジェクトが PrimitiveContent オブジェクトである場合の例として，コンテンツタイプインタフェース（Image インタフェース）の利用方法をあげた．しかしながら，コンテンツオブジェクトが ContentContainer オブジェクトの場合，その内部には，素材となるコンテンツオブジェクトの参照を含んでいるだけであり，実際に編集すべきコンテンツデータオブジェクトは含んでいない．そのため，ContentContainer オブジェクトは，素材として利用しているコンテンツオブジェクトに編集要求を転送する．この要求の転送は階層構造のリーフ要素にあたる PrimitiveContent オブジェクトに渡るまで再帰的に行われる．

3.7 実行時編集とサーバ側シナリオ

コンテンツオブジェクトは再生時（実行時）に素材コンテンツを編集する．そしてこれらのコンテンツを収集した後に再生を行う．この特徴により，つねに最新の素材状態を反映することが可能になり，更新可能性を満たすことができる．実行時編集を実現するため，ContentContaienr オブジェクトには素材を編集するための情報を編集操作（メソッド）のリストとして格納する．一般に素材を利用するための情報をシナリオ

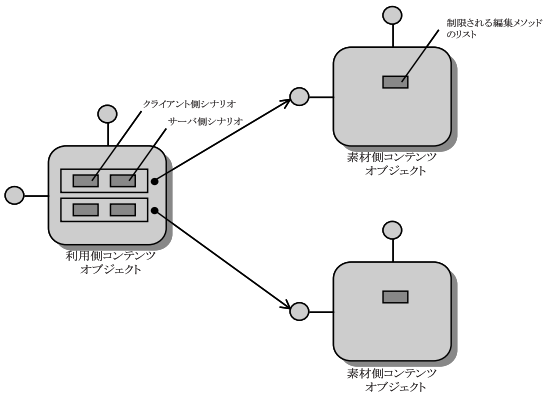

図 9 編集操作の制限
Fig. 9 Restrictions of editing operations.

というが，この編集操作のリストを，再生位置や再生時刻などのクライアント側で利用するシナリオと区別して，サーバ側シナリオと呼ぶ．

3.8 編集操作の制限

著作権保護の観点から，編集操作を制限したい場合がある．たとえば，芸術作品の場合は，グレースケールへの変換や解像度の変換を許可しない，などが考えられる．コンテンツオブジェクトは提供するコンテンツタイプインタフェースの中で定義されるメソッドのうち，実行を許可しないメソッドをリストとして保持し，編集メソッド実行時にその制限リストをチェックすることで実行を制限する．また，素材コンテンツを利用しているコンテンツの場合，階層構造を再帰的にチェックし，利用している素材コンテンツが制限をかけているメソッドの実行も制限する．**図 9** に利用側コンテンツオブジェクトがサーバ側シナリオを保持し，素材側コンテンツオブジェクトが制限される編集メソッドのリストを保持する様子を示す．

3.9 コンテンツの取得

クライアントはコンテンツを取得するために，コンテンツオブジェクトのコンテンツ取得メソッドを呼び出すだけでよい．コンテンツを要求されたコンテンツオブジェクトは自身が利用する素材コンテンツオブジェクトに対し，コンテンツを要求する（編集操作の転送をする）．クライアントは受け取った編集済みのコンテンツオブジェクトのリストを用いて再生を行う．**図 10** にクライアントがコンテンツオブジェクトから編集済みのコンテンツデータオブジェクトのリストを取得する様子を示す．

3.10 構　　造

本節では，MCOM の構造について詳細に述べる．階層構造を実現するために，**図 11** に示すように，PrimitiveContent クラス，ContentContainer クラスおよび Content クラスの関係を Composite パターン[13] を用いて設計した．また，PrimitveContent クラスは抽象クラスであり，コンテンツクリエータは PrimitiveContent クラスを継承したサブクラス（以下，ConcretePrimitiveContent クラス）を作成することにより，オブジェクトの振舞いを変更することができる．

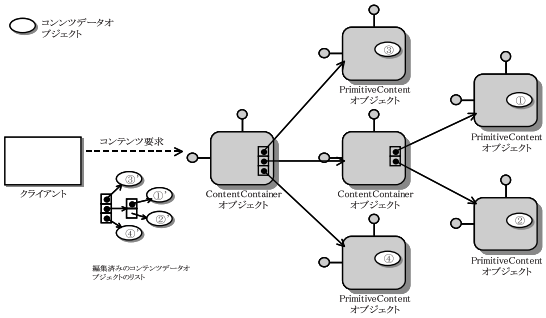

図 10 編集済みコンテンツデータオブジェクトのリスト
Fig. 10 A list of edited content data objects.

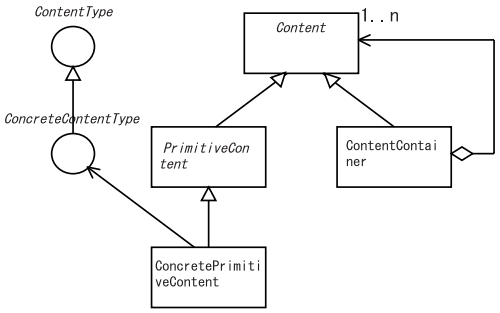

図 11 MCOM の構造
Fig. 11 Structure of MCOM.

また，ConcretePrimitiveContent クラスは ContentType インタフェースを継承したインタフェース（以下，ConcreteContentType インタフェース）を実装する．ここで，ContentType インタフェースは ConcreteContentType インタフェースがコンテンツタイプインタフェースであることを識別するためのインタフェースである．

図 12 に典型的な MCOM の構造を示す．ConcreteContentType インタフェースとして，Image インタフェース，Sound インタフェース，Movie インタフェースが設計されている．また，ConcretePrimitiveContent クラスとして，GIFImage クラス，JPEGImage クラス，WAVSound クラス，MPEGMovie クラスが設計されている．ここで，GIFImage と JPEGImage は画像コンテンツのため，Image インタフェースを実装し，WAVSound は音声コンテンツのため Sound イ

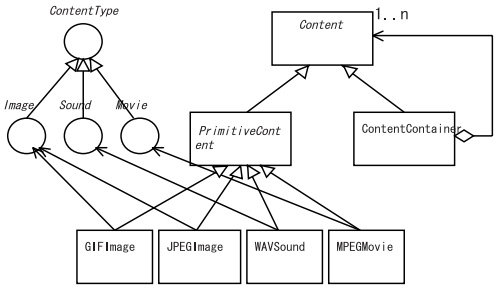

図 12 典型的な MCOM の構造
Fig. 12 Typical structure of MCOM.

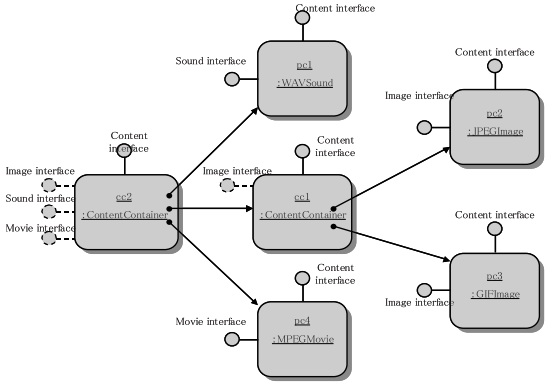

図 13 典型的なコンテンツオブジェクトの階層構造
Fig. 13 Typical hierarchical structure of content objects.

ンタフェースを実装し，MPEGMovie インタフェースは Movie インタフェースを実装している．

図 12 の構造を用いた，典型的なコンテンツオブジェクトの階層構造を**図 13** に示す．

4. Java を用いた MCOM の実装

本章では Java[14] を用いた MCOM の実装である MCOM フレームワークについて述べる．

4.1 MCOM フレームワークの構造

著者らは 3.10 節で示した MCOM の構造を用いて，MCOM フレームワークを設計および開発した．**図 14** に MCOM フレームワークの構造を示す．ここで，コンテンツオブジェクトを分散配置可能にするための機能である Java の RMI[15] を利用するために，抽象クラス `Content` は，`java.rmi.server.UnicastRemoteObject` を継承し，`java.rmi.Remote` インタフェースを継承した `IContent` インタフェースを実装する．また，コンテンツタイプインタフェースとして音声，画像，動画のためのコンテンツタイプインタフェースとして，それぞれ `ISound`，`IImage`，`IMovie` を設計した．また，`ISound` を実装した音声コンテンツオブジェクトを生

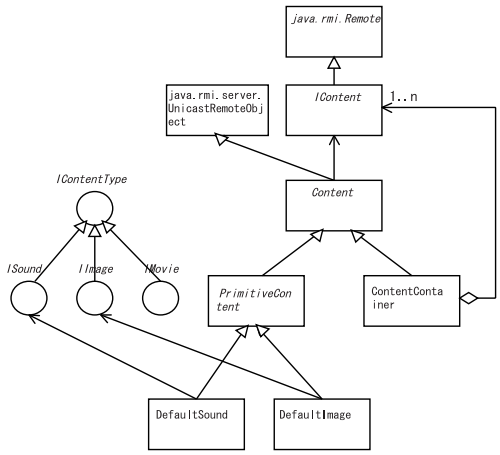

図 14 Java による MCOM の実装
Fig. 14 An implementation of MCOM using Java.

成するためのクラスである`DefaultSound` クラス，また，`IImage` を実装した画像コンテンツオブジェクトを生成するためのクラスである `DefaultImage` クラスを設計した．

4.2 IContent インタフェース

著者らはコンテンツインタフェースを MCOM フレームワークに実装する際に `java.rmi.Remote` インタフェースを継承した `IContent` インタフェースとして実装した．`java.rmi.Remote` インタフェースを継承することによりコンテンツオブジェクトはインターネット上で `IContent` インタフェースを公開することができるようになる．**図 15** に `IContent` インタフェースの詳細を示す．以下，図中で各メソッドのコメントとして与えた括弧付き番号をもとに説明する．

(1) のメソッドはメタ情報取得のためのメソッドである．クライアントはこのメソッドを通じてメタ情報ファイルを取得する．MCOM ではメタ情報記述ファイルのフォーマットを cIDf[4] あるいは XrML[3] のような XML による記述方式を使用することを想定している[16]．

(2) のメソッドはコンテンツオブジェクトが提供するコンテンツタイプインタフェースを取得するメソッドである．取得するコンテンツタイプインタフェースが静的コンテンツタイプインタフェースか動的コンテンツタイプインタフェースかを区別する必要はない．

(3)，(4)，(5)，(6) は編集制限情報を取得するメソッドである．

(7)，(8) はコンテンツオブジェクトからコンテンツデータオブジェクトをリストとして取得するメソッドである．特に (8) は引数としてサーバ側シナリオ（編集操作のリスト）を渡すことで，編集操作適用後のコ

```
public interface IContent extends Remote{
 public MetadataFile getMetadata()
    throws RemoteException;//(1)
 public Class[] getContentTypeInterfaces()
    throws RemoteException;//(2)
 public boolean hasRestriction(
    String methodName)
    throws RemoteException;//(3)
 public String[] getRestrictions()
    throws RemoteException;//(4)
 public boolean hasShallowRestriction(
    String methodName)
    throws RemoteException;//(5)
 public String[] getShallowRestrictions()
    throws RemoteException;//(6)
 public Object getContent()
    throws RemoteException;//(7)
 public Object getContent(
    ServerSideScenario sss)
    throws RemoteException;//(8)
 public void addChild(IContent child)
    throws RemoteException;//(9)
 public void addChild(int index,
    IContent child)
    throws RemoteException;//(10)
 public void removeChild(int index)
    throws RemoteException;//(11)
 public void removeChild(IContent child)
    throws RemoteException;//(12)
 public IContent getChild(int index)
    throws RemoteException;//(13)
 public int getIndexOfChild(IContent child)
    throws RemoteException;//(14)
 public int getChildCount()
    throws RemoteException;//(15)
 public boolean isLeaf()
    throws RemoteException;//(16)
}
```

図 15 IContent インタフェースの詳細
Fig. 15 Detail of IContent interface.

ンテンツデータオブジェクトを取得することができる．

(9)〜(16) は階層構造を形成するオブジェクトを操作する一般的なメソッド群[13),17)] である．

4.3 IImage インタフェース

本節ではコンテンツタイプインタフェースの 1 つである IImage インタフェースについて説明する．IImage インタフェースは画像コンテンツオブジェクトが提供するサービスを定義している．**図 16** に IImage インタフェースの詳細を示す．IImage では，代表的な画像の編集操作として，translate メソッド，rotate メソッド，scale メソッド，adapt メソッドを定義している．ここで，コンテンツをキーワードを用いてユーザニーズに最適化するメソッドである adapt メソッドも編集操作の 1 つとしてとらえ，IImage インタフェースで定義されるメソッドの中に加えた．また，他のメソッドと同様に adapt メソッドの実装も IImage インタフェースを実装するサブクラスで行う．

```
public interface IImage extends IContentType{
 void translate(double tx, double ty);
 void rotate(double angle);
 void rotate(double th, double x, double y);
 void scale(double sx, double sy);
 void adapt(String keyword);
}
```

図 16 IImage インタフェースの詳細
Fig. 16 Detail of IImage interface.

```
public class DefaultImage implements IImage{
 /** 画像を登録しておく */
 private HashMap imageRegistry_ =
      new HashMap();
 /** 画像の登録 */
 public void registImageWithKeyword(
    String keyword, String imageFilePath){
     省略
 }
 /** キーワードを用いたユーザ興味への最適化 */
 public void adapt(String keyword){
     省略
     ImageIcon suitableImage =
      (ImageIcon)imageRegistry_.get(keyword);
     省略
 }
     省略
}
```

図 17 DefaultImage クラスの詳細
Fig. 17 Detail of DefaultImage class.

4.4 DefaultImage クラス

本節では PrimitiveContent クラスを継承した ConcretePrimitiveContent クラスの 1 つである DefaultImage クラスについて説明する．DefaultImage クラスはコンテンツタイプインタフェースとして IImage インタフェースを実装する画像コンテンツのためのクラスである．**図 17** に DefaultImage クラスの詳細を示す．DefaultImage クラスは JPEG または BMP ファイルを javax.swing.ImageIcon クラスを利用してコンテンツデータオブジェクト化している．また，それを編集するメソッドを実装している．

また，adapt メソッドの実装としてコンテンツデータオブジェクトを複数格納しておき，キーワードに合わせて編集するコンテンツデータオブジェクトを入れ替える機能を実装している．

また，DefaultImage は以下の手順で編集を行う．

(1) サーバ側シナリオ（クライアントから渡される編集操作のリスト）に adapt メソッドがあれば，引数のキーワードを用いて，ハッシュテー

表 2 コンテンツ一覧
Table 2 List of contents.

名称	利用クラス	コンテンツ種別	説明	備考	利用している素材コンテンツ	サーバ側シナリオ
pc1	DefaultImage	画像	コンピュータ	回転（rotate）操作を許可しない設定	なし	
pc2	DefaultImage	画像	パラボラアンテナ		なし	
pc3	DefaultImage	画像	背景	adapt メソッドのため，以下のようにキーワードおよび画像を登録した． デフォルト：ビリヤードの画像．“kyoto”：金閣寺の画像．“sea”：海の画像．	なし	
pc4	DefaultSound	音声	BGM		なし	
cc1	ContentContainer	画像	画面付コンピュータ		pc1	size メソッド translate メソッド
					pc2	size メソッド rotate メソッド translate メソッド
cc2	ContentContainer	画像・音声	コンピュータの広告		pc3	adapt メソッド size メソッド
					cc1	size メソッド
					pc4	なし

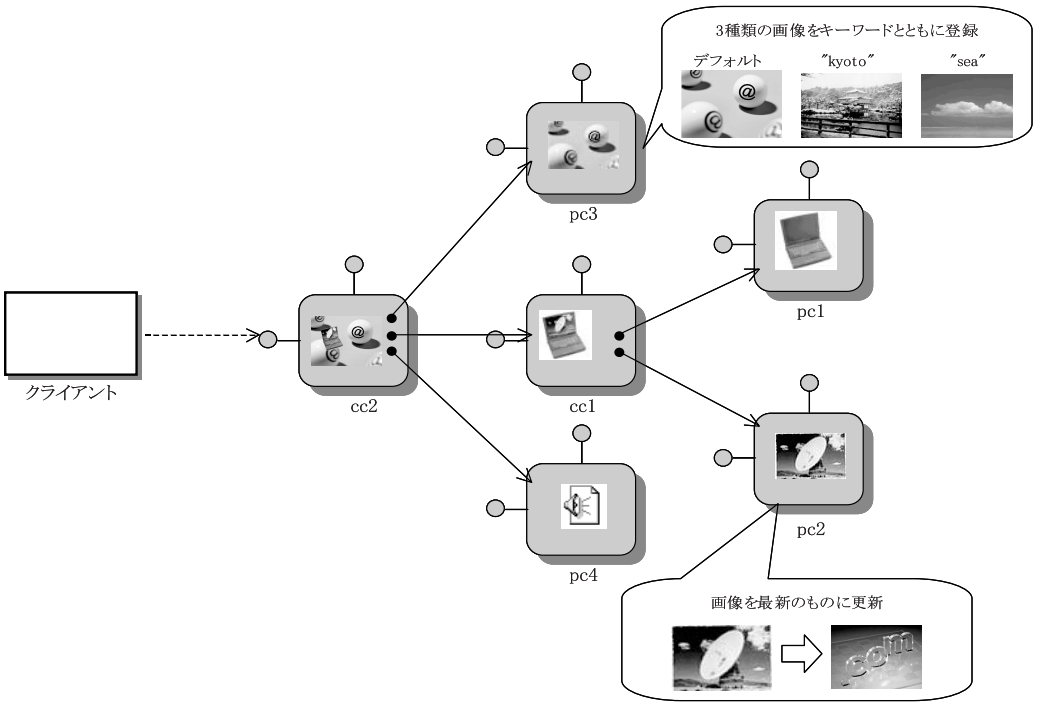

図 18 コンテンツの関係
Fig. 18 Relation of contents.

ブルに登録されたコンテンツデータオブジェクトを検索し，自身のコンテンツデータオブジェクトを置き換える．

(2) `adapt` メソッド以外の編集メソッド（移動，回転，サイズ変更）を順次実行する．

5. MCOM の評価

この章では具体的なコンテンツを作成し，2.1 節で述べた要求項目を MCOM が満たすことを示す．作成したコンテンツ一覧を**表 2** に示す．また，表 2 のコンテ

図 19 cc2 が生成する画像
Fig. 19 An image created by cc2.

図 20 pc3 にキーワードを設定した場合に cc2 が生成する画像
Fig. 20 An image created by cc2 using pc3 with keyword.

ンツ間の関係を**図 18** に示す．コンテンツオブジェクト pc1，pc2，pc3，pc4，cc1，cc2 を生成し，分散配置した．pc1，pc2，pc3，cc1 は画像コンテンツ，pc4 は音声コンテンツである．cc2 は画像と音声を含むコンテンツである．

再利用可能性 cc1 は pc1 および pc2 のコンテンツを素材として利用している．また，cc2 は pc3，cc1，pc4 のコンテンツを素材として利用している．これより，再利用可能性を満たすことを確認した．cc2 が生成した画像を**図 19** に示す．

適応可能性 cc1，cc2 は各素材を利用する際に編集を行い，自らのニーズに適した形でコンテンツを利用している．また，cc2 のコンテンツは pc3 を素材として指定する際にキーワードを用いて画像を最適化するよう設定することができる．**図 20** にキーワード “kyoto” を設定した際に cc2 が生成する画像を示す．

このほか，クライアントツールが cc2 にコンテンツを要求する際に，size メソッドを呼び出すことで画像を小さいサイズに変更可能なことを確認した．これらにより適応可能性を満たすことが確認できた．

更新可能性 pc2 の画像をパラボラアンテナの画像からドットコム（.COM）の画像に変更し，cc2 が生成する画像が人手を介することなく自動的に更新されることを確認した．**図 21** に最新状態を反映した画像を示す．

図 21 最新状態を反映した画像
Fig. 21 Updated image.

参照可能性 クライアントツールを用い，素材コンテンツの参照をたどることにより，各素材コンテンツが格納しているメタ情報ファイル（XML ファイル）を取得できることを確認した．

保護可能性 pc1 に回転（rotate）操作に対する制限を行い，利用側（cc1）から回転操作ができないことを確認した．

6. まとめと課題

本稿では，次世代のコンテンツモデルとして MCOM を提案し，また，その実装として MCOM フレームワークについて述べた．MCOM では，従来データとして扱われていたコンテンツを，オブジェクト指向に基づいたコンテンツオブジェクトとして再定義し，メタ情報，コンテンツデータおよび振舞いを備えるオブジェクトとして設計した．オブジェクトは自律的であり，保持するコンテンツデータを自らの規範に従い編集する．また，編集操作はコンテンツ再生時（実行時）に動的に行う．また，オブジェクトが階層構造を形成し，インターネット上に分散配置可能なように設計した．これらの特徴により，次世代コンテンツへの要求項目としてあげた，再利用可能性，更新可能性，適応可能性，参照可能性，保護可能性を満たすことが可能であることを示した．

今後の課題として，1) 通信するデータの暗号化，2) コンテンツデータへの透かしの挿入，3) データ更新時の通知，がある．現状では，インターネット，およびそこで利用されるオープンアーキテクチャをプラットフォームとして利用する場合，データがキャプチャ，またはコピーされることを完全に防止することができない．しかしながら，そのような状況下においても著

作権保護を支援する必要がある．通信するデータを暗号化することにより，ネットワーク上でのデータの改竄，キャプチャやコピーによる不正利用を防ぐ必要がある．また，透かしの挿入になどにより，クライアント側でのデータの不正な再利用を抑制する必要がある．また，コンテンツのデータが更新された際にクライアント（素材利用側）に通知することにより，意図しないコンテンツが再生されることを抑制することも検討する必要がある．

参 考 文 献

1) SMIL2.0. http://www.w3.org/TR/smil20/
2) Boll, S. and Klas, W.: ZyX—A Multimedia Document Model for Reuse and Adaptation of Multimedia Content, *IEEE Trans. Knowledge and Data Engineering*, Vol.13, No.3, pp.361–382 (2001).
3) XrML. http://contentguard.com/index.htm
4) コンテンツ ID フォーラム. http://www.cidf.org/
5) 谷口，森賀，久松，櫻井：マルチメディア情報ベースとその格納単位 Matryoshka，情報処理学会 DICOMO シンポジウム (1999).
6) RightsShell. http://www.digigacha.com/setumei/index.html
7) 谷口，阿部，塩野入：Java を用いた動画配信用著作権保護カプセル，情報処理学会研究報告 DPS (2000).
8) 木俵，田中，上原：版権管理のための Java による画像データカプセル化，情報処理学会研究報告 DBS (1997).
9) SMIL1.0. http://www.w3.org/TR/REC-smil/
10) XML. http://www.w3.org/XML/
11) Rumbauch, J., Blaha, M., Premerlani, W., Eddy, F. and Lorensen, W.: *Object-Oriented Modeling and Design*, Prentice-Hall (1991).
12) COM. http://www.microsoft.com/com/resources/comdocs.asp
13) Gamma, E., Helm, R., Johnson, R. and Vlissides, J.：オブジェクト指向における再利用のためのデザインパターン，ソフトバンク (1995).
14) Gosling, J., Joy, B., Steele, G. and Bracha, G.: *The Java Language Specification*, 2nd Edition, Addison-Wesley (2000).
15) RMI. ftp://ftp.java.sun.com/docs/j2se1.3/rmi-spec-1.3.pdf
16) 鈴木 正，池田哲夫，青木輝勝，安田 浩：Multimedia Content Object Model 及びその実装，情報処理学会研究報告 2001-EIP-13, pp.33–40 (2001).
17) JTree. http://java.sun.com/products/jdk/1.2/docs/api/javax/swing/tree/package-summary.ht

(平成 14 年 7 月 8 日受付)
(平成 14 年 12 月 3 日採録)

鈴木 正（学生会員）

平成 8 年東京理科大学大学院理工学研究科物理学専攻修士課程修了．現在東京大学大学院工学系研究科先端学際工学専攻博士課程在学中．次世代のマルチメディアコンテンツモデルに関する研究に従事．平成 14 年映像情報メディア学会鈴木記念賞受賞．電子情報通信学会，映像情報メディア学会各会員．

池田 哲夫（正会員）

昭和 54 年東京大学理学部情報科学科卒業．昭和 56 年東京大学大学院理学系研究科情報科学専攻修士課程修了．同年日本電信電話公社（現 NTT）電気通信研究所入所．NTT 在籍中，データベース管理システム，データベース応用システムの研究開発等に従事．平成 14 年 10 月より岩手県立大学教授．専門は，データベース工学，情報検索等．ACM，IEEE CS 各会員．工学博士（東京大学）．

青木 輝勝（正会員）

平成 5 年東京大学工学部電子工学科卒業．平成 10 年東京大学大学院博士課程修了．同年同大学先端科学技術研究センター助手，平成 14 年同大学同センター講師．テラビット IP ルータ，ギガビット LAN/MAN のアクセス制御方式，高性能マルチメディアプロトコル，次世代ビデオ会議システム，高能率画像符号化方式，ディジタルコンテンツ著作権保護技術等の研究に従事．平成 6 年財団法人電気電子情報通信振興財団猪瀬学術奨励賞受賞，平成 13 年本学会山下賞受賞，ほか 4 件の受賞．

安田　浩（正会員）

昭和 42 年東京大学工学部電子工学科卒業．昭和 47 年東京大学大学院博士課程修了，同年日本電信電話公社（現 NTT）入社．画像信号高能率符号化，画像信号処理，コンピュータ通信の研究に従事．昭和 53 年米国カリフォルニア州ジェット推進研究所客員研究員．NTT ヒューマンインターフェース研究所画像情報研究部長，NTT 法人営業本部システムサービス部開発部長，理事，NTT 情報通信研究所所長．平成 9 年東京大学教授，先端科学技術研究センター所属．現在，東京大学国際・産学共同研究センター副センター長．標準化関連では，国際的な活動として ISO/IEC・JTC1/SC29（AV&マルチメディア符号化）委員長．DAVIC（画像サービス関連業界標準化国際組織）プレジデント．昭和 47 年電子情報通信学会論文賞，平成 6 年度電子情報通信学会業績賞，平成 7 年度本学会情報規格調査会標準化貢献賞，平成 9 年度本学会情報規格調査会標準化功績賞，1995–1996 年米国 TV アカデミーエミー賞（技術開発部門）受賞等受賞多数．IEEE フェロー．