社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2007－DBS－143（27）
2007／7／2

コンテンツ演出のためのシナリオ記述言語

小宮山 哲† 国島 丈生†† 横田 一正††

† 岡山県立大学情報系工学研究科 〒719-1197 岡山県総社市窪木 111
†† 岡山県立大学情報工学部 〒719-1197 岡山県総社市窪木 111
E-mail: †{komiyama,kunishi,yokota}@c.oka-pu.ac.jp

あらまし 近年マルチメディアコンテンツが普及しているが、それらについて様々な意図や魅力を十分に伝えることは難しい。そこで我々は、学芸員による説明やテレビにおける美術館案内のような演出を生成するシステムを開発している。本システムではシナリオ記述言語を用いて意図する演出を記述することにより汎用性の高い Flash による演出を生成する。本稿ではシナリオ記述言語およびその生成実行システムについて述べる。
キーワード コンテンツ，演出，シナリオ，記述言語，Flash

Scenario Description Language for Contents Direction

Satoru KOMIYAMA†, Takeo KUNISHIMA††, and Kazumasa YOKOTA††

† Graduate School of Systems Engineering, Okayama Prefectural University Kuboki 111, Soja-shi, Okayama, 719-1197 Japan
†† Faculty of Computer Science and System Engineering, Okayama Prefectural University Kuboki 111, Soja-shi, Okayama, 719-1197 Japan
E-mail: †{komiyama,kunishi,yokota}@c.oka-pu.ac.jp

Abstract Recently, the multimedia contents are widespread. But it is difficult to convey the charm of contents enough only by sentences. Therefore, we have been developing a scenario description language for contents direction. And, we implemented the system to interpret and to execute a direction scenario on the Flash. In this paper,we propose a scenario description language and the system of interpreting and executing that one.
Key words Contents, Direction, Scenario, Description Language, Flash

1. 序 論

計算機の性能向上やインターネットの普及に伴い，大量の静止画や動画といったマルチメディアコンテンツが蓄積されるようになっている．これにより，美術館や博物館においても，所蔵品をディジタル化して Web 上で広く公開するようになってきている. [1] [2]

これらのコンテンツは，そのコンテンツに対する詳細な説明文と共に提示されていることが多い．コンテンツの閲覧者にとって，文章を読みながらコンテンツの閲覧を行うのは負担のかかる作業だと考えられる．説明文が長くなればなるほど負担も増してくる上に，コンテンツに対する興味や集中力が失われる可能性もある．また，コンテンツの提示は静的であるため，作成者は強調したい箇所などを文章のみで表現しなくてはならない．そのため，作成者の意図を閲覧者に対して正確に伝えることは難しいと考えられる．

そこで，個々のコンテンツに対して美術館の学芸員が説明するように，あるいはテレビでの美術館案内 [3] [4] のように，演出・説明を行える必要があると考えた．演出とは，説明文の発話をしながらコンテンツに対してズームをしたり，テロップを表示したりすることである．演出を行うためには，演出のシナリオを作成する必要がある．我々はこれまで，容易にシナリオを作成できる記述言語の研究・開発を行ってきた. [5] [6]

本論文ではその成果として，コンテンツ演出のためのシナリオ記述言語 CDL（Contents Direction Language）を提案する．さらに，CDL で記述されたシナリオを解釈実行するコンテンツ演出システムについて述べる．

2. コンテンツ演出

2.1 コンテンツ演出とは

コンテンツ演出とは，個々のコンテンツを美術館の学芸員が説明するような，あるいはテレビでの美術館案内のようなものを行うことである．例えば，絵画等のコンテンツについて，そのコンテンツの説明文の発話をしながら，コンテンツに対してズームをしていったり，あるいはテロップを表示させたりといったことが行われる．

また，コンテンツの作成者にとっても，コンテンツは静的であるため，強調したいあるいは注目してほしい箇所などを文章のみで表現しなくてはならない．そのため，作成者の意図やコンテンツの魅力を閲覧者に対して正確に伝えることが難しいと考えられる．

演出を行うことによってこの両者の問題は解決される．コンテンツの注目させたい部分にズームをしたり，ポインタを表示させて説明文の発話を行うことで，作成者はコンテンツの魅力を伝えることが容易となる．

関連研究としては，テレビ番組をまるまる 1 本記述できる言語 TVML (TV program Making language) [7] [8] や，XML に準拠したマークアップ言語 SMIL (Synchronized Maltimedia Integration Language) [9] が挙げられる．

2.2 演出機能の要件

効果的な演出を行うために必要だと考えられる演出機能を以下に挙げる．

- 音声によるナレーション
- カメラの動き (ズーム，パン，チルト，瞬時の視点変更)
- テロップの表示，画像の挿入
- ある部分を強調するためのポインタ，吹き出しの表示
- 2 つのコンテンツを表示させての見比べ
- コンテンツの入れ替え
- 動画ファイルの再生
- BGM の再生

これらの機能は互いに独立だが，演出のためにはこれらを並行に実行，または同期させることも求められる．さらに以下を要件として設定した．

- 簡単に記述できること

プログラミング言語などとは異なり，簡単な記述の仕様を覚えるだけでライトユーザでもシナリオを作成できる必要がある．

- 多様な演出を表現できること

テレビでの美術館案内のような演出を実現するためには，様々な演出が記述可能でなければならない．

- 正確な同期制御ができること

音声やカメラの動きによる視点の移動，テロップの表示等の各演出機能の同期を正確にコントロールできなければならない．

- 専用のブラウザやプレイヤーを必要としないこと

作成した演出シナリオは専用のプレイヤーを必要とせず，汎用ブラウザなどで気軽に鑑賞できる必要がある．

CDL ではこれらの要件を満たす記述ができることを目的としている．

3. 演出モデル

演出は実行単位及びその集合をノード，ノード間の順序関係を有向枝とした非巡回有向グラフ (DAG) としてモデル化できる．実行単位とは発話，ズーム，テロップの表示といった演出機能である．これらの実行単位を制御していくことによって演出が進められていく．ここで集合は並行実行を表すが，これは 3.2 節で説明する．図 1 に演出モデルを示す．

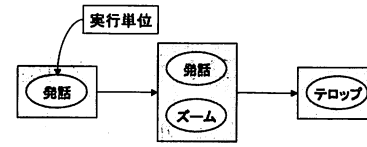

図 1 演出モデル

3.1 実行単位

実行単位は演出の最小単位となるものである．2.2 節で挙げた演出の要件を基に，実行単位としては表 1 に示した 14 種類を考えている．これらの実行単位を本論文ではアクションと呼ぶ．

表 1 演出の種類

アクション	アクションの効果
Move	瞬時の移動
View	パン，チルト
Zoom	ズーム
Speak	発話
Telop	テロップの表示
Balloon	吹き出しの表示
Pointer	注目させたい部分の強調
Comparison	コンテンツの比較，参照
Link	関連コンテンツへのリンク
Movie	動画の再生
Replace	画像の入れ替え
Insert	画像の挿入
BGM	BGM の再生
Wait	アクション間の待ち時間

以下では各アクションがどのような働きをするのかを説明する．

Move - 瞬時の視点の切り替えを行う．ある人物の全体像を見せて説明を行っていて，次に人物の口元の説明を行いたいときに，全体像が見えている視点から口元をズームアップした視点に切り替えるといったことに利用する．

View - 左右方向に視点を移動させるパンと，上下方向に移動させるチルトを行う．両者を組み合わせて斜めに移動させたりといったことも可能である．コンテンツをなめるように閲覧させたい場合などに利用する．

Zoom - コンテンツに対してズームインやズームアウトを行う．コンテンツの全体像から徐々にある部分にズームアップしていくことで細部を見せるといったことに利用する．

Speak - 発話を行う．コンテンツの説明を発話させるナレーション機能として利用する．

Telop - テロップの表示を行う．説明中に作品の題名や重要だと考えられる言葉などを表示させたい場合に利用する．

Balloon - 画面上に吹き出しとして説明を表示する．

Pointer - ポインタの表示を行う．説明中に強調したい箇所にポインタを表示させることによって注目させるといったことに利用する．

Comparison - 現在表示しているコンテンツに加えて，もう 1

つ別のコンテンツを並べて提示する．コンテンツの説明を行う際に，関連するコンテンツを並べることによって，より深い説明を行いやすくなる．

Link - 関連コンテンツへのリンクである．説明したコンテンツに加えて，このコンテンツも見てもらいたいといった場合などに利用する．

Movie - 動画の再生を行う．

Replace - 現在表示しているコンテンツと別のコンテンツとの入れ替えを行う．同じ作者の作品を順次入れ替えながら説明を行っていくといったことに利用する．

Insert - 画像の挿入を行う．説明中に地図などを補助的に画像として表示するといったことに利用する．

BGM - BGM の再生を行う．

Wait - 各アクションの待ち時間の設定を行う．説明が終わって次の説明に進む前に，閲覧者にしばらくコンテンツを眺めてもらう時間を設けたい場合などに利用する．

3.2 実行制御

実行制御とは，各アクションを実行するにあたっての制御手法である．実行制御として逐次実行，並行実行，同期制御の3つが挙げられる．図2にアクション実行の流れを示す．図2におけるアクション実行の流れであるが，まず最初に発話が行われる．発話が終了したらズームと発話が行われ，View により視点を移動させる．最後に発話とテロップの表示が行われるといった流れである．

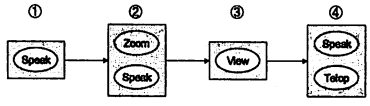

図2 アクション実行の流れ

逐次実行

図2において①が終了したら②を，②が終了したら③を，③が終了したら④を実行する．このようにあるアクションが終了したら次のアクションを開始と順次アクションを実行していくのが逐次実行である．

並行実行

ズームの後に発話，その後にテロップの表示と，アクション単体のみの実行ならば逐次実行のみで十分である．しかし，ズームをしながら発話をするといったように，複数のアクションを組み合わせて実行させたい場合がある．並行実行は，複数のアクションを同時に実行させる制御手法である．図2において②ではズームと発話が同時に実行されることを表しており，ズームをしながら発話が行われる．同様に④では発話中にテロップが表示される．

また，並列実行の単位が構造体であることも考えられる．これは例えば逐次実行とアクションを並行実行するものである．しかし，それらの同期が保証できないので単純な実行順としている。

同期制御

並行実行により，独立した複数のアクションを同時に実行させることができる．しかし，アクションの実行は同時に開始されるので，並行実行のみではアクション開始時の同期しかとることができないため，アクション終了時の同期をとることができない．また，アクションの開始時間をずらして実行することができない．これらを実現するためには，各アクションを実行するタイミングを細かく指定して，正確にコントロールできる必要がある．これが同期制御である．同期制御により，図3において②のようにズームと発話の終了時点をそろえることができる．あるいは，④のように発話が開始されてから5秒後にテロップを表示し，その5秒後に消去するといったことができる．

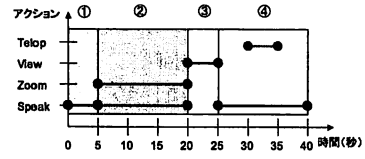

図3 時系列グラフによるアクション実行の流れ

4. コンテンツ演出記述言語 CDL

本章では，コンテンツのシナリオを記述する言語 CDL について述べる．CDL の位置づけを図4に示す．

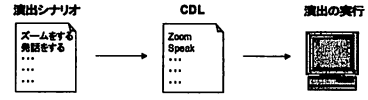

図4 CDL の位置づけ

4.1 CDL の概略

図4に示すように演出シナリオを CDL を用いて記述することで，コンピュータで解析することができ，演出を実行することができる．演出シナリオはいくつかの演出単位に分割することができる．演出単位は入れ子にすることが可能で他の演出単位から参照できる．そのために，各演出単位には演出 ID が必要となる．演出 ID は1つの演出シナリオ内で一意である．

まず “演出 ID(title = <タイトル>) = [ ]” を記述する．“[ ]” の中にアクションの列を記述する．

title 属性の属性値には演出のタイトルを記述する．title 属性は記述しなくてもよい．また，“演出 ID() = [ ]” は入れ子にすることができる．

アクションは 3.1 節で挙げた以下の 14 種類である．

また，マクロ定義およびインクルードを使用することができる．マクロは文字列の置換を定義する．“演出 ID() = [ ]” の外部で冒頭に記述しておくと，それ以降の該当文字列が置換される．インクルードは外部ファイルを指定することで，そのファイルの内容をそのまま挿入することができる．

4.2 実行制御

4.2.1 逐次実行および並行実行

3.2節で述べた実行制御のうち，逐次実行は“;”で，並行実行は“,”で各アクションを区切って記述することにより実現できる．

並行実行では複数のアクションを同時に実行することができるが，アクション開始時の同期のみしかとることができない．そこで，アクションの開始時間と継続時間を指定することでこの問題を解決する．これがアクションの同期制御である．start属性でアクションの開始時間を，duration属性でアクションの継続時間を指定する．

このようにstart属性とduration属性を用いることで，各アクションの実行を正確にコントロールすることができる．

4.2.2 コンテンツ比較時の制御

別コンテンツを提示させて比較を行うアクションとしてComparisonがある．コンテンツは通常1つだけが表示されているが，Comparison使用時にはコンテンツの提示は2つになる．そのため，どちらのコンテンツに対してズームを行うか，ポインタを表示するかといったことを指定できる枠組みが必要となる．これを解決するために，Comparisonのid属性に提示させるコンテンツのidを記述する．そして他の各アクションのactive属性にそのidを記述することによって，どちらのコンテンツに対してアクションを行うかを決定する．

4.3 インデックス

インデックスを記述しておくことによって，インデックスで指定した箇所からの演出を実行することが可能となる．また，どのような演出が行われるかを知ることもできる．インデックスはIndexを用いて記述する

4.4 再利用性の向上

Replaceを使用することで，現在提示しているコンテンツを別コンテンツへと切り替えられる．これにより関連のあるコンテンツを連続して説明することができる．

また，CDLでは“演出ID() = [ ]”を入れ子にすることができる．これにより，図5に示すように，モナリザに対する演出，聖ヨハネに対する演出，聖アンナに対する演出をはっきりと区別することができる．また，別コンテンツの演出を別ファイルに記述しておくことで，インクルードによって利用することができる．これにより，聖アンナの演出を単体で扱うことも可能となり再利用性の向上にもつながる．図6及び図7にインクルードの記述例を示す．

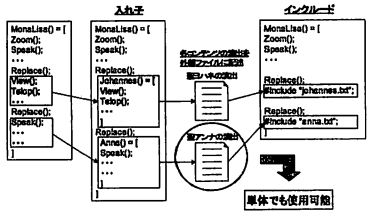

図5 インクルードによるシナリオの再利用

```
1 #define CENTER [320,240];
2
3 Sample(title = "サンプル") = [
4 BGM(condition = "start", src = "bgm.wav");
5 Zoom(coordinates = [0.2,0.5]
6 , magnification = 3, velocity = 2),
7 Speak(text = "レオナルド・ダ・ヴィンチは … ");
8 View(coordinates = CENTER, velocity = 2),
9 Speak(src = "setumei.txt");
10 Movie(src = "movie.avi");
11
12 #include "xyz.txt"
13
14 BGM(condition = "stop", src = "bgm.wav");
15 ]
```

図6 CDLによる演出シナリオの記述例

```
1 Pointer(type = "round", pointersize = "large",
2   coordinates = [235,90]),
3 Speak(src = "sample.mp3");
```

図7 インクルードファイルの内容

5. CDLを用いたコンテンツ演出システムの実現

5.1 システムの概要

コンテンツの演出は，演出シナリオを演出エンジンに読み込ませることによって実行することができる．演出エンジンの実装にはFlashのActionScriptを用いている．Flashを用いた理由は，普及率の高さと動作の軽さからである．

システムの実行の際には，CDLで記述された演出シナリオをコンパイルにより中間言語にする必要がある．ActionScriptでは，音声ファイルはmp3形式でしか読み込めないなどといった制約があるからである．中間言語については次節で詳細を述べる．図8にコンテンツ演出システムのモデルを示す．

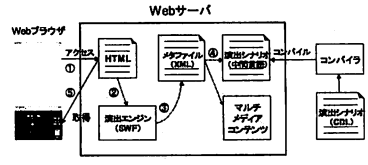

図8 システムモデル

演出実行までの流れをまとめると以下のようになる．

(1) HTMLファイルにアクセス

HTMLファイルへは，URIの末尾にメタファイル名を付与してWebブラウザによりアクセスする．メタファイル名を変更することにより，異なるコンテンツの演出を実行でき

る．演出エンジンが記述されている HTML ファイルが “direction.html” であり，メタファイルが “monariza.xml” であれば “direction.html?monariza” としてアクセスする．

（2） 演出エンジンを読み込む

1 で読み込んだ HTML ファイルに記述されている演出エンジンを読み込む．

（3） メタファイルを読み込む

演出エンジンによって，URI の末尾で指定したメタファイルが読み込まれる．メタファイルの形式は XML で，演出の対象となるコンテンツの URI と演出シナリオのファイルの URI を記述する．

direction 要素の子要素として img 要素と cdl 要素を記述する．img 要素の src 属性に演出の対象となるコンテンツの URI を，cdl 要素の src 属性に演出シナリオを記述したファイルの URI を記述する．

（4） 演出シナリオの読み込みとマルチメディアコンテンツの取得

3 で読み込んだメタファイルから演出シナリオを読み込み，必要なマルチメディアコンテンツを取得する．

（5） 表示

ブラウザに表示させる．

通信速度による問題とその解消方法

演出エンジンは記述された演出シナリオを “ ; ” ごとに区切ってインタプリタ形式で実行をしていく．アクションによって画像や音声を外部ファイルから読み込む場合があり，読み込みを行う際に通信速度によって遅延が生じる．そのため，発話が開始されていないのにズームが始まったりするなど，同期制御にずれが生じる．そこで，演出の過程で必要となる画像，音声などの外部ファイルを前もって全て読み込んでおくことによって問題を解決している．

5.2 中間言語への変換

5.2.1 中間言語とは

演出エンジンは Flash の ActionScript で作成されている．ActionScript では音声ファイルは mp3 形式で，動画ファイルは flv 形式でしか読み込めないといった制約がある．そのため，CDL で記述された演出シナリオをそのままでは実行できない場合がある．そこで，ActionScript で読み込める実行可能な形式へ変換をした演出シナリオを中間言語と呼んでいる．

5.2.2 中間言語の書式

CDL から中間言語へどのような変換が行われるかについて述べる．図 6 及び図 7 に CDL による演出シナリオの記述例と指定したインクルードファイルの内容を示した．これらを CDL から中間言語へ変換した形を図 9 に示す．

中間言語は CDL に対して以下のような変換が為される．

- マクロの処理

マクロの定義に従い該当文字列の置換を行う．

- インクルードの処理

インクルードファイルの内容を挿入する．

- 直接記述された発話内容から mp3 ファイルを生成

Speak アクションにおいて，直接記述された発話内容から mp3 ファイルの生成を行う．

- wav ファイルを mp3 ファイルへ変換

Speak あるいは BGM アクションにおいて wav ファイルがあれば mp3 ファイルへ変換する．

- 動画ファイルの形式を flv 形式へ変換

Movie アクションにおいてファイル形式が flv でなければ flv 形式へ変換する．

- 相対座標を絶対座標へ変換

座標の指定を相対座標で記述している場合，演出対象のコンテンツのサイズを基に絶対座標へ変換する．

- 属性値のみの記述へ変換

属性名を消去し属性値の記述のみに変換する．省略された属性に関しては “_” を記述する．

```
 1 Sample("サンプル") = [
 2 BGM("start", "bgm.mp3", _);
 3 Zoom([128,240], 3, 2, _, _, _),
 4 Speak("explain1.mp3", _, _, _);
 5 View([320,240], 2, _, _, _),
 6 Speak("setumei.mp3", _, _, _);
 7 Movie("movie.flv");
 8 Pointer("round", "large", [235,90], _, _, _),
 9 Speak("sample.mp3", _, _, _);
10 BGM("stop", "bgm.mp3", _);
11 ]
```

図 9 中間言語による演出シナリオの記述例

コンパイルは，変換を行う演出シナリオを読み込みコンパイルボタンを押すと実行される．相対座標から絶対座標への変換は，入力されたコンテンツのサイズを基に変換される．しかし，この方法では Replace アクションによるコンテンツの入れ替えに伴う演出対象の変更に対応できないため，コンテンツの URI の記述からコンテンツのサイズを取得するなどの方法を考える必要がある．

5.2.3 CDL コンパイラの実装

CDL から中間言語への変換コンパイラを Visual C++ 2005 を用いて作成した．コンパイラの使用画面を図 10 に示す．

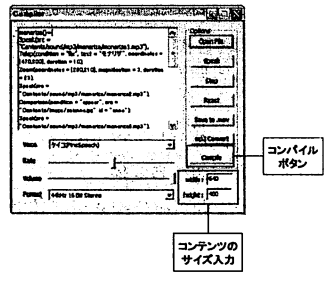

図 10 コンパイラ

5.3 ユーザインタフェース

演出を実行するためのインタフェースを図 11 に示す．

図 11　インタフェース

演出の再生ボタンを押すことで演出が実行される．停止ボタンを押すと演出が停止し初期状態に戻る．巻戻しボタンで巻戻し，早送りボタンで早送りができる．インデックスリストに表示されているインデックスを選択して選択再生ボタンを押すことで，インデックスを付与した箇所から再生することができる．

また，拡大，縮小ボタンにより，手動でコンテンツを拡大したり縮小したりすることができる．

5.4　システムの応用

システムの応用として，我々が開発している D-Cubis [11] [12]，PasQ [13] での利用，また，Internet Explorer などの Web ブラウザでの利用を考えている．

D-Cubis - 我々が開発したディジタルミュージアム提示システムである．ミュージアム作成者は，簡易的な 3D 空間を作成し，そこに様々なマルチメディアコンテンツを貼り付けていくことでミュージアムを作成する．閲覧者は Web ブラウザで閲覧することができ，快適にウォークスルーしながら展示してあるマルチメディアコンテンツを閲覧することができる．

PasQ - 我々が開発した複数のパノラマ画像を組み合わせて仮想空間を構築するシステムである．PasQ ではパノラマ画像に対してズーミングを行い，パノラマ画像を自動的に切り替えることで仮想空間のウォークスルーを表現する．

演出は独立に記述されているので，HTML ファイルへのリンクを設定することで様々なシステムから呼び出し可能である．D-Cubis や PasQ のように動的に仮想空間を構築するアプリケーションの場合でも，元のシステムを拡張することなく利用することができる．

6. 結　　論

本論文では，テレビでの美術館案内のような演出を容易に作成するためのシナリオ記述言語として CDL を提案した．そして，CDL で記述された演出シナリオを解釈実行するコンテンツ演出システムを実現し，効果的な演出を行うことができることを示した．

今後の課題や発展の方向性として以下のことが挙げられる．

- サーバ側での中間言語への変換

現在のところ CDL から中間言語への変換はクライアント側で手動で行い，その後サーバ上に配置しなければならない．その手間を軽減するため，サーバ側で自動変換を行える必要がある．

- CDL オーサリングツールの実現

CDL で演出シナリオを作成するにあたって，座標の指定のためにコンテンツのサイズを，同期制御のために発話の時間を把握しておかなければならない．そこで，演出シナリオの作成を補助するためのオーサリングツールを実現する必要がある．また，タイムラインを用いて演出の作成を視覚的に行うことも考えられる。

サーバ側で変換を行うことを考えると，Web アプリケーションを用いたオーサリングを実現するのが妥当だと考えられる。

- CDL の記述の再考察，演出機能の洗い出し

容易な記述性，柔軟性のある言語を目指して記述の再考察を進めていく．また，本論文で挙げた演出機能だけではまだまだ十分だとは言い難いため，さらに効果的な演出を行う演出機能を考えていく．

- 発話音声の多様化

現在発話のための音声は，男声と女声の 2 種類でしかできないので，さらに多くの種類の音声を扱える必要がある．しかし，コンパイラの音声合成に用いている Microsoft Speech SDK 5.1 等では 2 種類の日本語音声しか扱えないので，別の音声合成ソフトを利用することを検討しなければならない．

- キャラクタの必要性

TVML のようにキャラクタを登場させ説明を行わせることで，閲覧者の注目を引くことが期待できる．

- ユーザインタフェースの充実

各演出の実行時間などを基にして，タイムラインバーを作成することでユーザインタフェースの充実を目指す．

- 共有機能を用いた双方向システム

本システムをクライアントとして，多数のクライアント間で同期をとることにより双方向システムとすることが考えられる。例えば，問題を解く時間を予め定めておき，全員が同時に問題に取り組むことなどが考えられる．

文　　献

[1] 大原美術館 WEB 展示室, http://www.ohara.or.jp/200606/jp/1_web/menu.html

[2] 玉野市 3D ヴァーチャル海洋博物館, http://media2.city.tamano.okayama.jp/3d_kaihaku/

[3] 新日曜美術館, http://www.nhk.or.jp/

[4] 美の巨人たち, http://www.tv-tokyo.co.jp/kyojin/

[5] 野田英志，國島丈生，横田一正，“ディジタルミュージアムにおける自動案内のための機能と実現方法の提案”，夏のデータベースワークショップ DBWS2003，網走，June 16-18，2003.

[6] 辻圭一，國島丈生，横田一正，“ディジタルミュージアムにおけるコンテンツ演出のための記述言語の拡張”，電気・情報関連学会中国支部第 57 回連合大会，pp.261-262，2006.

[7] TV program Making Language. http://www.nhk.or.jp/strl/tvml/index.html

[8] 林正樹，“番組記述言語によるテレビ番組自動生成”，第 2 回知能情報メディアシンポジウム（1996）．

[9] Synchronized Multimedia Integration Language. http://www.w3.org/TR/SMIL2/

[10] 灘本明代，服部多栄子，近藤宏行，沢中郁夫，田中克己，“Web コンテンツの受動的視聴のための自動変換とスクリプト作成マークアップ言語”，情報処理学会論文誌 Vol42 No.SIG1(TOD8)，pp.103-116，2001.

[11] D-Cubis Official Site，http://alpha.c.oka-pu.ac.jp/D-Cubis/

[12] 江本守，石崎勝俊，大河内久貴，国島丈生，横田一正，“利用者指向ディジタルミュージアムの共有化とモジュール化”，日本データベース学会 Letters Vol.3 No.1，pp.137-140，2004.

[13] 池田隼，国島丈生，横田一正，“パノラマ画像を用いた仮想空間の構築”，日本データベース学会 Letters Vol.5 No.1，pp.97-100，2006.