

モバイル複合現実感システムにおける コンテンツ記述言語の設計と実装

橋本 崇 吉田 友祐 柴田 史久 木村 朝子 田村 秀行

立命館大学大学院理工学研究科
〒525-8577 滋賀県草津市野路東1丁目1-1
E-mail: hashi@mclab.ics.ritsumeai.ac.jp

あらまし 我々は、携帯電話、PDA、ウェアラブル・コンピュータ等、様々な種類の可搬型機器にMR機能を搭載できる共通のフレームワークの構築を目指している。提案フレームワークでは、クライアント・サーバ・アーキテクチャを採用し、クライアントとサーバで処理を分散させることで端末の処理能力の差異を吸収している。この際に、サーバ・クライアント間で情報をどのように送受信するかが重要となる。本稿では端末の種類に依存しない記述を可能にし、仮想オブジェクトの位置や形が時間によって変化する場合にも対応可能なコンテンツ記述言語SKiT-XML 2.0を設計し、実装したのでこれを報告する。

キーワード 複合現実感、モバイル端末、コンテンツ記述言語、XML

Design and Implementation of Contents Description Language for Mobile Mixed Reality System

Takashi HASHIMOTO Yusuke YOSHIDA

Fumihisa SHIBATA Asako KIMURA and Hideyuki TAMURA

Graduate School of Science and Engineering, Ritsumeikan University
1-1-1 Nojihigashi, Kusatsu, Shiga 525-8577, Japan
E-mail: hashi@mclab.ics.ritsumeai.ac.jp

Abstract We are now aiming at establishment of a general framework for mixed reality (MR) systems that can be installed on various types of mobile computers such as cellular phones, personal digital assistants (PDAs), and wearable computers. In our framework, the system consists of a server computer and some client terminals. Some functions are distributed into the server and the clients in order to absorb differences in performance of terminals. It is important how to transmit information between the server and the clients. In this paper, we propose SKiT-XML, a language for describing MR contents in XML. This language is independent of types of mobile devices. SKiT-XML supports the ability to change positions and orientations of virtual objects over time. MR content can be animated using SKiT-XML 2.0.

Keyword Mixed Reality, Mobile Device, Contents Description Language, XML

1. はじめに

計算機を利用した新たな情報提示技術として複合現実感 (Mixed Reality; MR) に関する研究が脚光を浴びつつある[1-3]。今後、この技術は屋内外の広い範囲において手軽に利用できるモバイル・ウェアラブル型のシステムへと発展することが予想される。

そこで我々の研究グループでは様々な可搬移動型端末の性能や種類を吸収可能な複合現実感システムのための共通フレームワークの構築を目指している。これまでの研究において、共通フレームワークを設計し、実際にそれを実現する基本アーキテクチャを複数種類

の可搬移動型機器の上に実装し実験を進めてきた[4]。本フレームワークは、サーバ・クライアント方式を採用し、クライアントの能力不足をサーバで補う機構を実現している。この機構では、クライアントとサーバ間においてMRのコンテンツに関わる情報を伝達するための共通の情報記述方式が必要となる。本研究では、このような情報伝達を目的としたコンテンツ記述言語SKiT-XML 2.0を設計し、その処理系を実装したので、本稿でそれを報告する。

2. 共通フレームワークの概要

我々は様々なモバイル機器に対応可能な複合現実

感システムの共通フレームワークの実現を目指している。ここで目標としているのは、アプリケーションに依存しないプラットフォーム層を開発すると同時に、想定する各種端末の性能や種類の差をも吸収可能な共通フレームワークを実現することである。本フレームワークではクライアント・サーバ・アーキテクチャを導入し、クライアントとサーバで処理を分散させることで端末の処理能力の差異を吸収する。本フレームワークのシステム・アーキテクチャを図 1 に示す。

本フレームワークでは端末をその処理能力に応じて、それぞれ軽量クライアント、中量クライアント、重量クライアントと呼ぶ 3 種類に分類する。

軽量クライアントでは、複合現実感を実現するために必要最低限の機能である画像取得機能と MR 情報提示機能のみをクライアント側に配置し、残りの処理はサーバが行う。すなわち、軽量クライアントでは、現実の光景を撮影し、その画像に合わせて MR 情報がレンダリングされて提示されるという、静止画によるサービスとなる。

中量クライアントは、軽量クライアントに自身の位置姿勢を検出するための機能及び MR 情報を生成する機能を追加したクライアントである。サーバから自身の周囲の局所的なコンテンツを受け取ることで、リアルタイムな情報提示が可能となるが、提示すべきコンテンツの取捨選択はサーバ側で行い、クライアントはそのコンテンツを位置姿勢に合わせてレンダリングするのみとなる。

重量クライアントは、MR を実現するための機能を全てクライアント側に配置した自己完結型のクライアントである。コンテンツ情報をサーバとの間で同期をとることによって、クライアントのコンテンツを常に最新の状態に保ち、クライアントが自ら提示するコンテンツを決定し提示する。

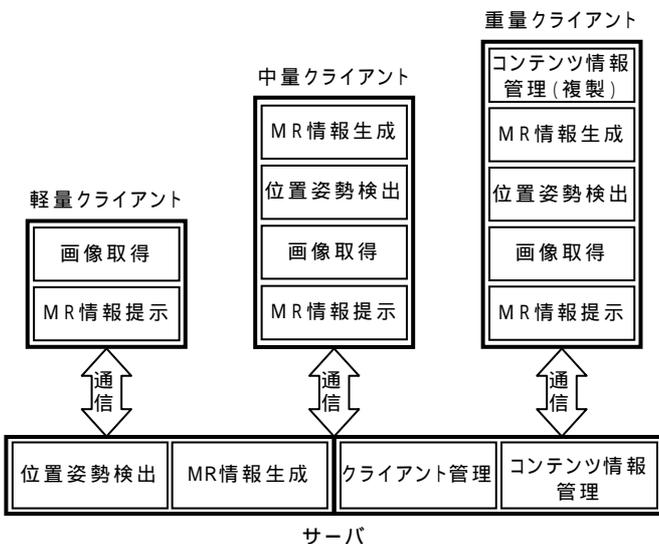


図 1 システム・アーキテクチャ

3. コンテンツ記述言語の設計

3.1. 設計方針

本フレームワークにおいて MR 情報として提示可能なコンテンツは、3 次元空間中に存在するものと端末の画面上に平面的に提示されるものの 2 つに分けて考える。

3 次元空間中に存在するコンテンツとしては、(1)VRML[5]やプリミティブ形状で表現した 3D モデル、(2)文字による注釈情報にあたるアノテーション、(3)3 次元空間中に配置される画像、の 3 種類を扱う。

一方、画面上に提示されるコンテンツとしては、(1)画面上に提示される画像、(2)画面上での注釈情報にあたるアノテーション、(3)決められた領域に文字として提示されるメッセージ、の 3 種類を扱う。

2 章で述べた各クライアントは、MR 情報を提示するためにサーバとの間で各種の情報を送受信する必要がある。これらは大きくクライアントからサーバに送られるリクエストとサーバからクライアントに送られるレスポンスに分けられる。クライアントの種類によってサーバに依頼する処理が異なるため送受信される情報も異なる。これらの情報をサーバ・クライアント間で送受信する際に用いるのが、本稿で提案する SKIT-XML 2.0 である。各クライアントとの間で送受信する情報のイメージを図 2 に示す。

複合現実感システムを構築する上で、システムに採用するモバイル端末と前述のクライアントの種類は、直接関係はない。例えば、ポイントとなる地点のみで MR 情報を確認すればいいようなナビゲーションやガイドのようなアプリケーションであれば、軽量クライアントでの実装が適切であろう。この場合、ユーザの使い勝手を考えると携帯電話か PDA での実現が現実的であり、具体的にどの機種を採用するかでかなりの幅ができ得る。一方、機器の組み立て作業支援などを想定すると、様々な角度からコンテンツを確認する必要性から、中量クライアントか重量クライアントでの実装が望ましい。これには、ウェアラブル PC のような端末の使い勝手がよいが、端末の重量の違いなどからいくつかの選択肢がある。したがって、サーバとクライアント間で送受信する情報をモバイル端末ごとに別々に記述すると、新たな端末への対応が難しくなる。そこで本研究では、共通の情報記述方式を利用して情報をやりとりすることとする。ここでは、モバイル端末に依存しない記述が可能である、様々な機種向けに処理系が公開されており移植容易性が向上する、という理由から、メタ言語 XML の仕様に基いてコンテンツ記述言語の設計を行うこととした。

3.2. コンテンツ記述言語の構造

コンテンツ記述言語の構造を図 3 及び図 4 に示す。

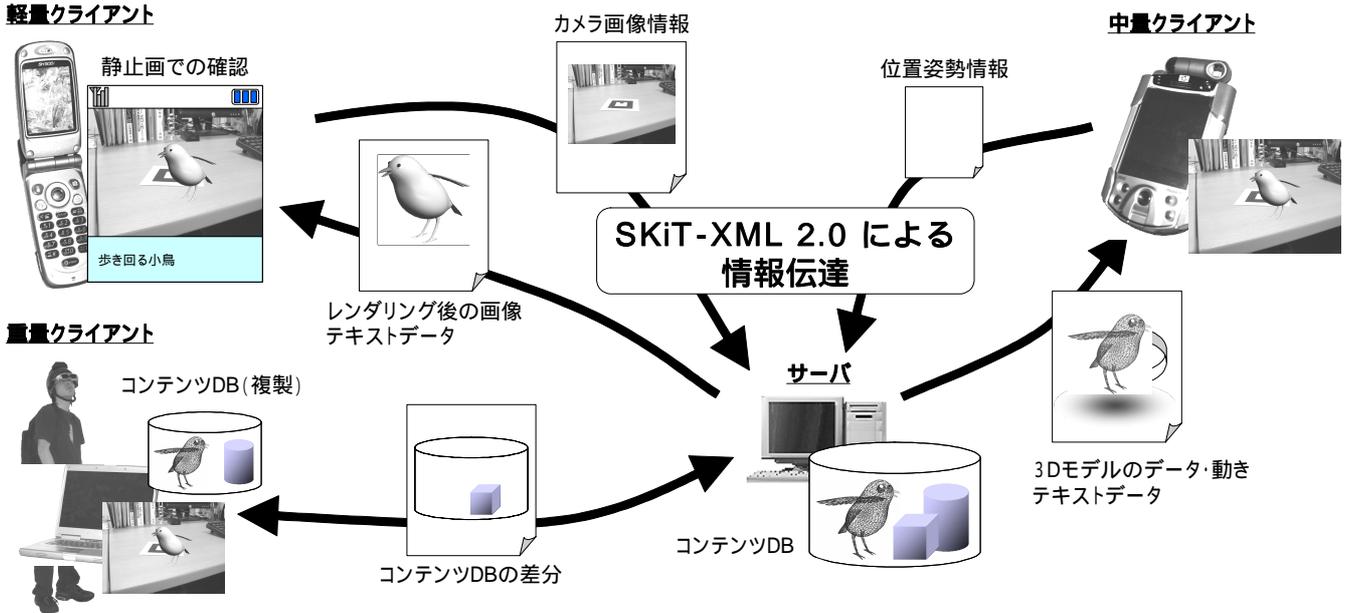


図 2 SKiT-XML のリクエストとレスポンスの内容

基本的な処理は、(1)クライアントからサーバに対して、依頼する処理の種類を RequestType 要素に格納して SKiT-XML を送信、(2)サーバ側で依頼された処理を実行し、処理のステータスを Status 要素に格納した上で、処理結果とあわせて SKiT-XML として返送、という流れになる。

軽量クライアントを用いて MR 情報の提示を行う場合、クライアントは位置姿勢検出に利用するカメラ画像をサーバへ送信する必要がある。画像自体を XML で記述することは非効率的であるため、別途サーバに送信することとする。サーバではあらかじめ画像の大きさや画像形式などを把握しておかなければ処理が困難なため、カメラ画像の情報を CameraImage 要素に格納しサーバに送信する。さらに、軽量クライアントでは MR 情報をサーバが生成するため、Terminal 要素にクライアントの MR 情報を提示する部分の大きさや提示可能な画像形式などの情報を格納することとする。サーバで生成された MR 情報は MRInformation 要素に格納される。MRInformation 要素にはその MR 情報の種類を指定する Type 要素と実際の MR 情報を格納する要素を持つ。MR 情報の種類としては、サーバで 3D モ

デル等のレンダリングを行った結果の MR 画像、アノテーション、メッセージの 3 つがある。MR 情報の種類が MR 画像の場合とアノテーションの場合にはその情報をクライアントの画面中のどの場所に提示するかという情報も格納する。

中量クライアントの場合は、リクエストにコンテンツを取得する際の自身の位置姿勢を Position 要素に格納する。サーバはクライアントの周辺のコンテンツを、Contents 要素に格納してレスポンスとして返す。このとき 3 次元空間中に存在するコンテンツは、Object 要素に、画面上に提示されるコンテンツは ScreenObject 要素に格納される。以降では、コンテンツに含まれる 3D モデルやアノテーションなどの個々の要素を仮想オブジェクトと定義し、次節において仮想オブジェクトの記述方法の詳細について述べる。

重量クライアントでは、サーバとの間でコンテンツを送受信するため、リクエスト及びレスポンスに ContentsDBUpdate 要素を追加しコンテンツ情報をやりとりすることにした。ContentsDBUpdate 要素については 3.5 節で述べる。

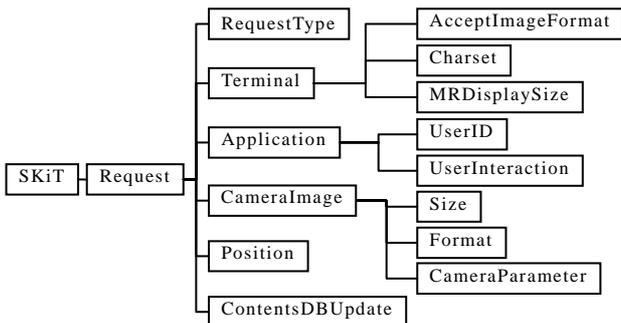


図 3 リクエストの構造

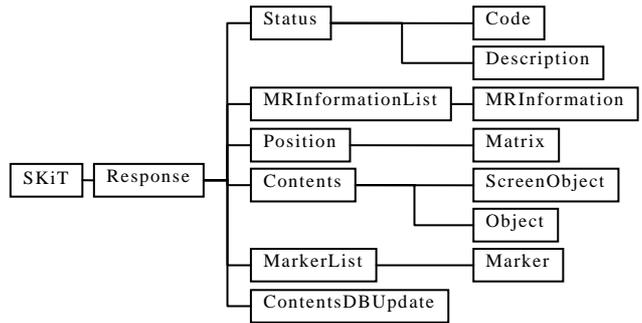


図 4 レスポンスの構造

3.3. 仮想オブジェクトの記述

Object 要素及び ScreenObject 要素以下は図 5 に示す構造を持つ。3D モデルとして記述可能な形状は、球、直方体、円柱、円錐、三角形と VRML データであり、これらを組み合わせて 1 個の 3D 仮想オブジェクトを構成する。この他に画像やテキストなども仮想オブジェクトとして扱うことが可能となっている。

我々は複数の端末が同一の複合現実世界を共有しユーザの協調作業が可能な共通フレームワークの実現を目指している。そこでは、複合現実世界を共有する全てのクライアントで同じコンテンツが提示されなければならない。特に提示するコンテンツが時間によって変化する場合、各端末で同期をとり同一の動きを提示可能なものとする必要がある。

そこでここでは、仮想オブジェクトの動きについて検討する。仮想オブジェクトの動きは、オブジェクト自身の形状が変化する場合とオブジェクトが環境中を移動する場合の 2 つに分けて考えることができる。このとき各クライアント及びサーバ間で同期をとる際の優先度は、オブジェクト自身が変化する場合よりもオブジェクトが環境中を動く場合の方が同期の優先度が高いと考えられるが、ここでは両方の動きに関して同期をとることを目指して設計を行う。

それぞれの変化について、各フレームでのオブジェクトの状態を全て記述することは現実的ではない。ここでは、SVG におけるアニメーションの記述と同様に [6]、任意の時刻におけるオブジェクトの状態を記述し、その間の状態については補間を行うこととした。本研究では、仮想オブジェクト自身の変化を Transform シーケンス、仮想オブジェクトの環境中における移動を Movement シーケンスと呼び、各々について記述方法を検討した。

Movement シーケンスについては、世界座標系における仮想オブジェクトの平行移動、回転、スケールの変化という要素が考えられる。これらの情報は、ある時間での位置、姿勢、スケールの 3 つの情報があれば

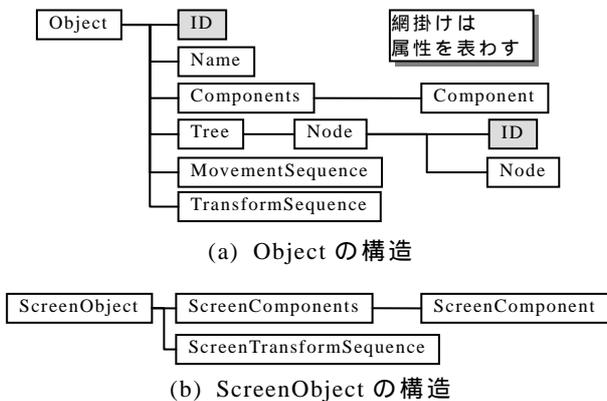


図 5 仮想オブジェクトの構造

記述が可能だと考えられるが、平行移動については速度の変化の記述を可能とするために、ある時間での位置およびその地点での速度ベクトルの 2 つの情報を利用することとした。

一方、Transform シーケンスについては、本研究では仮想オブジェクトは複数の部品で構成されており、各部品で構成される木構造で定義されるものとした上で、オブジェクト自身の形状の変化はある時間における各部品の親ノードに対する位置、姿勢、スケールを記述することとする。

これらの 2 つのシーケンスはいずれも、シーケンスの開始時間を格納する StartTime 要素を持つ。シーケンスは Block 要素と Step 要素から構成される。Block 要素は複数の Block 要素か複数の Step 要素を格納する要素である。Block 要素は繰り返しを指定するための Loop 属性と RepeatCount 属性を持つ。Loop 属性は真偽値を指定し、Loop 属性の値が true の場合にそのブロックが永久に繰り返されることを表わす。RepeatCount 属性はブロックの繰り返し回数を指定する。Step 要素は先に述べた各時点での状態を記述するための要素である。Step 要素にはそのステップの開始時間として親要素となる Block 要素の開始時から経過時間を指定する ElapsedTime 属性を格納する。図 6 にシーケンスの構造を示す。

上で述べたシーケンスの記述により、変化の流れが予め確定しているものはすべて記述可能となる。一方、MR におけるコンテンツの変化としては、ユーザのインタラクションによるものなど、何らかの条件分岐によって発生する変化が考え得るが、これについては、アプリケーションがコンテンツ、すなわち仮想オブジェクトのシーケンスを書き換えることで実現することとする。

ここまで述べたシーケンス記述をもとに MR 情報を生成するためには、現在の時間の前後の 2 ステップ間を補間して現在のオブジェクトの状態を求めるための機構が必要となる。シーケンス記述から実際に提示するコンテンツの状態を取得するためには、まず補間対象となる 2 つのステップを決定し、その後 2 つのステップ間を補間する。ステップを決定するための手順

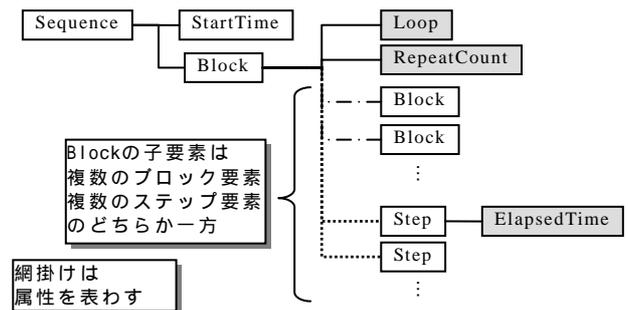


図 6 シーケンスの構造

としては、対象となるステップが含まれるブロックを決定しその後ブロック内のどのステップを補間するかを決定する。まずシーケンス中の各ブロックに要する時間を求め、現在の時間に対応するブロックを決定する。次にそのブロック内のどのステップが補間対象かを決定する。2つのステップが決定した後は、Movementシーケンスの回転、スケール及び Transform シーケンスの位置、姿勢、スケールについては線形補間を行い現在の状態を決定する。Movement シーケンスの平行移動については速度ベクトルを時間について積分することである時間で位置を求めることが可能である。速度ベクトルによって求められる位置と次ステップに記述されている位置が一致しない場合にはベクトルの値を補正して位置を決定する。

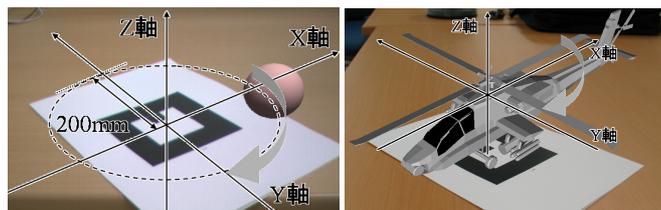
3.4. シーケンスの記述例

前節で検討した記述方法を利用した動きのあるオブジェクトの表現について述べる。ここでは単純な仮想オブジェクトの動きとして図7に示すものを想定する。このとき、図7(a)の SKiT-XML の記述例を図8に示す。このシーケンスは原点を中心とする半径 200mm の XY 平面上の円上を周期 2 秒で動く円運動を記述したシーケンスの一部である。

図7(b)の SKiT-XML の記述例を図9に示す。このシーケンスは図7(b)の例に示したように、ヘリコプターを表現する本体とロータの2つの部品から構成されるオブジェクトでロータがZ軸について回転している様子を記述したもので、ヘリコプターの本体が ComponentID 1 で、ロータが ComponentID 2 で記述されている。

3.5. コンテンツデータベースの同期

2章で述べたように、重量クライアントはサーバとコンテンツデータベースの同期を取る必要がある。本フレームワークではコンテンツの更新は全て重量クライアントのリクエストとして行われる。重量クライアントは起動時にサーバの全てのコンテンツを取得する。その後自身のコンテンツに変更が無い場合でも定期的にサーバに更新の有無を問い合わせる。サーバで更新があった場合はコンテンツデータベースの差分情報をサーバから受信する。重量クライアントは自身のコンテンツに変更があった場合にはその情報をサーバに送



(a) Movement シーケンスの例 (b) Transform シーケンスの例

図7 仮想オブジェクトの動きの例

信する。ただし、同時に複数の重量クライアントがコンテンツの変更を行おうとした場合にはサーバは排他制御を行い、最初に変更を行ったクライアントからの変更のみを受け取り、後から変更を行なったクライアントに対しては先に受理されたコンテンツデータベースの差分を送りクライアントに反映させることで複数のクライアントが同時にコンテンツの変更を行わないようにする。コンテンツの変更として想定されるものには、コンテンツデータベースの変更とコンテンツとして配置されている画像や VRML データなどのファイルの変更が考えられる。コンテンツデータベースの同期を行うときはデータベースを更新するためのSQL文をリクエスト又はレスポンスとして送信する。ファイルの変更については、クライアントでファイルの変更が発生した場合は、サーバ側で位置姿勢検出を行う際に画像を送信する手順と同様に、変更のあったファイルについての情報を SKiT-XML を利用してサーバに送信してから実際のファイルを別途サーバに送信する。サーバでファイルの変更が発生した場合にはサーバはクライアントが取得可能な場所にファイルを配置し、そのURLを SKiT-XML に記述してクライアントに渡す。

4. 実装及び実験

ここまで説明した SKiT-XML を利用してコンテンツ情報を送受信するサーバ及びクライアントを実装した。実装した環境はサーバについては CPU が Pentium4

```
<?xml version="1.0" ?>
<SKiT>
  <Response>
    :
  <Contents>
    <Object ID="1">
      <Name>test</Name>
      <Components>
        <Component ID="1">
          <Name>"test"</Name>
          <Type>Model</Type>
          <Model Mask="false">
            <SimpleModel>
              <Form>
                <Sphere>
                  <x>100</x><y>100</y><z>100</z>
                </Sphere>
              </Form>
              <Color>
                <Red>1</Red><Blue>0.7</Blue>
                <Green>0.7</Green><Alpha>1.0</Alpha>
              </Color>
            </SimpleModel>
          </Model>
        </Component>
      </Components>
    </Object>
    <MovementSequence>
      <StartTime>
        <time>...</time>
      </StartTime>
      <MovementBlock Loop="true" RepeatCount="1">
        <MovementStep ElapsedTime="0.0">
          <MovementMove>
            <VelocityVector>
              <dx>0.0</dx><dy>628.3</dy><dz>0.0</dz>
            </VelocityVector>
            <MovementPosition>
              <x>200.0</x><y> 0.0 </y><z>0.0</z>
            </MovementPosition>
            <MovementMove>
              <Orientation AutoDirection="true">
                :
              </Orientation>
            </MovementMove>
          </MovementStep>
        </MovementBlock>
      </MovementSequence>
    </Contents>
  </Response>
</SKiT>
```

仮想オブジェクト(球)の定義
仮想オブジェクトの動きの定義

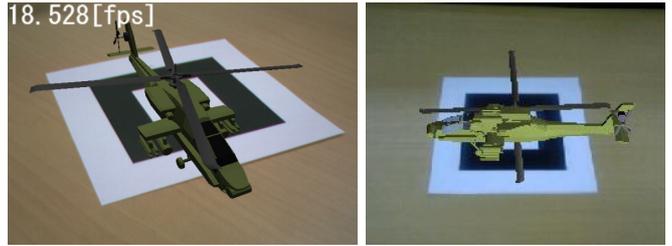
図8 SKiT-XML の記述例(一部)

3.06GHz,メモリが 1GB の PC を利用し,クライアントは Dell の Precision M60(CPU: Intel Pentium M 2 .1GHz,メモリ:2GB)と携帯電話 FOMA SH901iC を用いた .

仮想オブジェクトの提示とコンテンツデータベースの更新についての実験を行った .

仮想オブジェクトの提示については想定した表現が可能であるか確認する実験を行った . シーケンスによって動きを付した仮想オブジェクトを準備し,想定通りにコンテンツが提示されるかどうかを確認した . 図 10 に動作結果を示す .

またコンテンツデータベースの同期についての実験を行った . 実験は 2 台の重量クライアントを起動し,片方のクライアントでコンテンツの変更を行い,その変更がサーバのコンテンツデータベースに反映され,もう一方のクライアントに反映されることを確認するという方法で行った . 実験の結果クライアントでコンテンツの変更を行いサーバのコンテンツデータベースに反映されることと,サーバのコンテンツデータベースの変更がクライアントに反映されることが確認できた . サーバのコンテンツデータベースの変更も正しく行えることから,軽量クライアント及び中量クライアントにおいてもコンテンツの同期が実現できることが確認された .



(a) Precision M60

(b)携帯電話 SH901ic

図 10 仮想オブジェクトの提示例

5. むすび

本稿ではモバイル複合現実感システムにおけるコンテンツ記述言語の設計方針について議論した . サーバ・クライアント間で送受信が必要となる情報について検討を行い,実際にやりとりされるコンテンツの記述方法について XML の仕様に基づいてコンテンツ記述言語 SKiT-XML 2.0 を設計・実装した . また中量クライアントで MR 情報提示を行う際に必要となる動きのある仮想オブジェクトの記述方法について検討及び設計を行った . さらに,重量クライアントとサーバでコンテンツの同期を取るために必要な情報についての検討を行った .

処理系を実装し,設計した SKiT-XML 2.0 によって我々の提案するフレームワークでの情報記述が可能であることを確認した . 特に動的なオブジェクトに関する記述が想定通りに動作することが確認できた . また重量クライアントとサーバ間でのコンテンツの同期も想定通りに動作することが確認できた .

今後はフレームワーク上で複数のアプリケーションを実装し,SKiT-XML 2.0 によって必要な情報が十分に記述可能かを確認していく予定である .

謝 辞

本研究の開発・実装作業の一部に携わった古野光紀氏,佐々木亮一氏,平岡貴志氏,中本浩之氏,玉田裕貴氏に感謝いたします . また,本研究は科学研究費補助金(基盤研究(B) No.17300039)及びハイテク・リサーチ・センター整備事業の一部の補助を受けて行われました .

文 献

- [1] 「特集: 複合現実感」日本バーチャルリアリティ学会論文誌, Vol.4, No.4, 1999.
- [2] 「特集: 複合現実感 2」同上, Vol.7, No.2, 2002.
- [3] 「特集: 複合現実感 3」同上, Vol.10, No.3, 2005.
- [4] 柴田 史久, 木村 朝子, 橋本 崇, 古野 光紀, 平岡 貴志, 田村 秀行: “多様な可搬型機器に対応可能な複合現実感システムの共通フレームワークの設計と実装”, in [3], pp.323-332, 2005.
- [5] The Virtual Reality Modeling Language Ver.1.0 Spec, <http://www.web3d.org/x3d/specifications/vrml/VRML1.0/index.html>
- [6] Scalable Vector Graphics (SVG) 1.1 Specification, <http://www.w3.org/TR/SVG11/>

```
<?xml version="1.0" ?>
<SKiT>
  <Response>
    <Contents>
      <ScreenObject>...</ScreenObject>
      <Object ID="1">
        <Name>test</Name>
        <Components>
          <Component ID="1">
            <Name>body</Name>
            <Type>Model</Type>
            <Model Mask="false">
              <VRML Serial="0">
                <URI>http://fafnir.mclab.ics.ritsumeit...
              </VRML>
            </Model>
          </Component>
          <Component ID="2">
            ...
          </Component>
        </Components>
      </Object>
      <TransformSequence>
        <StartTime>
          <time>...</time>
        </StartTime>
        <TransformBlock Loop="true" RepeatCount="0">
          <TransformStep ElapsedTime="0.0">
            <Transform ComponentID="1">
              <Rotation>
                <x>0.0</x><y>0.0</y><z>0.0</z>
              </Rotation>
              <TransformMove>
                <x>0.0</x><y>0.0</y><z>0.0</z>
              </TransformMove>
              <TransformScale>
                <x>1.0</x><y>1.0</y><z>1.0</z>
              </TransformScale>
            </Transform>
          <Transform ComponentID="2">
            <Rotation>
                <x>0.0</x><y>0.0</y><z>-90.0</z>
            </Rotation>
            <TransformMove>
                <x>0.0</x><y>0.0</y><z>0.0</z>
            </TransformMove>
            <TransformScale>
                ...
            </TransformScale>
          </Transform>
        </TransformStep>
      </TransformBlock>
    </TransformSequence>
  </Contents>
</Response>
</SKiT>
```

仮想オブジェクト (VRML) の定義

仮想オブジェクトの動きの定義

図 9 Transform シーケンスの記述例 (一部)