

# 三次元陳列空間制御システム VisualShowcase における 問題点と解決手法

岡野真一 松岡寿延 荒川則泰 奥山浩伸 泉谷忠寛 大浦啓一郎 吉藤右子

日本電信電話株式会社 サービスインテグレーション基盤研究所 第一推進プロジェクト

## 1. はじめに

ブロードバンドネットワークの急速な普及に伴い、高速なネットワークを用いて大量情報のやり取りを行うさまざまなサービスの提供が始まっている。このようなサービスの一つとして、我々は三次元陳列空間制御システム VisualShowcase の研究開発を行っている。VisualShowcase は、大量のカタログ情報をブロードバンド回線を介して取得し、それらを三次元空間上に配置するシステムである。これにより、利用者の興味を引くユーザインタフェースを提供するとともに、三次元空間への配置を活かした商品陳列を行うことによって、商品選択能力を向上させることを目的としている。



図1 VisualShowcase の画面例

## 2. 問題点

VisualShowcase のように三次元空間上に多数のカタログ情報を陳列するソリューションには以下のような問題点が存在すると考えられる。

- コンテンツ再利用が困難  
多くのソリューションで、特定のクライアントソフトの利用が前提であるため、異なる閲覧環境でのコンテンツの再利用が難しい。
- コンテンツ作成手順の複雑化  
三次元空間上における動作の設定など、エンドユーザレベルでのコンテンツ編集作業が困難である。

*Problems and the solutions for 3D visual showcase system*  
Shin-ichi OKANO, Hisanobu MATSUOKA, Noriyasu ARAKAWA, Hironobu OKUYAMA, Tadahiro IZUMITANI, Keiichirou OOURA, Yuko YOSHIFUJI  
NTT Service Integration Laboratories

## ● サーバ負荷の増大

サーバ側に多くの機能を持たせる場合、単純なクライアント-サーバの構成では、サーバへの負荷が増大する場合が多い。以下では、このような問題点を解決するために VisualShowcase で用いた手法を述べる。

## 3. 解決手法

### 3.1. ブラウザ非依存な中間表現の採用

VisualShowcase では、VSML (Visual Service Markup Language) [1] と呼ぶ XML ベースの記述言語を用いて三次元陳列空間を定義している。VSML は利用する三次元ブラウザに依存しない中間表現となっており、今回三次元ブラウザとして用いたフィールドエクスプローラ [2] の他、VRML ブラウザや SVG ブラウザを利用者側のブラウザとして利用することも可能な構成となっている。これにより、さまざまな閲覧環境でのコンテンツの再利用が容易となっている。

### 3.2. 動作の部品化とその制御

表示される三次元空間(シーン)を作成するにあたって、その空間上での動作を設定する作業は、非常に困難なものとなっていた。これを解決するために、アクションの概念を導入した。アクションとは、三次元陳列空間上で処理される動作の最小単位であり、例えば画像(ページ)のピックアップ動作やカメラ位置の移動といったプリミティブな動作を含んでいる。これらのアクションを、順次実行、繰り返し実行といった単純な制御構造にしたがって実行可能とした。例えば、ボタン押下のユーザ操作に対して、カメラ移動、ページレイアウト変更の指定回数繰り返し、ページ回転開始、といった一連の動作は VSML で図2のように記述することができる。また、繰り返し処理中に指定されたイベントが発生した場合に、例外処理を行うことも可能である。この方式の採用により、単純な動作を組み合わせての複雑な動作の設定や、動作の再利用、動作順序の入れ替えなどの作業を効率的に行うことが可能となった。また、繰り返し実行と例外処理を利用することによって、無操作時

のオートデモとユーザ操作による復帰などを実現することも可能となった。

なお、VisualShowcase では、陳列するページの設定や、操作ボタンに対するアクションの設定を行うための GUI オーサリングツールを提供している。これを用いることによって、三次元空間上でさまざまな動作を組み合わせることで人目を引くシーンを簡単に構築することが可能だけでなく、画像ファイルの差し替えなどの単純な作業によって、ページを更新することも可能であるため、エンドユーザ側で必要に応じてシーンを更新することが容易となっている。

```
<ActionDef id="actionlist1">
  <InternalService>
    <Seq>
      <DoAction id="action1" name="MoveCamera"/>
      <ltr>
        <ltrCondition expression="5" type="Times"/>
        <Seq>
          <DoAction id="action2" name="SetLayout"/>
          <DoAction id="action3" name="SetLayout"/>
          <DoAction id="action4" name="SetLayout"/>
        </Seq>
      </ltr>
      <Exception>
        <EventConditionAction>
          <Event name="event1"/>
          <Action>
            <DoAction id="ex1" name="GotoTopScene"/>
          </Action>
        </EventConditionAction>
      </Exception>
    </ltr>
    <DoAction id="action5" name="StartRotate"/>
  </Seq>
</InternalService>
</ActionDef>
```

図2 アクション記述の例

### 3.3. サーバ機能配置方法の柔軟化

サーバ側での VSML の処理負荷を軽減するために、端末フロントエンドを導入した。端末フロントエンドとは、サーバ側の機能であった表示制御機能を、独立したモジュールとして切り出したものである。端末フロントエンドは、必要に応じて VSML や商品情報を http 経由で外部サーバから取得し、三次元ブラウザの表示制御を行う。端末フロントエンドは、システム内のさまざまなノード上に配置することが可能となっている。利用者端末上に配置すれば 3 次元ブラウザの制御は利用者端末のみで完結し、サーバ負荷を大幅に軽減することができる。また、利用者端末の性能が低い場合には、LAN 内に端末フロントエンド単体のノードを置いて共有するなど、さまざまな運用形態に柔軟に対応することができる。図 3 に利用者端末に端末フロントエンドを配置

した場合の基本的なシステム構成を示す。

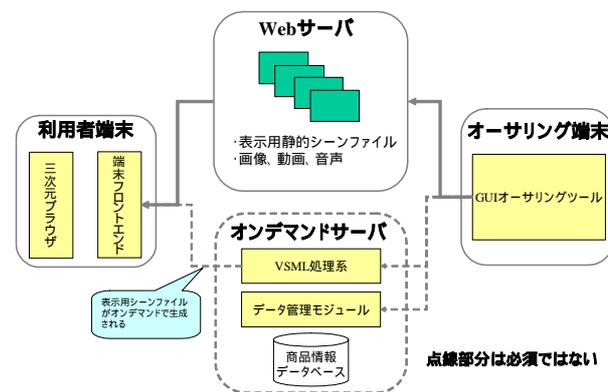


図3 VisualShowcase のシステム構成

### 3.4. 事前変換方式の採用

VisualShowcase では、VSML を実行時に解釈し、その結果によって三次元ブラウザを制御することが可能となっている。これにより、検索結果の表示のような動的な表示変更に対応し、商品の価格変動などをシーンに反映することが容易となっている。しかし、全てのシーンについてこの方式を採用すると、サーバ、あるいは端末フロントエンドの処理が非常に重く、現状のシステムではレスポンスタイムの面で不利なものになってしまう。

この問題を解決するために、動的に変化しない情報のみを含むシーンに対しては、表示に利用する三次元ブラウザの実行形式に予め変換しておく方式を採用した。この方式により、実行時に変換処理を行う場合と比べて約 60%の速度向上を実現した。

### 4. おわりに

VisualShowcase を用いることにより、複雑なユーザインタフェースを簡単に構築し、またさまざまな運用形態で利用することが可能となった。今後は、外部システムとの連携などを実現し、カタログ陳列分野以外での適用も検討したい。

### 参考文献

[1]松岡・岡野・泉谷・小野・佐藤:”三次元陳列空間制御システムの検討”, 電気学会 通信研究会 CMN-03-06, 2003  
 [2]木綿・阿部・中野・竹内・川村・金井:”ネット空間提供プラットフォーム InfoLead”, 情報処理学会 第63回全国大会 1W-03, 2001