

3D-CG Animation のシナリオ記述言語 CPSL と Cyber Teaching Assistant の開発

新 藤 義 昭[†] 松 田 洋[†] 鈴 木 誠 史[†]

情報処理演習室での質疑応答コミュニケーション用に開発した Cyber Teaching Assistant (CTA) という CG-Animation システムと、そのシナリオを記述するハイパーテキスト言語 CPSL (Cyber Person Scenario Language) について述べる。CTA は、受講者からの質問が殺到したとき、人間型 3DCG モデル (Cyber Person) を回答代行者とした CG-Animation を提示するシステムで、CPSL Browser と CTA Starter で構成される。CPSL Browser は各々の受講者側 PC で別個に動作する。CTA のシナリオは CPSL で記述し、ネットワークサーバに格納する。講師は受講者の画面を見て、CTA のシナリオで質問に回答できると判断したときは、シナリオを指定して質問者の CTA を遠隔起動する。CTA は、シナリオに従って必要な舞台や小道具を設定し、顔表情を変えたり体演技したりしながら台詞を音声合成で読み上げて質問に回答する。CPSL は、非専門家でも低コストで CG-Animation を制作できることを目標に開発したハイパーテキスト言語で、HTML 言語に類似した文法と簡単な平文表記で制作できる。本論文では、CPSL の機能と CTA のシナリオの制作技法を述べ、従来の技法による CG-Animation の制作コストとの比較実験を行い CPSL の有効性を確認した。

3D-CG Animation Scenario Language for Cyber Teaching Assistant: CPSL

YOSHIAKI SHINDO,[†] HIROSHI MATSUDA[†] and JOUJI SUZUKI[†]

This Paper describes CG-Animation system named Cyber Teaching Assistant (CTA), developed for new type communication between Teacher and Students in the Computer Seminar Room and Animation Scenario Language named CPSL (Cyber Person Scenario Language). CTA is a Cyber Person Animation system based on 3DCG technology, which includes CPSL Browser and CTA Starter. CPSL Browser is installed in each student's PC. CTA's Scenario files are written by CPSL and they are saved in the network file server. When teacher detects the student's question or help request and if student's difficult situation would be resolved by the CTA Scenario, he can start the CTA with appropriate scenario by using CTA Starter. CTA sets up the stage and properties and reads aloud the Japanese text by Voice Synthesizer with Facial Expression and plays specified body Action according to the Animation Scenario. CPSL is a Script Language similar to the HTML Language to produce the Animation Scenario by a low cost even for a non-expert. This paper describes a function of CPSL and production technique of a scenario of CTA. In addition, we did a comparison experiment with a production cost of CG Animation by conventional technique and confirmed the effectiveness of CPSL.

1. はじめに

筆者らは情報基礎教育として情報リテラシー講座やプログラミング基礎演習の講座を開講しているが^{19)~21)}、これらの講座を効果的に行うために専用の情報処理演習室を開発した。この演習室の受講者側 PC には、(1) 教師からの提示を行う専用の提示画面、(2) 数字キーと HELP キーが配置された小型の

KeyPad が配置されている。一方、教壇席 (Cockpit と呼ぶ) には、Class Analyzer Computer (CAC) と名付けた PC が設置されており、画面に表示される教室の Seat-Map をもとに、次のような受講生支援を行うことができる。

- ① 講師は受講者の提示画面に教材を提示する。
- ② 選択した受講者の PC の画面をモニタする。
- ③ 受講生は、困った状況に陥ると、自分の KeyPad を使って質問を発信する。
- ④ 受講生の質問を Seat-Map で検知すると、その受講者の画面をモニタする。

[†] 日本工業大学工学部情報工学科
Department of Computer and Information Engineering,
Nippon Institute of Technology

⑤ 講師は、受講生を 1 対 1 の音声対話で支援したり、受講生の PC を遠隔操作して問題解決を行う。しかし、多数の質問が教壇の Cockpit に殺到すると講師は順番に対応するので、受講者は長い間待たなければならない。この問題を緩和し効率的な演習授業を可能にするため、実時間の 3D-CG Animation を利用した Cyber Teaching Assistant (CTA と呼ぶ) システム^{13),14)}を開発した。

2. 先行研究

CTA のようなコミュニケーションを行う方法として文字、写真、図を組み合わせたハイパーテキスト提示が考えられ、Animated Agent を用いた映像は、これよりも、学習者の理解度が高いことが報告されている^{10)~12)}。実写による Movie の利用 (Video On Demand)⁹⁾も有効であるが、多数の Movie をデータベース化する場合、制作コスト (制作の人的負荷) が大きい点、Video Movie の容量が大きいなどが難点である。

CTA を 3D-CG Animation で実現する (以後 CTA-Animation と呼ぶ) 技法としては、次の 3 つが考えられる。

- (a) 対話型 CG-Animator を用いて、Animation、音声、効果音、音楽などを別々に制作し、Video Editor で CTA-Animation Movie に統合する。
- (b) CTA-Animation を実時間提示するプログラムをプログラミング言語で開発する。
- (c) CG-Animation、音声、効果音、音楽などを統合して扱うことのできるハイパーテキストまたはスクリプト言語で CTA-Animation のシナリオを記述し、専用のブラウザで表示する。

まず、(a) による制作技法について検討する。汎用の CG-Animator は数多く製品化されているが、制作コストの観点から人体モデル専用の CG-Animator を対象²⁹⁾とする。人体の 3DCG モデルはライブラリから選び、体の演技はキーフレームをタイムチャート上に配置する。顔の表情は登録コンポーネントを利用する。ただし、合成音声の録音、効果音、音楽、字幕機能は、それぞれ別途に制作し VideoEditor³⁰⁾で統合し Movie にする。この技法は、映像の美的洗練度は高いが、高解像度でレンダリングすると長時間の CPU コストが必要であること、Movie の容量が大きいこと、デマンド起動する際の通信負荷が大きいことが難点である。また合成音声と、顔表情や体演技の同期をとる作業が難しく、制作コストを大きくしている (6 章で詳しく述べる)。

次に、(b) による制作技法について検討する。OpenGL²³⁾、Microsoft SpeechSDK などの低水準ライブラリのみで制作する方法もあるが、制作コストの観点から、人間型 3DCG モデルを前提とした SDK (System Design Kit) を利用して制作する方法を検討する。このような研究報告として、米国ペンシルバニア大学が開発した Jack^{1),2)}が有名である。Jack は、Lisp 言語用の SDK として軍用に開発され、民生化された際に工場生産ラインのシミュレーションなどの適用事例が報告されている³⁾。Jack は、手操作や自律歩行、多人数の同期動作など優れた機能を有しているが、合成音声、効果音、音楽、顔表情などを同期して統合する機能が低い。

人間型 3DCG モデルを用いたインタフェースエージェントの研究事例としては、電総研のジョニー⁴⁾、オムロンの Stellar⁵⁾などがある。これらは、音声による発声機能、音声と同期して顔表情を変えたり、体をある程度演技させる機能を有しており、SDK も頒布されている。しかし研究目的がエージェントの自律動作や環境認識であるため、舞台や小道具の配置や動き、多人数のエージェントの登場 (さらにジョニーは上半身のみ) やエージェントどうしの同期処理などが考慮されていない。

筆者らも、同様な目的で 3D-CG Animation を制作する PiasGL4 Library^{24),25)}や、C++ 言語用クラスライブラリ CPDK (Cyber Person Design Kit) を開発し報告してきた^{26),27)}。プログラミングを容易にするための数々の工夫を開発してきたが、オブジェクト指向プログラミングの専門技術や CG や音声合成に関する知識と経験が必要となり、非専門家が容易に制作できる段階には達しなかった。

(c) の研究事例として、WEB サイトの Presenter としてマスコットキャラクタを用いる MPML という記述言語⁶⁾が報告されている。これは Microsoft Agent³¹⁾によるマスコットキャラクタの発話、動作、簡単な演技をタグ形式で記述可能で、HTML 言語との親和性や音声認識機能など優れた機能を有している。しかし Agent が人体の 3DCG モデルでないこと、舞台、小道具類のモデリング機能がなく、小道具を動かす機能もないこと、カメラワーク (ズームインによる部分的拡大機能など) などの 3 次元空間制御構造がないこと、などから CTA-Animation に必要な機能を満たしていない。

人体モデルの演技を前提とし、舞台の 3 次元空間制御構造を備えたスクリプト言語の研究事例として、NHK が報告した TVML^{7),8)}がある、TVML はニュー

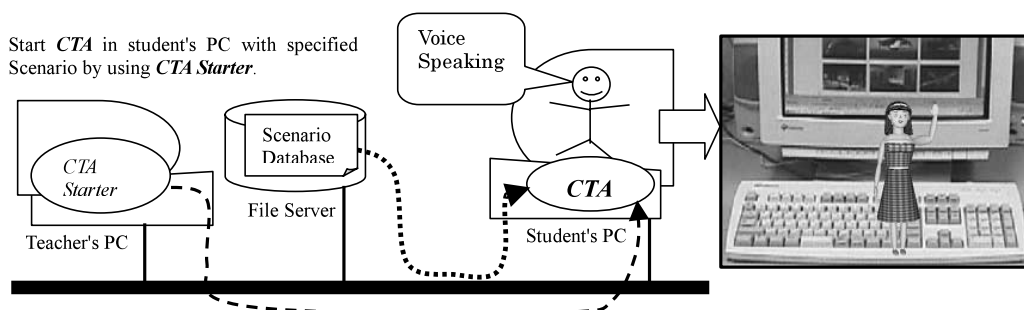


図1 Cyber Teaching Assistant のシステムの概要
Fig. 1 System outline of Cyber Teaching Assistant.

ス番組の自動制作を目指したスクリプト言語で、TVスタジオ設備の忠実な再現機能、発話、字幕、若干の顔表情の変化（6種類）を記述できる。しかし、TVスタジオ設備に忠実である結果として環境設定が複雑なこと、体の演技がニュースキャスタに必要なものに最適化され自由度が低いこと、舞台に配置された小道具を動かす機能が整備されていないこと、言語文法がコマンド形式で多数のパラメータを指定しなければならないため制作工程がプログラミングと似ていること、などから CTA-Animation 制作には適さないと思われる。

3. Cyber Teaching Assistant

本章では、CTA のシステム構成および特徴について述べる。CTA は人間型 3DCG モデル（Cyber Person と呼ぶ）を出演者とした実時間描画の 3D-CG Animation システムで、質問が殺到したときに講師の代わりに質問者に回答する^{13),14)}。CTA の特徴を次にまとめて述べる。

- (1) CTA は、受講者 PC で動作する専用のブラウザ CPSL Browser と、演技を記述するシナリオ CTA-Scenario で構成される。
- (2) CTA は女性の Cyber Person で、その形状データなども Browser とともにインストールされる。
- (3) CTA は、日本語または英語の文章を音声合成機能を用いて読み上げる。また、音声に合わせて顔の表情を変えることができる。
- (4) CTA は、体を自由に動かして具体的な説明（たとえばキーボードを指差しながら）演技を行うことができる。
- (5) CTA-Scenario（舞台や小道具の設定、喋る台詞文章、顔の表情変化、体の演技など）は、CPSL（Cyber Person Scenario Language）というハイパーテキスト言語で記述する。

- (6) CTA-Scenario はネットワークファイルサーバに格納される（Scenario Database）。

講師はサーバ内の CTA-Scenario を指定して、質問者の CTA を、CTA Starter を用いて遠隔起動する。

CTA は受講者 PC 側で動作するので、講師は同時並行して多数の CTA を別々の CTA-Scenario で起動できる。受講者が質問キーを押すと、講師は質問者の画面をモニタするが、その質問が CTA-Scenario で解答できる場合には、自ら答えずに CTA を遠隔起動する（図1）。どうしても 1 対 1 の対話を行わなければ解決しない質問だけを選んで自ら解答する。現在、キー操作や、マウスによる範囲指定操作など、初心者が覚えにくい操作手順の解説や、プログラムのコンパイルエラーの対処などのシナリオを制作して、演習時に講師支援用ツールとして使用している。しかし大きな問題は、CTA-Scenario（または映像）制作の制作コストである。CG やプログラミングに精通していない非専門家でも、容易に CG 映像を制作できる記述方式（または制作技法）が必須である。筆者らは、この観点から非専門家でも低コストで CTA-Animation を制作可能なハイパーテキスト型のシナリオ記述言語 CPSL（Cyber Person Scenario Language）および CPSL Browser を開発した^{13)~16)}。本論文では、CPSL のタグ機能、記述形式および同期処理について詳しく述べる、

また CPSL Browser のシステム構成および Cyber Teaching Assistant としての利用方法を示す。さらに CTA-Animation の制作コストについて、3 種類の技法で同様な映像を制作した際の制作コストを比較する実験を行い、CPSL の特徴および優位性について検証した結果を報告する。なお、CPSL Browser のインプリメンテーションについては、実時間 CG 表示におけるフレームレート保持の問題¹⁷⁾、人体や小道具類との接触、衝突判定の問題¹⁸⁾、人体モデルの臨場感を増すための自律運動の問題⁵⁾など、様々な要素技術が統合

されているが、本論文ではこれらは取り扱わないこととする。

4. CPSL の機能と構文規則

4.1 CPSL の開発目標

CPSL は、CTA-Animation の Scenario を非専門家が低コストで制作できることを目標に設計したデータベースのハイパーテキスト言語である。記述形式は、HTML 言語と類似している。CPSL で記述された Scenario は、テキストエディタやワードプロセッサで編集できる。開発目標を次に述べる。

- (1) 人体の 3DCG モデル (Cyber Person) は、CPSL Browser の一部として用意され、1 つのシーンに何人でも登場できる。また異なる音声合成器 (またはドライバ) を割り当てることができる。
- (2) Scenario の主軸を音声合成で発声する台詞とし、文法形態を HTML 言語と同じタグ形式とする。
- (3) 顔表情の変化と体の演技は、台詞原稿の中にタグで挿入していく。これにより、音声、表情、体演技の同期が自然に確保できる。
- (4) 動作指示にともなうパラメータの指定をできるだけ省略可能にするため、環境設定、動作モード、作動量などは**状態変数方式**とした。ほとんどの場合は初期値で動作する構造としたため、初心者は多量のパラメータの指定から解放される。
- (5) プログラム言語のループ処理構造の代わりに、フレーム区間処理という技法を導入した。これについては、4.2, 4.7, 4.8 の各節で述べる。
- (6) Block 同期処理、状態変数のスコープ制御により、音声、表情、演技、字幕のタイミング同期や、複数の Cyber Person の並行動作を記述できる。これらについては 4.2 節で述べる。
- (7) 表情の変化は、データベースに登録した顔表情と、これを呼び出す抒情詞を台詞の中に挿入することで行う。これについては 4.4 節で詳しく述べる。
- (8) 体の演技は 2 種類の方法を開発した。1 つは、人体の各関節の可動域をクラスタリングした符号 (BACS と呼ぶ) に名前を付けてデータベース化し、これを呼び出す演技指示語と作動量 (100% 方式) を台詞の中に挿入する方法である。これについては、4.5 節で詳しく述べる。
- (9) 体の演技を行うもう 1 つの方法は、CG Anima-

```

1) Case of Non-Nesting Block
  <TAG OptionSwitch>Body Text</TAG>

2) Case of Nesting Block
  <TAG Option Switch>Body Text1-1
    <TAG Option Switch>BodyText2
  </TAG>
  BodyText1-2
</TAG>

3) Case of Nesting Blocks
  <TAG Option Switch>Body Text1-1
    <TAG Option Switch>BodyText2
  </TAG>
  BodyText1-2
  <TAG Option Switch>BodyText3
  </TAG>
  BodyText1-3
</TAG>

```

図 2 CPSL の構文形式

Fig. 2 Coding syntax of CPSL.

tor と類似した操作性で体演技を対話形式で作り出す演技ツール (BPN Encoder) を用いる方法である。これについても、4.5 節で述べる。

- (10) 舞台となる背景や小道具は、対話形式で制作できるツールで制作し、これを自由に読み込める構造とする。これについては、4.7 節で述べる。

4.2 CPSL の構文形式と状態変数

CPSL の構文形式を図 2 に示す。1Block の記述は <TAG> 記号と </TAG> 記号に囲まれた部分に Body Text と呼ぶ文章を入力する。Body Text の任意の位置に、別の Block を挿入することができる (Block Nesting と呼ぶ)。挿入された Block を、下位 Block と呼ぶ。この場合、第 1 レベルの Body Text は、前半部分と後半部分に分断される。この Block Nesting は何段でも可能である。また、Body Text の複数箇所にも別の Block を挿入することもできる。この場合は、Body Text は、そのつど分割される。

CPSL の記述を容易にするため、各処理が動作パラメータとして参照する値を CPSL Browser が状態変数として保持する (表 1)。<TAG> は、状態変数を参照して指示された処理を実行する。<TAG> ブラケット内の Option Switch は、状態変数に値をセットする。Option Switch でセットされた状態変数の値は、その <TAG> Block およびその Block の下位 Block 群でのみ有効である (状態変数のスコープ機能)。Option Switch を記述した Block の外側にでると、状態変数の値は元に戻る。

状態変数 PERSON は、Cyber Person を複数人登場させたとき、アクティブな Cyber Person を選択する。

表 1 CPSL の状態変数

Table 1 State variables of CPSL.

State Variable	Content
PERSON	Name of Active Person
SCRIPT	ON or OFF (Switch of Subtitles)
NOTIFY	SYNC, ASYNC or ABORT
TTC TIMING	SYNC point of TTC
WAIT	Person's waiting time
RATE	Intermediate frames
REPEAT	Repetition Times of body action
GATE	Gate time of TTC
TRANSIT	Transition time of TTC
STABLE	Stable time of TTC
RELEASE	Release time of TTC
GAIN	Stable gain of AU (%)
REMAIN	Remain gain of AU (%)
POSITION	Located Position (x, y, z)
SCALING	Scaling Factor (sx, sy, sz)
DISTANCE	Distance of walking
ROTAION	Turn angle of Rotation
FRAMES	Frames for Specified Work 1 Frame is 100 mS (Default)

状態変数 FRAMES は、フレーム区間処理を制御する。カメラの移動、Cyber Person の歩行や回転、舞台中の小道具の移動や回転など、プログラミング言語ならばループ構造で記述する部分を代行する。指示された処理は、FRAMES 値で区間分割され、非同期に並行処理で Animate される。

シナリオの中で音声、表情、体演技を同期させたり、複数人の動作を正確に時間制御したりするため、Block 同期処理を定義した。<TAG> は、指示された処理の実行を開始し、</TAG> は、その処理についての同期処理を行う。この同期処理には 3 つのモードがある（表 2）。

SYNC モード（初期値）では、</TAG> は処理終了を待つ。ASYNC モードでは、処理終了を待たずに通過する。ABORT モードでは、</TAG> は処理を強制終了して Block の外側に出る。後述する FACS-AU と BACS-AU の場合は、Time Transition Code（4.3 節で述べる）のどの位置で待ち合わせするかを指定する。

Wait スイッチは、無条件に指定された時間だけ処理を休止する。なお、これらのモードもすべて状態変数である（欄内の○値が初期値）。

4.3 音声合成のための台詞の記述

CPSL の基本は、Cyber Person の台詞原稿である。<TALK> は、台詞原稿と顔表情の自動変化を記述する。<TALK> を用いると、Cyber Person は Body Text を音声合成で読み上げ、顔の表情を台詞に合わせて自動的に変化させる。このため、台詞文章の中から Lyric Word

表 2 ブロック同期処理に関する Option Switch

Table 2 Option switches related to block notification.

Switch	Value	Function
NOTIFY	SYNC	Wait the end of work. (○)
	ASYNC	Passes without waiting.
	ABORT	Terminate work forcibly.
TIMING	NONE	No wait. (Same as ASYNC)
	eTRANSIT	Wait the end of Transition time. (○)
	eSTABLE	Wait the end of Stable time.
	eRELEASE	Wait the end of TTC
WAIT	Integer	Wait for specified time.

（抒情詞）を検出し、顔表情を変化させる。<TALK> は便利であるが、時に制作者の意図とは異なる顔表情を見せる場合もある。<SPEECH>で台詞原稿のみを記述し、<FACE> で顔表情の変化を明確に挿入することもできる（4.4 節参照）。次の 2 つの記述は同等の動作となる。

```
<TALK>今日は誕生日なので私はうれしい。</TALK>
<SPEECH>今日は誕生日なので<FACE>幸福</FACE>
私はうれしい。</SPEECH>
```

初期値では、台詞を音声合成で発声するとともに字幕が表示される。SCRIPT スイッチで自動的な字幕表示をオフにし、<SCRIPT> を用いて音声原稿とは異なる字幕を表示することもできる。

```
<SCRIPT>サイバーガールの登場です。
<SPEECH SCRIPT="OFF">私は恵美です。
  奈々ちゃんの姉です。</SPEECH>
</SCRIPT>
```

この機能は、日本語と英語が混在した文章や、略字表記が使われているときに有効である。次に例を示す。

```
<SCRIPT>キーボードの Enter キーを押してください。
<SPEECH SCRIPT="OFF">
  キーボードのエンターキーを押してください。
</SPEECH>
</SCRIPT>
<SCRIPT>キーボードの Ctrl キーを押してください。
<SPEECH SCRIPT="OFF">
  キーボードのコントロールキーを押してください。
</SPEECH>
</SCRIPT>
```

Cyber Person が複数登場している場合には、PERSON スイッチで Cyber Person を選択することができる。

```
<SPEECH PERSON="恵美">私は恵美です。
  奈々ちゃんの姉です。
</SPEECH>
```

<TALK> と <SPEECH> は音声合成を開始させ、

表3 <TALK> と <SPEECH> の Option Switch

Table 3 Option Switches of <TALK> and <SPEECH>

Switch	Content	Function
PERSON	Person's name	Name of active person
SCRIPT	ON	Display the subtitle. (○)
	OFF	Display nothing

</TALK> と </SPEECH> は同期処理を行う。SYNC モードでは、</SPEECH> と </TALK> は Body Text の読み上げ終了を待つ。次の例では、奈々が話した後に、恵美が話し始める。

<SPEECH PERSON="奈々">こんにちは。 </SPEECH>
<SPEECH PERSON="恵美">こんにちは。 </SPEECH>

しかし、以下の例では奈々と恵美は同時に喋る。

<SPEECH PERSON="奈々">こんにちは。
<SPEECH PERSON="恵美">こんにちは。 </SPEECH>
</SPEECH>

表3に Option Switch を示す。

4.4 顔表情の変化の記述

<FACE> は、Body Text を解析して Lyric Word (嬉しい、ひどいなどの感情を含む抒情詞) を抽出し、顔の表情の変化させる。図3に、顔の表情を変化させるデータフローを示す。

Lyric Word は、Lyric Word データベースを参照して、顔表情語 (FEN: Facial Expression Name) に変換される。さらに、FEN は、Facial Expression データベースを参照して、FACS-AU sequence に変換される。FACS (Facial Action Coding System) は、Ekman らによって開発された顔表情を記述する符号²²⁾で、AU (Action Unit) という顔筋肉の動作部位の番号表である。CPSL では、FACS を左右別々に拡張して 59 個の AU (表4に例を示す) を定義した。Ekman らは基本6表情 (幸福、悲しみ、驚き、怒り、恐れ、嫌悪) を定義したが、CPSL では FEN Encoder (図4) という表情作成ツールで自由に表情を定義して、FEN をデータベースに登録できる。現在、54 種類の FEN を登録した。<FACE> の Body Text として、FEN を直接記述することもできる。

顔表情の例を図5に示す。AU の動作の時間的遷移を記述するため、TTC (Time Transition Code) を定義した。AU の作動量は%で記述し、時間はミリ秒で記述する6個の符号である (図6)。この値は、状態変数として記憶され、<FACE> の Option Switch で値を指定することもできる (表5)。<TALK> は、Body Text 中の Lyric Word を自動検出して顔表情を変化させる。このため、322 語の Lyric Word をデータベー

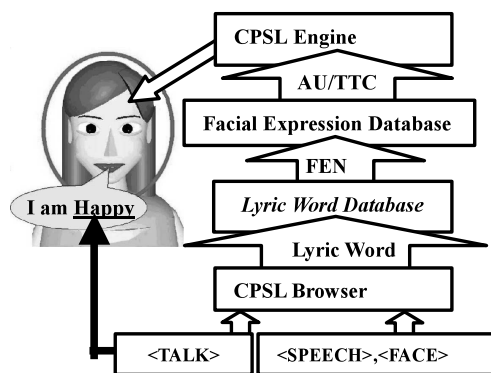


図3 顔表情変化のデータフロー

Fig. 3 Data flow of facial expression.

表4 FACS-Action Unit (AU) の例

Table 4 Samples of FACS-AU.

FACS-AU	Moving Parts
10	Lift up the both sides of eyebrow.
51	Lift up the right side of eyelid.
62	Lift up the left side of cheek
260	Open the mouth
430	Close the both eyes

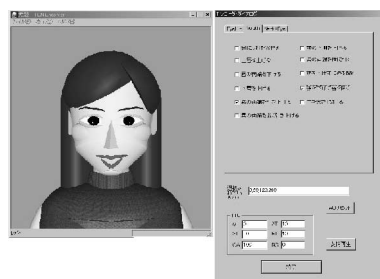


図4 FEN encoder

Fig. 4 FEN encoder.

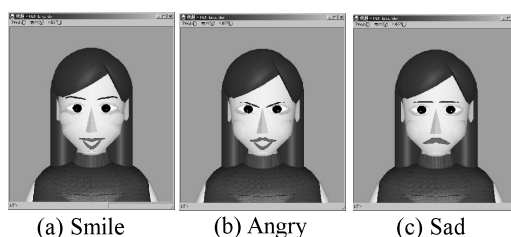


図5 顔表情の例

Fig. 5 Sample of facial expression.

スに登録した。

<FACE> の使用例を次に示す。この例では、「みなさんこんにちは」と話した後に顔の表情を変え始める。笑顔になるまでの遷移時間は1秒である。</FACE> は笑顔になるまで (1秒間) 待ちつづける。そして、「どうぞよろしく」と話す。

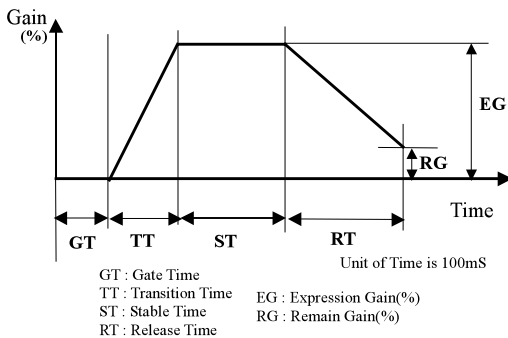


図6 Time Transition Code (TTC)
Fig. 6 Time transition code (TTC).

表5 <FACE>のOption Switch. ()内は初期値
Table 5 Option Switches of <FACE> (default).

Switch	Value	Function
TTC	6 Variables (Comma Separated Value)	GATE, TRANSIT, STABLE, RELEASE, GAIN, REMAIN
GATE	Integer more than 0	Waiting Time(0)
TRANSIT	Integer more than 1	Transition Time(10)
STABLE	Integer more than 1	Stable Time(50)
RELEASE	Integer more than 0	Release Time(10)
GAIN	Integer more than 1	Stable Gain(100)
REMAIN	Integer more than 1	Remain Gain(70)

<MODE TTC=" 0, 10, 50, 10, 100,0">

<SPEECH>みなさん, こんにちは.

<FACE>幸福</FACE>

どうぞよろしく.

</SPEECH>

4.5 体演技の記述

<ACTION>と<MOTION>は,ともにBody Textに記述されたAction Wordを解析して,体演技を行う.図7に体演技を行うデータフローを示す.

Action Wordは, Action Wordデータベース(現在150語を登録)を参照して演技指示語(BPN: Body Performance Nameと呼ぶ)に変換される.<ACTION>の場合, BPNはBody Motionデータベースを参照して, BACS-AU sequenceに変換される.

BACS-AUは,筆者らが開発した体関節動作符号で, FACS-AUと同じ考え方で人間の全関節の動作部位の動き方をそれぞれ分類して,固有の番号を付けた符号表である. CPSLでは, 92個のBACS-AUを定義した(表6に例を示す). 作動量は, FACSと同じく最大可動量を100%として%表示で記述する. BACS-AUはFACS-AUと同じようにTTCを参照して時間的狀態遷移を実現する. 次に<ACTION>の記述例と, 作動後のCyber Personの姿勢を図8に示す.

<ACTION Gain="40,40">体を左に回す.</ACTION>

<ACTION Gain="85,85">左手を前に上げる.</ACTION>

<ACTION Gain="50,50">顔を右に回す.</ACTION>

<ACTION Gain="35,35">左腕を前に上げる.</ACTION>

<MOTION>は, BPNをBody Motionデータベースを参照してMotion Capture Dataが格納されたファイル名に変換する. Motion Capture Dataは, 人体の各関節に割り当てられたアフィン変換行列を, 一定のサンプリング周波数でキャプチャしたデータ列で, BPN Encoder(図9)で対話形式で作成する. 操作方法は, 人体モデル用のCG-Animatorと類似したもので, マウスで手足を動かしてKey Frameを作成してタイムチャート上に割り当てていく方式である. 再生速度や繰返し回数をSwitchで指定できる(表7). <MOTION>は手間はかかるが, <ACTION>よりも洗練された体演技を実現できる. 現在, 30種類の代表的な仕草(お辞儀, さよなら, 歩く, など)を登録した.

<MOTION>の記述例を次に示す. 奈々は「みんな. こんにちは.」と話した後, お辞儀しながら「私は奈々です.」と喋る.</MOTION>は, お辞儀動作が終了するまで待つ.

<SPEECH>みなさん. こんにちは.</SPEECH>

<MOTION>おじぎ

<SPEECH">私は奈々です.</SPEECH>

</MOTION>

4.6 Cyber Personの設定

Cyber Personの3DCG形状データは, 専用のモデリングツールPiasArtist^{(24),(25)}で作成し, スクリプト言語PiasScript^{(24),(28)}で保存してCPSL Browserとともにインストールする. PiasScriptは, 形状情報だけではなく, 面の材質, テクスチャマッピング, 照明, 関節のリンク情報(シーングラフ)も定義する. したがって, CPSLでは, 上記の3DCG情報を定義する必要はない.<PERSON>は, Cyber Personを1人登場させるタグである. 必要なパラメータはOption Switchで与える(表8). これは状態変数ではないので省略不可能である. Cyber Personは何人でも登場させることができる.

次に, Cyber Personを1人登場させる例を示す.

<PERSON TDM_FILE="Nana.tdm"

NAME="奈々" SPEAKER="お姉さん (Toshiba)"

FEN_FILE="LyricDB.mdb"

BPN_FILE="ActionDB.mdb"

POSITION="0.0, -50.0, -2400.0"

ROTATION="0.0, 10.0, 0.0"

SCALE="1.0, 1.0, 1.1">

4.7 舞台の設定と小道具の配置

舞台の3DCG形状データもCyber Personと同じようにPiasArtistで作成し, PiasScriptで保存する. 舞台をすべて3DCGモデルで作成した場合は, 材質情報,

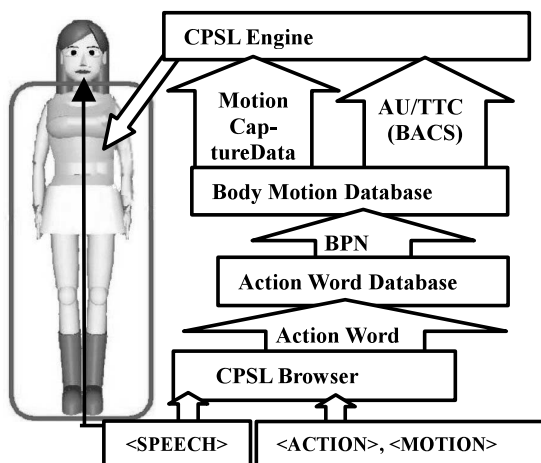


図7 体演技のデータフロー

Fig.7 Data flow of body action.

表6 BACS-Action Unit (AU) の例 (全部で 92 個)

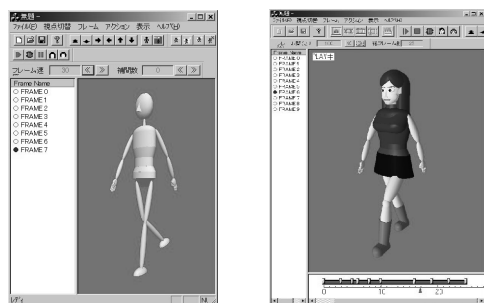
Table 6 Samples of BACS-Action Unit (AU), all is 92.

BACS-AU	Action Unit の動作
1180	腰を曲げて上体を前に倒す
1241	右手を前に出す。
1292	左腕を前に上げる。
1351	右手首を内側に曲げる
1381	右足を前に上げる。
1442	左の膝を折り曲げる
1470	両足首を外側に回す
1630	首を前に倒す



図8 BACSによる体演技の例

Fig. 8 Example of body action.



Key Frame Editor

Preview Motion

図9 BPN encoder.

Fig. 9 BPN encoder.

表7 <MOTION> の Option Switch

Table 7 Option Switches of <MOTION> .

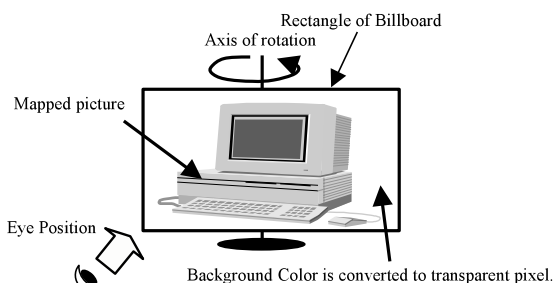
Switch	Value	Function
RATE	Integer	Specify the Playback Speed. Value is Intermediate Frames.(1)
REPEAT	Integer	Specify the Repetition Times.(1)

表8 <PERSON> の Option Switch

Table 8 Option Switches of <PERSON>.

Switch	Value	Function
TDM_FILE	Filename	File name of PiasScript
NAME	Naming	Naming of Cyber Person
SPEAKER	Name of Speaker	To select the Voice Synthesizer by using SPEECH API
FEN_FILE	Filename	Filename of Lyric Word Database
BPN_FILE	Filename	Filename of Action Word Database
POSITION	PX, PY, PZ	Coordinate of Located Position
ROTATION	RX, RY, RZ	Angles of Rotation
SCALE	SX, SY, SZ	Scaling Factor

Sideline of Rectangle and Axis of rotation are invisible.



Rotate to face the Camera automatically.

図10 ビルボード

Fig. 10 Billboard primitive.

テクスチャマッピング、照明情報などは、すべてこの段階で設定する。簡単な舞台を作る方法として、形状簡略化法の1つである Billboard Primitive^{17),23),24)}も使用可能である。これは、自動的にカメラの方向に回転するテクスチャマッピングされた板で、背景色が透明処理されるので、デジタルカメラで撮影した小道具などを簡単に設置できる(図10)。

<STAGE> は舞台を設定する。必要なパラメータは <PERSON> と同じように、Option Switchで指定する。舞台を設定する例を示す。

```
<STAGE TDM_FILE="Scene1.tdm"
  POSITION="300.0, -1370.0, -3000.0"
  ROTATION="0.0,0.0, 0.0"
  SCALE="1.0,2.0,1.0">
```


<STAGE>を実行すると、それまでの舞台を消去し、新たな舞台に切り替わる。したがって<STAGE>の記述回数が、シナリオ内のシーンの数となる。舞台の中に動く小道具（移動、回転）を配置する場合は、<PROPERTY>を用いる。<PROPERTY>は、<STAGE>で設定された舞台の中に小道具を配置し、<MOVE>でフレーム区間処理で動かす。記述例を次に示す。また、Option Switchを表9に示す。

```
<PROPERTY TDM_FILE="Mouse.tdm"
  NAME="Mouse"
  POSITION="300.0, -1370.0, -3000.0"
  ROTATION="0.0,0.0, 0.0"
  SCALE="1.0,2.0,1.0">
<MOVE NAME="Mouse"  NODE="Hand/Mouse"
  TRANSLATION="100.0,100.0,0.0"
  ROTATION="0.0,0.0, 0.0"  FRAMES="200">
</MOVE>
```

<PROPERTY>は何回でも記述できるが、舞台が<STAGE>で切り替わると、<PROPERTY>で作成された小道具はすべて消去される。

4.8 Cyber Personの歩行と回転

<WALK>、<SLIDE>、<JUMP>はCyber Personをフレーム区間処理で移動させる。この際、<MOTION>と同じようにBody TextにAction Wordを指定してMotion Capture Dataを用いて体演技させることができる。次に、歩く身振りをしながら歩行する例を示す。

```
<WALK DISTANCE="1000" NOTIFY="ABORT"
  REPEAT="100" FRAMES="50">歩く
  <SPEECH NOTIFY="SYNC" >私は歩くよ。
</SPEECH>
</WALK>
```

上記の例では、<WALK>に「歩く」というAction Wordを指定している。これは、1歩歩く体動作を記録したMotion Capture Dataである。体演技の繰り返し回数が100回という大きな値になっているが、同期モードがABORTなので、目的地に到着すると、</WALK>が歩行動作も強制終了する。<ROTX>、<ROTY> <ROTZ>は、Cyber Personをフレーム区間処理で回転させる。TURNが角度を示す。同じようにAction Wordを指定可能である。Option Switchを表10に示す。

4.9 その他のタグ

CPSLのその他の<TAG>を表11に示す。

<CAMERA>は、カメラを指定された場所へフレーム区間処理で少しずつ移動する（カメラワークと呼ぶ）。説明に必要な重要部位をズームアップする際に有効である。Option Switchを表12に示す。

<COM>は、シナリオ制作時に有効なタグで、囲まれ

表9 <PROPERTY>と<MOVE>のOption Switch

Table 9 Option Switches of <PROPERTY> and <MOVE> .

Switch	Value	Function
TDM FILE	Filename	File name of PiasScript
NAME	Naming	Naming of Property
NODE	Object Name	Moving Point
POSITION	PX,PY,PZ	Located Position
ROTATION	RX,RY,RZ	Angles of Rotation
SCALE	SX,SY,SZ	Scaling Factor
TRANSLATION	TX,TY,TZ	Moving Distance
FRAMES	Integer	Working Frames

表10 移動と回転のOption Switch

Table 10 Option Switches of translation and rotation.

Switch	Value	Function
DISTANCE	Integer	Distance of Movement
TURN	Integer	Angles of Rotation
REPEAT	Integer	Repetition Times.
FRAMES	Integer	Working Frames

表11 その他の<TAG>

Table 11 All the other <TAG>.

CPSL Tag	Body Text	Function
<COM>	Comment Text	Includes the Comment text.
<CAMERA>	None	Move the Viewing Camera
<LENS>	None	Setup the View Volume.
<MUSIC>	None	Play the specified MIDI file.
<SOUND>	None	Play the specified WAVE file.
<EXIT>	None	Stop to interpret the CPSL text and exit the application.
<MODE>	None	Set the state variables.
<TIMER>	None	Wait Specified Time.
<AU>	AU sequence	Apply AU sequence directly.

表12 <CAMERA>のOption Switches

Table 12 Option Switches of <CAMERA>.

Switch	Value	Function
POSITION	PX, PY, PZ	New Camera Coordinate
DIRECTION	DX, DY, DZ	New Aiming Position
ANGLES	AX, AY, AZ	New Vector to define the Top of Camera
FRAMES	Integer	Working Frames

たBlockをコメント化して、実行をスキップする。

<LENS>は、カメラの特性（ビューボリュームの形状）を設定する。<SOUND>は効果音を、<MUSIC>は音楽を演奏する。<MODE>は、状態変数の初期値を変更する。<EXIT>は特殊なタグで、Animationを終了して、CPSL Browserの実行も終了する。遠隔起動してAnimationを表示した後、自動的にウィンドウを消滅させるときに使用する。

5. CPSL Browser と Cyber Teaching Assistant

CPSL を用いて Cyber Teaching Assistant を実現するため、CPSL Browser を開発した。システム構成図と実行画面を図 11 に示す。CPSL Browser には、シナリオ制作を支援するツールバー（ボタン群）を装備した。CPSL テキストの入力、編集、修正を行うテキストエディタの呼び出しと即時再実行（デバッグ）を行うことができる。Cyber Teaching Assistant のシナリオは、次の 2 つのテーマについて制作を行った。

- (1) コンピュータリテラシ講座用ワードプロセッサによって、キーボードの機能キー操作、マウス操作、範囲指定、挿入、削除、複写、ツールバーによるファイル操作と印刷に関する質問回答シナリオ（26 個）を制作した。
- (2) プログラミング基礎・演習講座用主なキーワードの基礎文法、コンパイルエラーの対処法、デバッグの方法、関連するマウスやキー操作の質問回答シナリオ（32 個）を制作した。

シナリオは現在も制作続行中である。演習授業で運用した結果は、おおむね好評であった。当初、運用開始した時点では、4.7 節で述べた小道具をフレーム区間処理で動かす機能はなかったが、シナリオ制作者たちの強い要望で追加した。Cyber Person として女性 2 人（恵美と奈々）を開発した。通常は 1 人が回答者となるが、2 人の対談形式（1 人が質問者でもう 1 人が回答者）も試作した。CTA は講師が遠隔起動しなくても、受講者がサーバ内の CTA-Scenario ファイルをマウスでダブルクリックすれば閲覧できる。これを知った受講者は、質問キーを押さずに自分で CTA を呼び出す傾向を見せた。図 11 に示されているのは、コンピュータリテラシ講座で使われている映像である。

6. CTA-Animation 制作コストの比較実験

アニメーションの制作コストを実験で測定して客観的に比較するのは、文献 6) の中でも述べられているように困難である。しかし、CTA-Animation 制作という限定された条件の中で、2 章で述べた 3 つの技法のコスト差が、大差なのか僅差なのかを明確にすることは、1 つの指標として有用であろうと考え比較実験を行った。

6.1 制作コスト比較実験のテーマの設定

CTA-Animation には、次に示す特徴および必要条件がある。

- (1) 個々の映像の時間は比較的短い（1 分～3 分）

が、背景や小道具の入替え回数は比較的多く、通常 4～8 シーンで構成される。

- (2) 音声および字幕で提示する台詞が主体であり、これを映像の都合で変更できない。音声と字幕表示は同期していなければならない。
- (3) Cyber Person の指差す小道具や画面の位置が明確に認識できる程度の映像解像度が必要である（筆者らの経験では、最低 640×480 Pixel）。
- (4) サーバにデータベース化して蓄積しネットワークを通じてデマンド起動するので、映像（またはそのシナリオ）は、低容量であることが望ましい。
- (5) 映像の美的芸術性（アンチエイリアシング、影、人体の忠実性など）はあまり重要ではない。

このような観点から、制作コスト比較実験のテーマは、「文章編集時に、キーボードで文字列を範囲指定して削除する方法」のシナリオにした。約 2 分 10 秒の Animation で、6 Scene で構成される。

6.2 制作コスト比較実験の概要

実験は次の 5 チームに分かれて行った。

A1 チーム：CPSL 熟練者 2 人

A2 チーム：CPSL 未経験者 2 人

B1 チーム：CG プログラミング熟練者 2 人

B2 チーム：CG プログラミング初心者 2 人

C チーム：CG Animator 熟練者 2 人

B チームは、Microsoft Visual C++、SpeechSDK、OpenGL、PiasGL4²⁴⁾、CPDK²⁶⁾を用いて、C++ による OOP (Object Oriented Programming) で制作した。B2 チームは、C++ による OOP は熟知しているが、CG、音声合成、CPDK の機能は初めて使用する 2 人の Programmer である。C チームは、CG-Animator の Poser4J²⁹⁾、DV-Editor の Premiere³⁰⁾ を使用して、2 人の共同作業で制作した。制作コストは 2 人のコストの合計値とした。この 2 人は、これらのツール類を駆使して 45 分の映画を自主制作した経験を持つ Expert である。B チームと C チームの作業は、非専門家には無理と判断した。各チームにストップウォッチと工程時間記録表を渡し、作業工程ごとの時間を計測してもらった。また、必要なソフトウェアのインストール、台詞原稿の入力、背景画像や小道具画像の準備などの準備時間の個人差を消去するため、各チームに素材キットを作成して配布した。

6.3 制作コストの分析

各チームの平均制作コストを、制作コストの昇順に工程積み上げグラフで図 12 に示す。A1 チームと A2 チームの制作コストに大きな差は現れなかった。A1

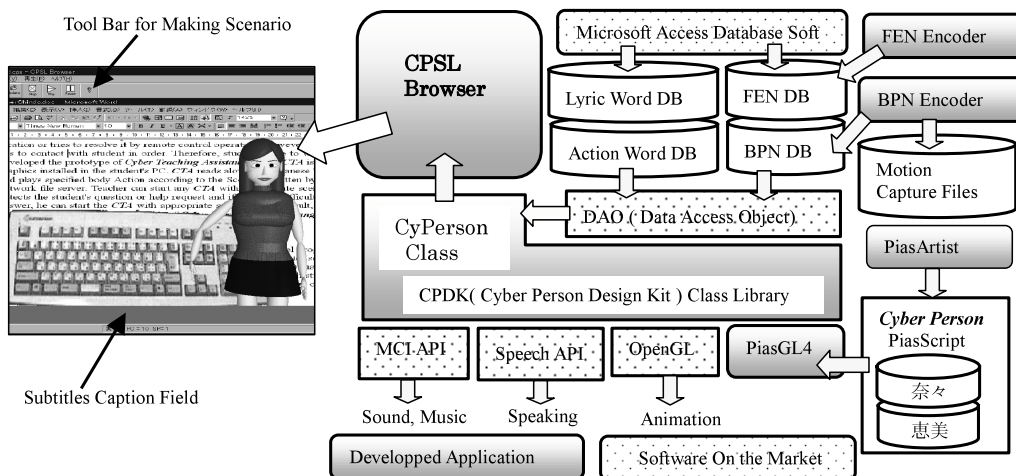


図 11 CPSL Browser のシステム構成

Fig. 11 CPSL browser.

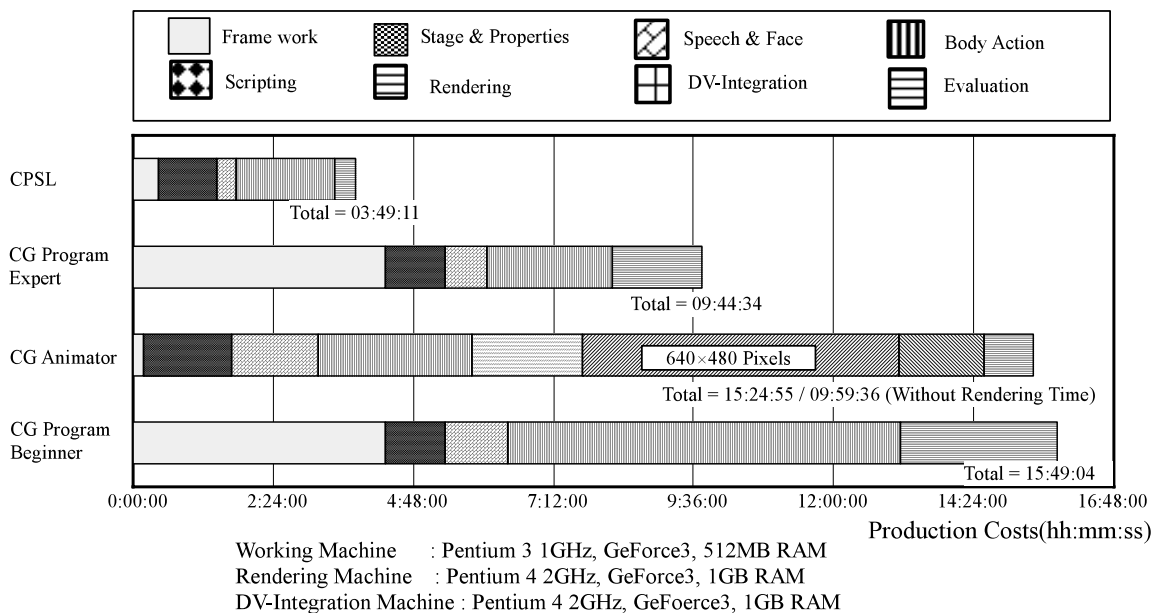


図 12 制作コストの工程別の比較

Fig. 12 Comparison of production costs.

チームの 1 人は BPN-Encoder で演技を制作し、残り 3 人は BACS 符号を用いたが、制作コストに差異は見られなかった。

これに対して、B チームの制作コストは大きく分散した。図 13 に B チームの各々の作業工程コストを示す。B2 チームは OOP には熟練しているの、初段階での制作コストに差異は見られなかった。しかし、体演技を制作する部分、統合評価・修正作業に大きな制作コストの差異が見られた。CG や音声合成の API 関数は、指定パラメータの数が多く、デバッグ作業に

おける追跡・解析・修正コスト増大の要因になっていると思われる。

C チームの制作コストは、A チームと比較するとずっと大きい。ある程度の解像度を求めるため、レンダリング時間が長いことも原因の 1 つであるが、レンダリング時間を差し引いても、A チームの 3 倍程度の制作コストであった。制作者がコストが大きい理由としてあげたのは、「台詞である音声と、顔表情、体の演技の時間的同期をとるのが難しい」という点であった。CG-Animator や DV-Editor では、台詞音声タイム

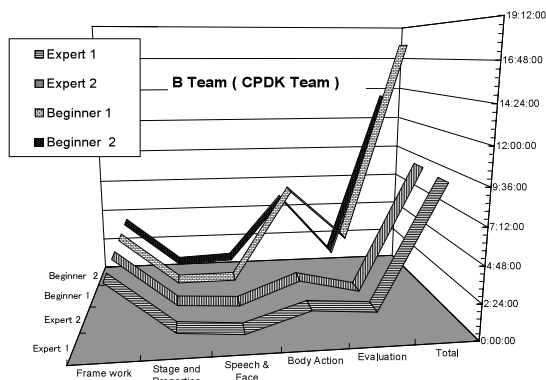


図 13 B チームの各々の制作コスト
Fig. 13 Production costs of B-team.

チャート上での波形となり、「台詞文字が見えない」ことが原因であった。

CTA-Animation では、体の演技で手先が指し示す位置などに正確さが求められる。この位置合わせのために、簡易レンダリング作業を繰り返す必要があるが、台詞文字が見えないので、振り付けと音声と同期させる試行錯誤的調整作業にコストがかかった。

同様に、音声に合わせて顔表情や唇の動きを振り付ける作業、字幕を作成して音声に合わせて映像に貼り付けていく作業などもすべて手作業になるので、制作コストが大きくなった。これは、CG-Animator がまず演技を作り、その後に音声をアフレコする、という作業工程を前提にしているためと思われる。また、部分的な修正（字幕の誤字脱字の修正や、音声の間違ひの修正など）が全体コストに及ぼす影響が大きい。これは、シーン単位で再度レンダリングする必要があるためである。結果として、制作コストは小さい順に、

A1, A2 チーム < B1 チーム < C チーム < B2 チーム となった。

6.4 シナリオデータベースとしての評価

制作した CTA-Animation は、約 2 分 10 秒である。これを、データベースとして格納する際の容量を比較検討した。CPSL で制作した Scenario は、ハイパーテキスト、舞台や小道具データ、マッピングする画像などを含めて 541.57 KB であった。

CPDK Program は、人体データや動作のための共用コンポーネントが再生 PC 側にインストールされているならば、実行プログラムと画像や小道具データだけとなるので、1.04 MB で記録できた。これらをネットワークサーバ内に配置してデマンド起動することも可能と思われる。C チームの Movie を各種解像度でレンダリングして、いろいろな圧縮処理で圧縮し

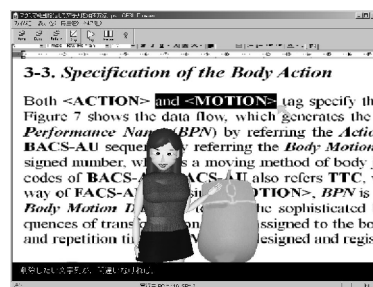


図 14 第 2 テーマのシナリオ
Fig. 14 2nd theme of CTA-animation.

たところ、19.2 MB～221 MB の範囲であった。最低容量の MPEG-1 (352 × 298) の画像では、背景画像内の文字が乱れて、画像拡大しても判読困難な部分があった。CTA-Animation として使用可能だったのは、MPEG-4 (640 × 480) で音声をモノラルにしたもので容量は 32.6 MB であった。

これに対し、CPSL シナリオの容量は画像や小道具データも含めてわずか 550 KB 程度、と Movie の 2% にも満たない。解像度も実行時に決定されるので、解像度不足による判読困難も発生しなかった。

6.5 同様のテーマで再制作した場合

同様な制作を繰り返した場合、前回の経験や資産が継承される度合いを測定するため、第 1 テーマ制作後、類似の第 2 テーマを制作した。テーマは、「文章編集時にマウスで文字列を範囲指定して削除する」解説である。テーマ 1 と異なるのは、小道具マウスを動かす課題が入っている点である（図 14）。

この実験は、A チームと B チームのみで行った。A チームの作業結果を図 15 に示す。CPSL は、第 1 テーマのシーンを簡単に流用できるため、フレームワークの制作時間は 12% に低減した。また、演技もテーマ 1 を流用可能な部分が多いので、時間は平均で第 1 テーマの 63% に高速化した。A1 チームと A2 チームの間の差異はほとんどみられなかった。

B チームの制作コストの比較結果を図 16、図 17 に示す。B1 チームの制作コストの改善率は 78% であったが、A チームの約 3 倍のコストを要した。B2 チームの制作コストは 56% と高速化した。B2 チームが、CG、音声合成、CPDK などの知識を習得した結果と思われる。

6.6 総合評価

市販の CG-Animator を用いて制作した映像の人体モデルの美的洗練度は秀逸であるが、CTA-Animation を制作するうえではあまり効果的ではなかった。主な理由は、

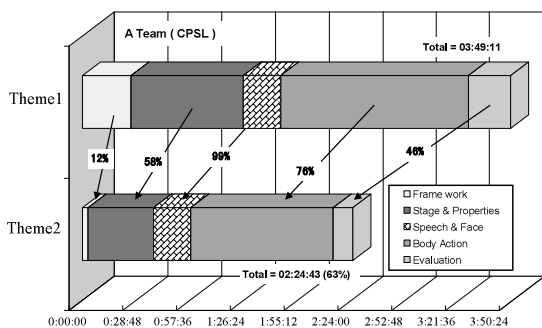


図 15 A チーム (CPSL) の制作コスト比較

Fig. 15 Comparison of A-team production cost.

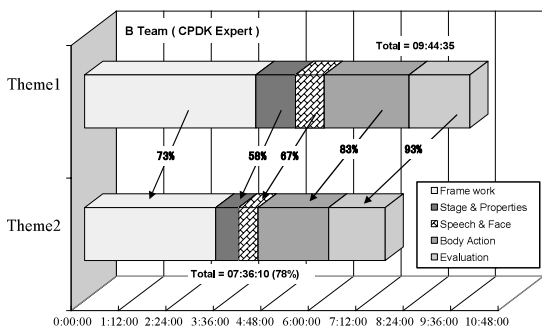


図 16 B1 チーム (CPDK) の制作コスト比較

Fig. 16 Comparison of B1-team production cost.

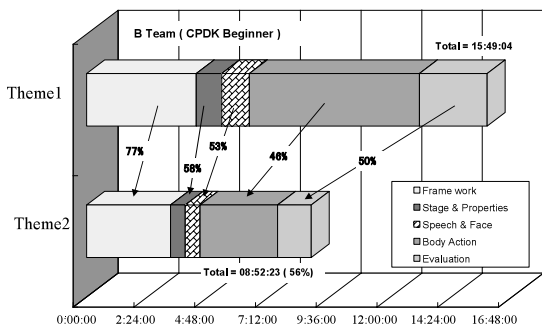


図 17 B2 チーム (CPDK) の制作コスト比較

Fig. 17 Comparison of B2-team production cost.

- (1) 解像度を上げると、制作コストが増大するだけでなく、データベースの容量も増大する。
- (2) 解像度が低いと、背景の文字や小さなツールバーボタンなどがよく見えない。
- (3) 台詞音声と、表情、体演技、字幕を同期させる作業が、台詞文字が見えないためやりにくい。
- (4) 使用するツールに熟練した専門家でないと、制作作業は難しい。
- (5) 人体モデルに自律機能（たとえば瞬きや体の自然な揺れ）がなく、手作業で作成しないと組み

込めないで、モデルが美しいわりには予想よりも臨場感がない。

プログラミングで制作する場合には、洗練された SDK を用いれば、自由度の高い CG-Animation の制作が可能である。しかし、高度なプログラミング技術と CG や音声合成に関する専門知識が必要である。非専門家が手軽に CTA-Animation をプログラムで制作するのは困難である。スクリプト型の言語でも、コマンド型の命令と多数のパラメータの指定を必要とするものは、同様な傾向を示すものと思われる。

これに対し、CPSL で制作した場合には、制作コストは最も小さく、専門家と非専門家の間に大きなコスト差も現れなかった。しかも同種のテーマの制作を続ける際、前回の経験や資産が継承される度合いも高いことを実証した。これらの実験結果から、ハイパーテキスト型の CPSL が CTA-Animation の制作に最も適していると結論した。

7. おわりに

本論文では、情報処理演習室で講師の代わりに質問者に回答する Cyber Teaching Assistant システムを提案した。さらにこの Animation のシナリオを非専門家でも簡単に制作できるハイパーテキスト言語 CPSL の詳細を述べた。また、従来の技法との制作コスト比較実験を行い、市販の CG-Animator やプログラム言語で制作するよりも低コストで制作できることを示した。

CPSL は、質問に対する回答 Animation 用に設計したが、高齢者向けの家庭電化製品の操作解説や、小中学校での交通安全やスポーツのルール解説などへの利用を考えている。このためマウス操作に対するイベント応答や、音声認識による対話機能などを簡単に記述できるように言語仕様を拡張していく予定である。さらに、植物や動く動物などの機能を追加し、童話や昔話を上演するサイバー人形劇場として、幼稚園や小学校低学年での教材として利用できないか検討中である。

参考文献

- 1) Phillips, C. and Badler, N.I.: Jack: A toolkit for manipulating articulated figures, *ACM/SIGGRAPH Symposium on User Interface Software* (1988).
- 2) <http://www.cis.upenn.edu/~hms/jack.html>
- 3) <http://ecust.isid.co.jp/public/products/jack/>
- 4) 長谷川修, 坂上勝彦, 速水 悟: 実世界視覚情報を対話的に学習・管理する人間型ソフトウェア

- ロボット, 信学論 (D-II), Vol.J82-D-II, No.10, pp.1666-1674 (1999).
- 5) 牛田博英, 平山祐司, 中嶋 宏: 自律的行動決定モデルに基づくインタフェースエージェント, 信学論 (D-II), Vol.J82-D-II, No.10, pp.1655-1665 (1999).
- 6) 筒井孝之, 石塚 満: キャラクタエージェント制御機能を有するマルチモーダル・プレゼンテーション記述言語 MPML, 情報処理学会論文誌, Vol.41, No.4, pp.1124-1133 (2000).
- 7) 道家, 林, 牧野: TVML を用いた番組情報からのニュース番組自動生成, 映像情報メディア学会誌, No.7, pp.1097-1103 (2000).
- 8) Hayashi, M., Ueda, H., Kurihara, T. and Yasumura, M.: TVML (TV program Making Language) — Automatic TV Program Generation from Text-based Script, *Proc. Imagina'99* (1999).
- 9) Okamoto, T., Matsui, T., Inoue, H. and Cristea, A.: A Distance-Education Self-Learning Support System based on a VOD Server, *Proc. International Workshop on Advanced Learning Technologies (IWALT2000)*, pp.71-72, IEEE Computer Society Press (2000).
- 10) Cupp, E.L., Danchak, M.M., Foster, K.J., Johnson, A.D., Kim, C.K. and Sarlin, D.D.: The RSVP Uni-browser: Bringing Humanness and Interactivity to e-Learning, *Proc. IEEE International Conference on Advanced Learning Technology (ICALT2001)*, pp.67-69, IEEE Computer Society Press (2001).
- 11) Graesser, A.C. and Hu, X.: Teaching with the Help of Talking Heads, *Proc. IEEE International Conference on Advanced Learning Technology (ICALT2001)*, pp.460-461, IEEE Computer Society Press (2001).
- 12) Baylor, A.L.: Cognitive Requirements for Agent Based Learning Environments, *Proc. IEEE International Conference on Advanced Learning Technology (ICALT2001)*, pp.462-463, IEEE Computer Society Press (2001).
- 13) Shindo, Y. and Matsuda, H.: Prototype of Cyber Teaching Assistant, *Proc. IEEE International Conference on Advanced Learning Technology (ICALT2001)*, pp.70-73, IEEE Computer Society Press (2001).
- 14) Shindo, Y. and Matsuda, H.: Design and Implementation of Scenario Language for Cyber Teaching Assistant, *Proc. International Conference on Computers in Education (ICCE2001)*, Vol.2, pp.643-650 (Nov. 2001).
- 15) 山室勝義, 新藤義昭: 仮想俳優の表情と演技を記述するシナリオ型言語, 情報処理学会第 63 回全国大会講演論文集, Vol.4, pp.59-60 (2001).
- 16) 丸山敬三, 新藤義昭: 3DCG Virtual Actor の演技と表情を記述するシナリオ型記述言語, 第 61 回情報処理学会全国大会講演論文集, Vol.4, pp.147-148 (2000).
- 17) 新藤義昭, 片山滋友, 坂本康治, 松田郁夫: 形状簡略化による 3 次元形状モデルの描画速度制御の一技法, 情報処理学会論文誌, Vol.36, No.10, pp.2452-2463 (1995).
- 18) 山地秀美, 新藤義昭: 3 次元仮想空間におけるオブジェクト間の衝突検出の一技法, 第 59 回情報処理学会全国大会講演論文集, Vol.4, pp.91-92 (1999).
- 19) Shindo, Y.: Programming Education Based on Computer Graphics Animation, *Proc. International Workshop on Advanced Learning Technologies (IWALT2000)*, pp.292-293, IEEE Computer Society Press (2000).
- 20) Matsuda, H., Shindo, Y. and Mukuda, M.: Effect of Using Computer Graphics Animation in Programming Education, *Proc. IEEE International Conference on Advanced Learning Technology (ICALT2001)*, pp.164-165, IEEE Computer Society Press (2001).
- 21) 松田 洋, 新藤義昭, 椋田 實: プログラミング演習教材ソフトウェア WinTK, 私立大学情報教育協会情報教育方法研究論文誌, Vol.1, No.1, pp.31-36 (1998).
- 22) Ekman, P. and Friesen, W.V.: *Manual for the Facial Action Coding System*, Consulting Psychologists Press (1978).
- 23) OpenGL Architecture Review Board: *The official Guide to Learning OpenGL*, Ver.1.1, Addison Wesley Publishers (1997).
- 24) 新藤義昭, 阿部正平: OpenGL リアルタイム 3D プログラミング, (株) 秀和システム (2000).
- 25) 新藤義昭, 山地秀美: パーチャルリアリティ・プログラミング, (株) NEC クリエイティブ (1997).
- 26) 丸山敬三, 新藤義昭: 表情生成可能な擬人化エージェントの構築支援システム, 第 59 回情報処理学会全国大会講演論文集, Vol.4, pp.59-60 (1999).
- 27) 大塚, 新藤, 松田, 樺澤: 仮想骨格モデルによる顔の 3 次元 CG モデルの表情生成, 第 57 回情報処理学会全国大会講演論文集, Vol.4, pp.81-82 (1998).
- 28) 岩谷, 新藤, 松田: 動きのある仮想空間をインターネット上に構築する記述言語 PiasScript, 情報処理学会第 53 回全国大会論文集, Vol.4, p.197, 198 (1996).
- 29) Poser4 ユーザーズガイド, Curious Labs.
- 30) Adobe Premiere ユーザガイド, Adobe Systems Inc.
- 31) Clark, D. and Stupple, S.J.(Eds.): *Developing for Microsoft Agent*, Microsoft Press (1998).

(平成 14 年 1 月 14 日受付)
(平成 14 年 6 月 4 日採録)



新藤 義昭 (正会員)

昭和 29 年生。昭和 51 年電気通信大学卒業。同年富士通(株)入社。平成 5 年より日本工業大学工学部電気電子工学科専任講師。平成 7 年同大学情報工学科専任講師。平成 10 年情報工学科助教授。画像情報工学、コンピュータグラフィックス、バーチャルリアリティの研究に従事。特に、これらを活用した新たな情報メディアの開発や、情報処理演習における教授法や講師支援システムの開発に力を入れている。著書「OpenGL リアルタイム 3D プログラミング」(秀和システム)、「バーチャルリアリティプログラミング」(NEC クリエイティブ)等。電子情報通信学会会員。



松田 洋 (正会員)

平成 7 年北陸先端科学技術大学院大学情報科学研究科情報システム学専攻博士前期課程修了。平成 7 年より日本工業大学情報技術センター助手。プログラミング教育支援システム、教材作成支援システム等の研究に従事。平成 10 年私立大学情報教育協会奨励賞。電子情報通信学会、教育システム情報学会、日本教育工学会各会員。



鈴木 誠史

昭和 30 年東京工業大学工学部卒業。郵政省電波研究所(現独立行政法人通信総合研究所)入所。以来音声情報処理、音声通信、通信システム等の研究に従事。昭和 63 年郵政省通信総合研究所長。平成元年埼玉大学工学部教授。平成 10 年より日本工業大学教授。平成 13 年より通信・放送機構、本庄情報通信研究開発支援センター長兼務。前島賞(昭和 61 年)等。工学博士。電子情報通信学会フェロー、映像情報メディア学会、日本音響学会、アメリカ音響学会、IEEE 会員。