

紙芝居言語 CSL

疑似ハイパーテキストシステムの構築

浦野収司 ・ 藤原敏樹
(株) ハイテックラボジャパン

CDROMなどの二次記憶装置上の画像データ、文字データ、音声データを紙芝居的に構成、表現する疑似ハイパーテキストシステムを試作した。当システムは映画などの制作段階で用いられているストーリーボード(シーン毎の仕上り画面を想定した紙芝居)に想を得、シーン毎のシナリオを記述する方式を採用している。この核となるコンピュータ言語 CSL および仮想機械 RUNCSL を SUN3、PC9801 上に移植し、制作工数の低減を達成したので報告する。

"CSL : THE STORYBOARD LANGUAGE" PSUDO HYPERTEXT SYSTEM

by Shuji URANO, Toshiki FUJIWARA
(HighTech Lab. Japan inc.
1-4-1 Higashiyama, Meguro-ku, Tokyo, 153 Japan)

We have developed a psudo Hypertext system which is capable of constructing graphics, text and audio data for secondary strage such as CDROM. In this system, the concept of "Storyboard," which is often used in film and video making processes, has been imported to multimedia production, where scenarios can be written in the scripting language for each scene. The system consists of a scenario language, called "CSL," and a virtual machine, called "RUNCSL," which has been implemented on SUN3 computer and NEC PC9801 personal computer. The system is considered to be a strong tool, which increases productivity and efficiency in interactive multimedia production.

はじめに

近年CDROMに代表される低価格大容量の記録媒体が開発され、辞書や特許情報などの応用が模索されている。当社でも、太陽系の天文図鑑などのCDROM化を行ってきた。図鑑のCDROM化にあたっては、画像、文字データに加えて、CD (ROM) が本来有している音声データもナレーションなどに利用している。

対象とするコンピュータはNECのPC9801で、ポインティングデバイスとしてマウスを想定している。マウス操作によって図鑑の検索を行なう。このマウス操作とこれらの各種データの再生表示を表現するためのシナリオ言語CSLを規定し、これを解釈実行する仮想機械RUNCSLをPC9801上に移植した。CDROMのデザイナーはこのCSLを用いてシナリオを記述する。CDROMのスタンプを行なう前に各種データ及びシナリオをシミュレートするためにSUN3上にも仮想機械RUNCSLを移植した。

ハイパーテキスト

コンピュータ上で、画像、文字、音声などの複数の形態の情報をテキストの延長として扱う概念が、1970年代にテッド・ネルソンによってハイパーテキストとして提唱された。例としてはAppleのMachintosh上のHyperCard、OWL InternationalのMS-Windows上のGuide、東大の坂村健助教授がBTRON上に提案した実身/仮身モデルがある。これらはいずれもマルチタスクが可能（あるタスクの実行を中断して他のタスクを生成実行する程度のマルチタスク）であり、ハイパーテキストの構築すなわちオーサリング機能もこのマルチタスクによって実現している。

我々の移植対象ではこのマルチタスクが困難なため、オーサリング機能を切り離し、

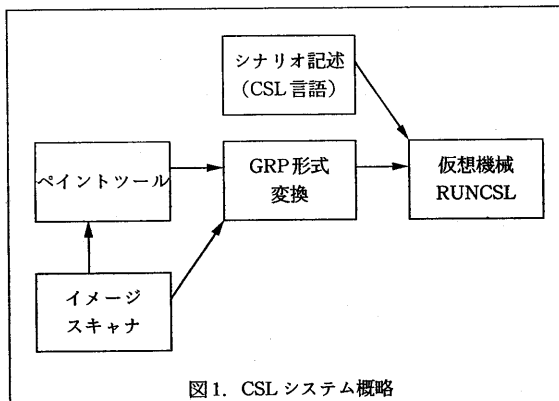


図1. CSLシステム概略

疑似的なハイパーテキストの実現を目指した。また、HyperCardが情報カードから発展したのに対し、我々は映画などの映像制作からのアプローチをとった。このため、オーサリング段階でシーンとシナリオという概念を導入している。

シーンとシナリオ

シーンとは、ある連続した時間または状況を包括する場のことであり、シナリオとはその状況の記述である。映画「ブレードランナー」で例を挙げると、

2020年ロサンゼルス場の場末のシーン：

うっとうしく雨の降り注ぐ夜、元警官のデッカーが本そばの屋台でうどんを食べている。

いきなりデッカーの肩を叩くものがあり、デッカーは振り向く。

「デッカーだな。ボスがお呼びだ。」、黒いコートを着た男がロシア語なまりで用件をきりだした。

「ほっといてくれ！俺はもうブレードランナーじゃない」

デッカーがそう答えてうどんを食べようとすると、男は無言を言わせぬ態度で強制した。

デッカーがスピナーに乗り込むと、非常灯を回転させながらスピナーは上昇を始めた。

ロサンゼルス上空の飛行シーンに移る。

以上のようなシーンを個別に撮影し、編集段階で繋げて一本の映画が出来上がる。各シーンを複数の制作チームで並行して制作すれば、制作日数が短縮出来る。

我々も初期の太陽系の天文図鑑のCDROM化において、制作日数の短縮が非常に大きな問題となった。この経験を踏まえて映画制作のシーン割りを導入したハイパーテキストシステムの開発を行なった。

映画制作では脚本・演出家がシナリオを書き、俳優がシナリオを解釈して演技する。そして観客が編集された作品を觀賞する。これに対してハイパーテキストでは演出・制作者がシナリオを記述し、コンピュータがシナリオを解釈して画像や文字、音声などの情報を提示し、観客（読者）は自分の興味に従って対話的に自分自身のために演出する。例えば先の「ブレードランナー」をのシーンでデッカーに反抗の選択枝を与えて、異なるシーンへ移ることもできるだろう。

CSL,RUNCSLの実現方法

HyperCardやBTRONの実身／仮身モデルではオブジェクト指向になっている。特にBTRONの付箋機能はオブジェクト指向そのものである。我々の設計段階でも、シナリオを解釈して演技するのだからやはりActor言語が適しているのではといった冗談めいた議論もあった。しかし、現在のコンピュータ、とくに移植対象となるマイクロコンピュータでは処理能力や主記憶容量が著しく不足しているため、オーソドックスなインタプリタ方式の仮想機械に決定した。

シナリオ記述言語 CSLの名前の由来はCDROM Scenario Languageである。またこれを解釈実行する仮想機械 RUNCSLはSUN3とPC9801の両方に移植した。

CSLのシナリオ記述例として、先の「ブレードランナー」をアドベンチャーゲームに仕立ててみる。

LosAngels2020 :

```
loadBack  Yatai (屋台の背景を表示)
loadGraph Deckert counter (デッカートを屋台のカウンターに表示)
loadGraph Police entrance (黒いコートの男が屋台の入口に現われる)
loadSound Dialogue (科白の再生)
waitEvent (イベントハンドラ)
    obey) (イベント名)
        moveTo  BirdViewOfLosAngels (ロサンゼルス上空のシーンに移る)
    Don'tObey)
        moveTo  YetAnotherScene
endEvent
```

以上のシナリオにおける要点は、loadBack,loadGraph,及びイベントハンドラのwaitEvent～endEventである。loadBack,loadGraphで表示する画像はGRP形式ファイルと呼ぶファイルを読み込み、ビットマップイメージに展開している。

GRP形式ファイルは以下の3種の部分から構成される。

1. ヘッダ

画像データのピクセル単位での巾・高さ・プレーン数及び画像データのフォーマット等

2. ボタン・スロット情報

ボタンはポインティングデバイスのセンス領域とイベント名
スロットは部分画面描画領域とスロット名

3. 画像データ

各種のフォーマットの画像データ。ビットマップイメージとバイトエンコード圧縮イメージなどのフォーマットがある。

シーンLosAngels2020では、loadBack Yataiで屋台の背景画像を表示するとともに、以後引用する部分画面描画領域(スロット) counter,entrance,spinnerをRUNCSLの内部情報として読み込む。また、loadGraphでボタン(センス領域)と、センス(マウスボタンのクリック)時に発生するイベント名も読み込む。

waitEvent～endEventではイベント待ちとイベント発生時の動作を記述している。

まとめ

シーンとシナリオをベースにした疑似ハイパーテキストシステムを開発した。当システムによる最初の製品として月刊誌 PENTHOUSE のフロッピー・ディスク版を制作した結果、画像・文字データなど合計約 2.6M バイト（最終的には画像圧縮によって 5 インチ 2HD のフロッピー・ディスク 1 枚に収めた）のオーサリングを行ない、いわゆる修羅場なしで納品することができた。制作工数はハイパーテキストシステムを用いない従来の方法に比べて半減した。またオーサリングに関しては、特にプログラミングの専門知識を必要としないため、人件費などの経済効果は大である。

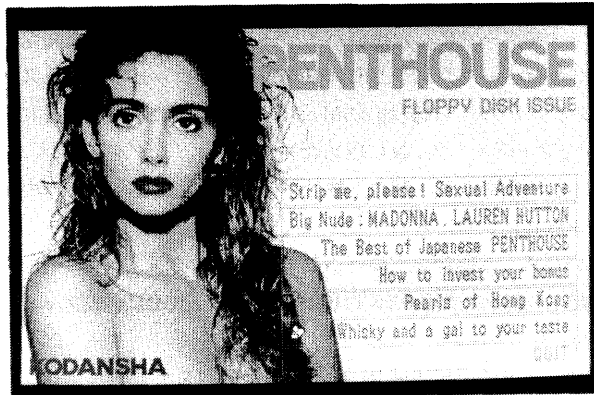
図 2 にフロッピー・ディスク版 PENTHOUSE の画面と GRP 形式ファイル、およびシナリオの例を示す。

また図 3 に SUN3 上での仮想機械 RUNCSL の実行例を示す。

参考資料

フィリップ・K・ディック原作 リドリー・スコット監督「ブレードランナー」ラッドカンパニー制作 ワーナーブラザース配給（1982）。

「フロッピー・ディスク版 PENTHOUSE」講談社刊（1988）。



```
pent_1.grp(graphics) (640 x 400 x 3) in 96000 byte(s), with 8 slot(s)
No Type Name Lef Top Wid Hei
0 Button panic_a 259 19 366 51
0 Button select1 290 167 334 30
0 Button select2 289 199 335 30
0 Button select3 289 231 335 30
0 Button select4 289 263 335 30
0 Button select5 289 295 335 30
0 Button select6 289 327 335 30
0 Button quit 289 359 335 30
```

```
PENT_1:
loadBack C_RIGHT
loadGraph PENT_1 fullscreen
loop:
waitEvent
select1)    changeDir "../game"
select2)    changeDir "../nudo"
select3)    changeDir "../his"
select4)    changeDir "../kab"
select5)    changeDir "../hongkong"
select6)    changeDir "../nikka"
quit)       changeDir "../end"
panic_a)    panic
            moveTo loop
endEvent
```

図2 フロッピー・ディスク版 PENTHOUSE 実行画面例 (上),
GRP 形式ファイルの内容 (中), シナリオ例 (下)

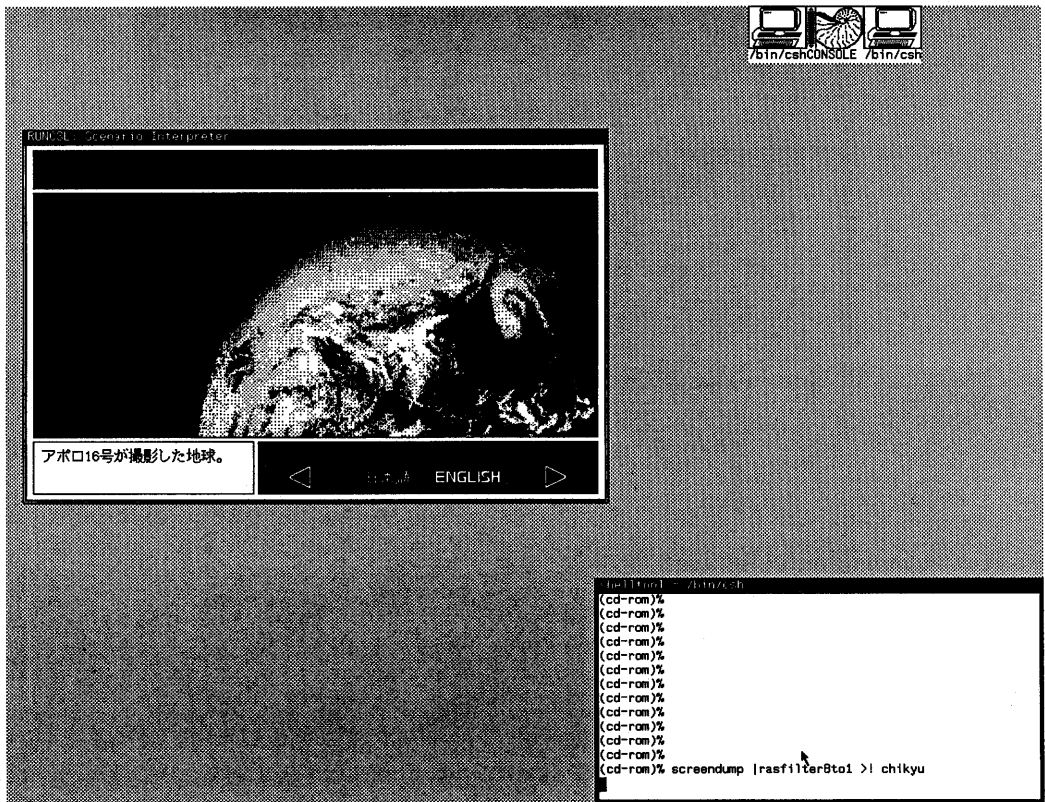


図3 SUN3上での仮想機械実行例