

# Java による 分散画像データベースシステムの研究と開発

柳 正 栄

柏崎情報開発学院

yanagi@ksc.kashiwazaki.niigata.jp

ネットワークに分散した画像データ情報を一括管理することによって、画像データの再利用を図るシステムを開発した。本システムは、システム利用によるユーザ負担を回避するため、GUIによるプレビュー機能、キーワード検索、外部アプリケーション連携機能を提供している。これらの機能は、WebとJavaの連携によって実現されている。

## A Study and Development for Distributed Image Data Base with Java

Masae Yanagi

Kashiwazaki Software College

This article reports the system which provides the utilization of image data distributed in a network. For improving the usability, this system provides the GUI functions ( preview, keyword searching, contacting with external application ). These functions are implemented by the connection with Web and Java.

### 1. はじめに

CG制作を支援するため、画像データの蓄積管理と再利用を図るシステムを開発した。本システムは、ネットワークに分散した画像データの所在やそれを作成したアプリケーション、プレビュー画像の所在等の情報をDB(データベース)に一括管理する。CG制作者は既存の画像のプレビューを覗いたり、キーワード検索等

によって、再利用したい画像の所在を知ることができる。さらに、画像の修正や再利用も個々のアプリケーションの違いを意識することなく、マウス操作だけでそのアプリケーションを起動することができる。

本システム設計のポイントは、CG制作者に本システム利用のための負荷を与えることなく、簡便に効率良く本来のCG制作に専念できる環境を提供することにある。

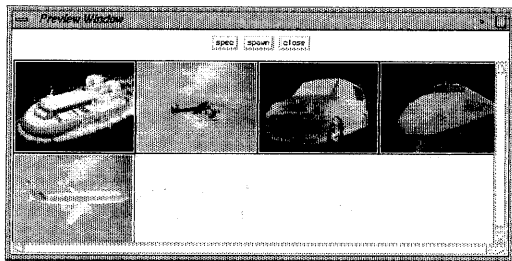


図 1 プレビュー表示

さらに、本システムは、インターネット環境も視野に入れ、CAD や地図などのマルチメディアデータ管理システムへの適用を考慮しデザインした。このため、本システム開発にあたっては、開発言語として Java 言語を採用した。これは、Java のもつマルチプラットフォーム性、オブジェクト指向プログラミング、AWT (Abstract Window Toolkit) 等により、大幅な開発工数の削減が期待できると考えたからである。[1]-[3]

本稿では、本システムの概要と開発経験を紹介するとともに、今後の課題について述べる。

## 2. 問題点とアプローチ

### 2. 1 問題点

多大な工数をかけ作成した画像データも、その CG 制作者が異動するとその所在すらわからず利用されなくなってしまう。ドキュメントを残すなどの対処方法はあるものの、イメージで管理できないなど決して適切な方法とはいえない。この解決策として、つぎの2つの方策がある。

- ① 画像データとその属性情報を DB に収集/格納し、これにより一元管理を行なう。
  - ② 画像データをネットワークに分散させたまま、画像に関する属性情報のみ一元管理する。
- ①の画像データを1ヶ所に収集管理する方法では、サーバの高機能/大容量化が必要であ

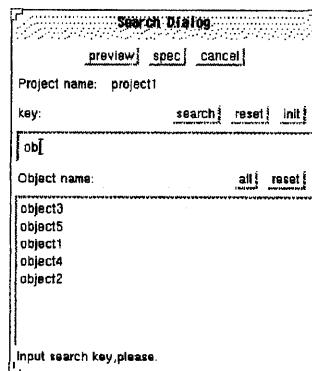


図 2 キーワード検索

るだけでなく、画像データをサーバへ収集/格納するための作業が別途必要とされる。

また、この方法では、収集された画像データの修正/再利用にあたって、CG 制作者は必要とする画像データをローカルにコピーするなどしてサーバから取得し、さらにそれに対応する外部アプリケーションを手動で起動するなど、CG 制作者にとってかならずしも快適な操作環境とはいえない。したがって、②の方策を採用する必要がある。

### 2. 2 アプローチ

もし、画像データを集中させる必要が無く、複数のホストに分散させたままですめば、CG 制作者にとって特別な手続きも必要無く、生産性を阻害することもない。

画像データの再利用を考えても、どこにどのような画像データがあるかを簡単に探し出せ、それを利用することも特にキーボードを叩くことなくマウス操作で可能であれば、快適であろう。

また、新しく画像を制作する場合に既存の画像のもとになっている部品も再利用できればさらに高い生産性が期待できる。そこで、画像がどのような画像から構成されているかという階層的管理も必要と考えた。

そこで、ネットワーク機能を活用することで画像データを分散させながらも一括管理する枠

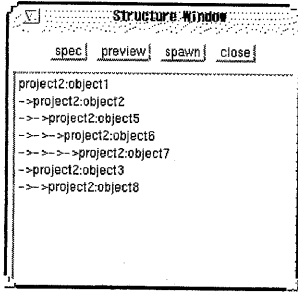


図 3 画像関係情報表示

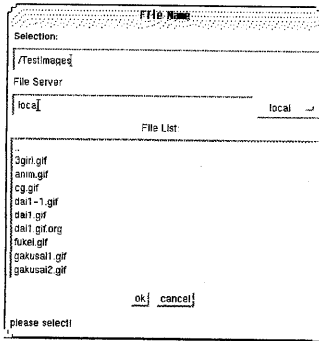


図 4 ファイル選択ダイアログ

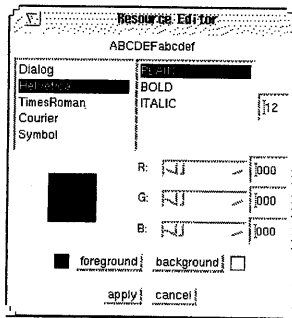


図 5 システムプロパティダイアログ

組みを提供するとともに、次の機能を実現することで CG 制作者への本システム利用による負担をかけないようなシステムを設計開発することにした。

- Web との連携
- GUI による操作環境の提供

## 2. 2. 1 Web との連携

Web との連携を図った理由はつぎのとおりである。

- 新規サーバ開発の工数を削減できる。
- 新規サーバの配布、運用のための工数を削減できる。
- Web 技術をベースにすることにより、システムの拡張性、長寿命化が期待できる。

Web 技術は進歩し続けている。この Web 技術発展に添う形でシステムを改良していくことで、そのメリットを享受できるだけでなく、システム寿命の長期化も期待できる。

## 2. 2. 2 GUI による操作環境

本システムは、次の機能を GUI で実現している。

- プレビュー表示
- キーワード検索
- 画像関係情報表示
- 各種ダイアログ
- 外部アプリケーション連携

プレビュー表示 (図 1) は、画像データのイメージ検索を可能とする。プレビュー画像としては静止画だけでなくアニメーション表示も可能で、さまざまなピクセルサイズを扱える。プレビュー表示から spec ボタンで画像オブジェクト情報を眺めたり、spawn ボタンで外部アプリケーションが起動できる。

キーワード検索 (図 2) は、個々の画像データにつけた説明文を対象にキーワード検索ができる。ユーザは文章形式でその画像に関するメモを残しておけば、そのメモからキーワード検索することができる。キーワード検索からもプレビュー表示やオブジェクト属性表示、外部アプリケーション起動ができる。

画像関係情報表示 (図 3) は、画像データ間の「全体一部分」関係を再帰的に検索し、ネストされた「全体一部分」関係を表示する。ユーザは任意階層のオブジェクト属性やプレビュー表示、外部アプリケーション起動が行える。

ダイアログには、ファイル選択ダイアログ(図 4)、システムプロパティダイアログ(図 5)などがあり、CG 制作者の画像データに関する情報入力作業の簡素化と視認性の向上を図る。

外部アプリケーション連携は、マウス操作で画像オブジェクトを選ぶだけで、それに対応する外部アプリケーションを起動するものである。これにより、ユーザはその画像がどのようなアプリケーションで作成されたかを全く意識する必要はなく、単に起動したい画像オブジェクトを選ぶだけで良い。

### 3. 画像データのモデル化

本システムは、画像データを次の属性情報を持つオブジェクトとしてデザインしている。

- プロジェクト名 : CG 制作グループ単位の管理名
- オブジェクト名 : 画像名
- 画像ファイル名 : 画像データの所在を示す URL
- 画像ファイルサイズ : 画像データのサイズ (KB)
- 画像ファイル更新日 : 画像データの最新更新日
- 著者名 : 画像制作者名
- 開発プラットフォーム名 : 開発ホスト名
- バックアップボリューム名 : バックアップ媒体ラベル
- アニメーションフレーム数 : プレビュー画像のアニメーションフレーム数
- プレビュー画像横幅 : 単位はピクセル
- プレビュー画像縦幅 : 単位はピクセル

| マシン種別             | マシン台数 |
|-------------------|-------|
| INDIGO Elan       | 2 台   |
| INDIGO XS24       | 4 台   |
| INDIGO2           | 1 台   |
| IRIS R4000-50VGX  | 1 台   |
| Sun SparcStation2 | 1 台   |

表1 ハードウェア環境

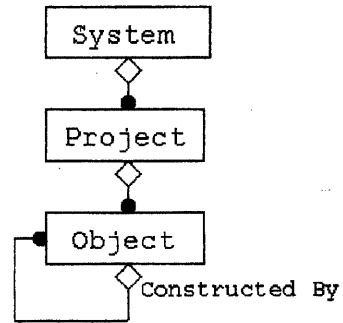


図 6 システム、プロジェクト、オブジェクトの関係

- プレビューファイル名 : プレビューファイルの所在を示す URL
- オブジェクト関係情報 : 部品として再利用したオブジェクト名 (複数指定可)

図 6のように、本システムは CG 制作グループ単位で管理できる「プロジェクト」から構成される。個々の「プロジェクト」は、いくつかの「オブジェクト」をグループ化する。個々の「オブジェクト」は、それを構成する「オブジェクト」と関連付けることができる。この関係は、「全体一部分関係」であり、「ConstructedBy」関係名で表現する。この関係属性は、画像オブジェクトの属性として記録される。

### 4. システム構成

#### 4. 1 ハードウェア/ソフトウェア

本システムのハードウェア環境は表 1 に、またソフトウェア環境は表 2 の通りである。

| プラットフォーム          | ソフトウェア                        |
|-------------------|-------------------------------|
| Sun SparcStation2 | Solaris 2.4                   |
|                   | JDK 1.0.2                     |
|                   | Illustra3.2LE                 |
| Silicon Graphics  | Netscape Communication Server |
|                   | IRIX 5.3(6.X)                 |
|                   | CosmoCode2.0(JDK1.0.2)        |
|                   |                               |

表2 ソフトウェア環境

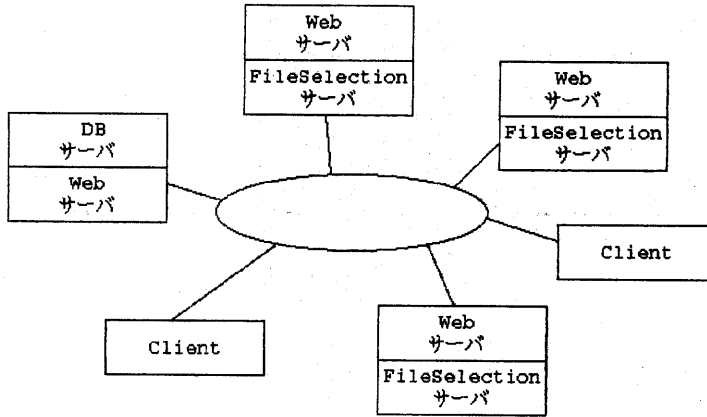


図 7 オペレーションアーキテクチャ

#### 4. 2 オペレーションアーキテクチャ

オペレーションアーキテクチャを図 7に示す。クライアントは FileSelection サーバからファイル情報を取得する。クライアントと FileSelection サーバ間は socket で接続される。

クライアントは、DB サーバから画像オブジェクト情報の問い合わせ、追加、修正、削除のための通信を行う。クライアントと DB サーバ間は Web の CGI 経由で接続される。

また、クライアントは、Web サーバからプレビュー画像や画像データをロードするための通信を行う。これは http で行われる。

#### 4. 3 ソフトウェアアーキテクチャ

本システムは、(図 8) で示すようなサーバクライアントシステムである。サーバとして

は、DB サーバ、FileSelection サーバ、Spawn サーバ、Web サーバがある。

DB サーバは、CG 画像オブジェクト (CG 画像属性) を一元管理している。管理対象データはローカルに存在するものに加え、Web 上で公開されているものも含むことができる。

FileSelection サーバは、画像データやプレビュー画像データを保持するホスト上で起動されるサーバである。本サーバは、クライアントからのファイル情報の問い合わせに応える。本サーバは、あらかじめ指定されたディレクトリ下のファイル情報のみ公開し、それ以外のディレクトリをクライアントは見ることができない。本サーバとのやりとりは、クライアント側の FileSelection ダイアログとの間で行なわれ、クライアントからは、マウス操作だけでファイル情報を閲覧できる。

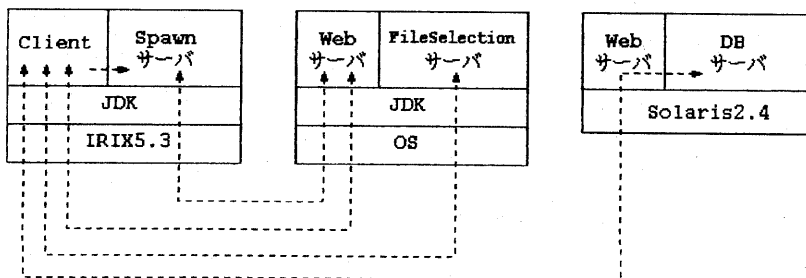


図 8 ソフトウェアアーキテクチャ

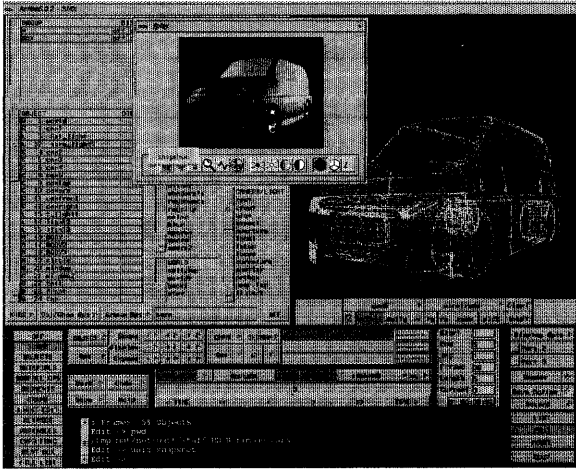


図 9 外部アプリケーションの起動 (PRISMS)

Spawn サーバは、クライアント側で起動されるサーバである。Spawn サーバは、クライアントからの外部アプリケーション起動要求を受け付け、これに関わる全ての操作を代行する。本サーバ機能によって、クライアントはマウス操作だけで指定画像ファイルと関連する外部アプリケーションを起動でき、ユーザは本来の作業を中断されることはない。ユーザは個々のアプリケーションの違いを意識せずにすみやかに画像編集に取り掛かることができる。また、本サーバによって、ユーザは画像データの所在を意識することはない。画像データは、ローカルにあっても、遠隔地の Web サーバ上にあっても良い。画像データが Web 上にあっても、その画像データを取り寄せ、関連した外部アプリケーションをローカルに起動する(図 9)。

本システムは Java アプリケーションとして

| オペレーション      | 所要時間 |
|--------------|------|
| 関係情報表示       | 9 秒  |
| 外部アプリケーション連携 | 4 秒  |

| オペレーション | 所要時間  |       |
|---------|-------|-------|
|         | 20 件  | 40 件  |
| キーワード検索 |       | 2 秒   |
| プレビュー表示 | 3 6 秒 | 7 6 秒 |

表 3 システムパフォーマンス

デザインされているが、FileSelection サーバと Spawn サーバを設定していることにより最小の修正で Applet としてインプリメントすることができる。これは、次のような Java アプレットのセキュリティ制約に対応しており、これによって複数の FileSelection サーバとの通信や外部アプリケーション起動を行なうことができる。

- Applet は、ロードしてきたサーバとのコネクションしか張れない。
- Applet は、プロセスを生成できない。

## 5. 評価

### 5. 1 パフォーマンス

本システム性能は、表 3 のとおりである。

関係情報表示は、「全体一部分」関係にあるオブジェクト総数が 5 件の例であり、その関係情報表示が完了するまでの所要時間である。

外部アプリケーション連携は、ある画像データの属性情報を DB から取得し、それに対応する外部アプリケーション (PRISMS 6.0.2) が起動し始めるまでの所要時間である。

キーワード検索は、検索対象 40 件からキーワードで検索しその結果が得られるまでの時間である。

プレビュー表示は、それぞれ 20 件、40 件のプレビュー表示が完了するまでの所要時間である。

上記いずれの場合も、DB アクセスと画像データのロードに多くの時間を要している。開発当初、DB アクセスは Web の CGI 経由で行っていたが、処理時間として許容できなかったためデーモン方式<sup>[5]</sup>に置換して、この結果表 3 に示すような実測値となった。

### 5. 2 開発経験

本システムの開発期間としては、開発環境の調査と動作検証、システム設計等を含め、5 人月程度を要した。

本システムの開発言語としては Java (JDK1.0.2) を、DB エンジンとして *Illustra* を利用した。<sup>[4]-[6]</sup> これらはすべて今回はじめて触れたものである。

Java を採用した理由は、「マルチプラットフォーム対応、オブジェクト指向プログラミング、AWT による早期開発」という点である。本システム開発の成功の鍵を握っているのが、いかに快適なユーザインタフェースを実現するかであり、早期にプロトタイプ試作を行ないたかったからである。もちろん、未経験の開発環境を採用したこともあり、システムの実現性も含めて早期に検証したかったこともある。オブジェクト指向環境によって、システムの段階的開発が可能であったばかりでなく、環境（言語仕様も含む）自体のテストを進めながら問題点の早期発見とその対処が段階的に行なうことができた。また、Java を利用して感ずるのは、その整備された AWT と、マルチプラットフォームに対応していることの恩恵である。

本システム開発に C++ や C 言語を利用していたら、このような単期間での開発は不可能であったであろう。

### 5. 3 実用性

本システムは、NFS (Network File System) 環境にも対応している。画像データやプレビュー画像の所在は、URL (Uniform Resource Locator) 形式で記録され、リモート画像データは通常 Web 経由でロードされる。しかし、NFS でアクセスできるものは Web 経由でなく NFS 経由でロードされる。システムは、これを NFS マウントポイントテーブルから自動的に判断する。このテーブルは、システム起動時に読み込まれるリソースファイルから設定される。

NFS の automount やダイアログ表示などのプラットフォームによる振舞いの差異への対処がテストの終盤にきて発生した。これらはどれも OS に起因するものである。現在は、UNIX ベースのホストで運用されているが、Windows95

や Mac への対応も OS に起因する部分の修正で対応可能であろう。

本システムは、現在 CG センターで運用されており、CG 制作者の意見を盛り込みながら、操作性の向上改善を続けている。

本システムの開発アプローチは、クライアントサーバシステムによってシステムのトータルコストを低く抑え、かつ操作性の高いシステムを構築しようとするものであった。さらに Web との連携を実現し、インターネット環境での活用を可能とした。この結果、CG 制作支援だけでなく各種マルチメディアデータの管理も行なえる拡張性を実現している。

### 5. 4 今後の課題

Web 経由での画像データ受信には多大な通信コストがかかる。そこで、画像データの圧縮伸長を取り入れるなどの対策が必要である。

また、DB への更新権限はユーザ単位で設定しているが、画像オブジェクト単位の保護制御を組み込むことにより、グループウェアとして活用することができる。

本システムは、画像の再利用の面から画像データの「全体一部分」関係を管理している。しかし、この関係管理だけでは、画像データやプロジェクトを対象としたバージョン管理には適用できない。そこで、「継承」関係を管理することが必要である。

## 6. まとめ

CG 制作における画像データの再利用を図るため、本システムの設計開発はスタートした。ユーザ環境やニーズを考慮しながらデザインを進めた。インターネット環境の活用も考慮した。これは、既存の技術の活用によって開発工数の削減だけでなく、システムの発展性や寿命の長期化も期待できるからである。

システムはますます複雑化し大規模化するなかで、既存技術の活用は必須である。

最後に、本システムの開発にあたりご支援を  
頂いた新潟工科大学の青山幹雄教授、富士通  
(株)の嵯峨喜次氏、柏崎情報開発センターの  
諸氏ならびに関係各位に心より感謝します。

### 参 考 文 献

- 1) 有我成城, 衛藤敏寿, 佐藤治, 白神一久, 西  
村利浩, 村上列: Java 入門,  
翔泳社 (1996).
- 2) Laura Lemay, Chales L.Perkins 共著, 武  
舎広幸ほか訳: Java 言語入門,  
(株)プレントィスホール出版 (1996)
- 3) Sun Microsystems Inc.: The Java  
Language Tutorial (1996).
- 4) Illustra User's Guide:  
Informix Software Inc.
- 5) Illustra Web DataBlade Module, User's  
Guide: Informix Software Inc.