

マルチメディア オブジェクトデータベース： Jasmine

Jasmine: Multimedia Object Database for the Web

あらまし

本稿では、マルチメディア オブジェクトデータベース：Jasmineを支える技術について説明する。とくに、オブジェクト指向とデータベース技術との融合とJasmineの特長、なぜオブジェクト指向データベースがマルチメディア管理と相性がよいか、またJasmineを使ったWebアプリケーション構築を行うにはどのような方法があるのかなど、従来あまり他では紹介されていなかった話題にも多く触れた。またJasmine上でのマルチメディア アプリケーション開発生産性を高めているキーテクノロジーである、様々なISV技術との連携およびJasmine Studioについての紹介も行う。

Abstract

This article explains the basic technologies of the Jasmine multimedia object database, some of which have rarely been mentioned before in Japanese articles on Jasmine. The subjects covered include the features of Jasmine, the integration of the object-oriented paradigm with database technology, the reason why an object-oriented DBMS is well suited to multimedia data management, and methods of building Web-centric applications on Jasmine. This article also describes the linkage with various ISV technologies and the Jasmine Studio, which are the technological keys for enhancing the productivity of multimedia application development on Jasmine.



吉澤庸一（よしざわ つねいち）

1985年東京大学理学部数学科卒。同年富士通入社。入社より1989年までビデオテックスシステムの開発、1989年よりオブジェクト指向データベースシステムの開発に従事。基幹ミドルウェア統括部第五開発部



吉澤尚子（よしざわ なおこ）

1988年カリフォルニア大学バークレー校応用数学科卒。同年富士通入社。以来オブジェクト指向データベースシステムの開発に従事。基幹ミドルウェア統括部第五開発部

まえがき

今日のデータベースの世界で重要なキーワードといえば、オブジェクト指向、マルチメディア、そしてWebからのデータアクセスである。またWebベースのアプリケーションモデルを考えた場合、とくに重要になってくるのは、Java^(注1)によるアクセス、Beans^(注2)(とくにEJB^(注3))のようなコンポーネントモデル、およびアプリケーション間のデータ交換手段として注目されつつあるXMLなどである。現在世界中の先進的なデータベースベンダがこのような分野のサポートを競い合っているという背景の中で、1997年12月、富士通と米国Computer Associates社(以下、CA社)との共同開発によるオブジェクト指向データベース: Jasmineはリリースされた(日本語版は1998年5月リリース)。

本稿では、オブジェクト指向データベース(Object Oriented DataBase: OODB)、マルチメディア、インターネット、および開発環境というテーマに絞って、Jasmineを支える技術について説明する。

オブジェクト指向に対する二つのアプローチ: ORDBとOODB

情報化社会の進展に伴い、ますます複雑化していくソフトウェア仕様、その要求に応えていかななくてはならないソフトウェア開発において、いかにして高生産性を追求していくのか。このような命題に対する答えの一つとして登場した新たなパラダイム「オブジェクト指向」は、プログラミング言語の分野では、1980年代半ばにC++、1990年代半ばにはJavaという形で、また分析・設計の分野においては、OOA/OOD(Object Oriented Analysis/Object Oriented Design)という形で具現化され、いまやソフトウェア開発の世界の主流となりつつある。データベースの世界においてこの新しいパラダイムとデータベース技術とを統合するという試みもまた、1980年代から始まり、現在ではそのアプローチの方法により大きく二つの流れがある。一つ目はオブジェクトリレーショナルデータベース(Object Relational DataBase: ORDB)というアプローチであり、これはちょうどC言語を拡張する形でC++が登場したように、リレーショナルモデル(およびその操作言語であるSQL)にオブジェクト指向としての機

能を拡張・追加するというものである。この範ちゅうに属するデータベースとして、OracleやInformixなどがある。もう一つのアプローチは、ピュアなオブジェクト指向データベース(Object Oriented DataBase: OODB)であり、オブジェクト識別子^(注4)、クラス^(注5) インスタンス^(注6)、メソッド^(注7)、継承^(注8)というようなオブジェクト指向固有の概念を取り入れたデータベースモデルを持つものである。さらにこの範ちゅうに属するデータベースは、OODBの草分けであるGemStone(Gemstone System社)やObjectStore(Object Design社)に代表されるようなオブジェクト指向言語に永続オブジェクト機能を追加するというアプローチをとるタイプ(本稿では言語型と呼ぶ)と、トランザクション・排他・リカバリなどのデータベース技術の上にオブジェクトモデルを構築するタイプ(本稿ではDB型と呼ぶ)とがある。Jasmineはこの後者(DB型)に分類されるものである。

一般的に言語型とDB型との違いは、言語型では、オブジェクトをメモリ上に展開し、オブジェクト間の参照もすべてポインタに置き換えていくというやり方(swizzlingという)を行う。このため比較的小さなサイズのデータや、ある程度のデータ量までならばプログラムのメモリアクセスと同等の性能で処理される。しかしマルチメディアのような大きなデータを数多く処理するとなると、もともとDBからのアクセス効率を考慮して設計されているJasmineのようなDB型の方が有利になってくる。Jasmineは、これ以外にも、オブジェクトサーバアーキテクチャをベースにしたオブジェクトロック(排他制御の最小単位をインスタンスまでサポート)や、ODQL(Object Definition and Query Language)というデータベース定義・操作・検索言語のサポート(ODQLのみ、またはCやC++などのホスト言語に対する埋め込み型のいずれでもメソッドの記述が可能)など、数々のデータベース技術を基盤とした機能的特長を持ち、一層エンタープライズ指向のデータベースとなっている。

Jasmine とマルチメディア

つぎに、なぜOODBがマルチメディア管理に適しているのかについて述べる。OODBがマルチメディアの管理に適している理由にはつぎの三つのことが挙げられる。

(注1) Sun Microsystems, Inc. が提唱する新たな言語体系。

(注2) Java Beans. Java によるコンポーネント部品。

(注3) Enterprise JavaBeans

(注4) システムが自動的に割り当てるオブジェクト個々に固有の識別子。

(注5) 同じ特性を持つオブジェクトの集合あるいはその特性に関する定義記述。

(注6) クラスに記述された特性を持つ具体的な一つ一つのオブジェクト。

(注7) オブジェクトに対して行うことのできる操作・手続き。

(注8) 上位クラスで定義された属性やメソッドが下位クラスにも引き継がれる仕組み。

(1) カプセル化

マルチメディアデータには、動画、静止画、音声、テキストのように様々なタイプがあり、かつそれぞれのタイプに応じて固有の操作(例えば動画であれば再生、早送り、ポーズなど)が付随している。このような対象物とそれに対する固有の操作との組合せという考え方はオブジェクト指向の考え方と共通するものである。

(2) 拡張性

マルチメディアデータの場合、一つのデータ種においてもその形式(例えば静止画におけるGIFやJPEG)には様々なバラエティがあり、さらにまた今後も新たな形式が増えてくる可能性がある。このような場合にオブジェクト指向では、継承というメカニズムを利用することで、新しい形式に容易に対応することが可能である。

(3) 関係管理

マルチメディアデータの管理には、データ相互の関連付けや包含関係など様々なデータ間の複雑な関係管理が伴う場合が多い。例えば文書データでは、一つのページとそれを構成するテキストや絵との包含関係、ページや項目相互の参照関係、関連する文書の分類関係など、データ間の様々な関係が必要となってくる。このような物と物との間の複雑で自由な関係を表現することは、オブジェクト指向モデルの最も得意とするところである。

より一般的な言い方をすれば、マルチメディアとオブジェクト指向のどちらもが、より現実に近いモデルをコンピュータ上で表現する手段として登場してきたものであり、その意味ではこの二つに親和性があるのは至って自然の結果と言えよう。とりわけDB型の特長を持つJasmineはマルチメディアデータの管理に適しているわけである。

Jasmine のマルチメディアクラスライブラリ

Jasmineは前述のOODBとしての特性に加え、様々なマルチメディアデータを管理するためのクラスとその操作を行うメソッド群をまとめてマルチメディアクラスライブラリという形式で提供している。これによって、ユーザがマルチメディア操作を行うアプリケーションの記述を容易にしている。Jasmineのマルチメディアクラスライブラリのクラス階層を図-1に示す。

クラスライブラリ

ところで、マルチメディアクラスライブラリはクラスライブラリ的一种である。クラスライブラリは、Jasmineの拡張性のキーとなる考え方である。以下、このクラス

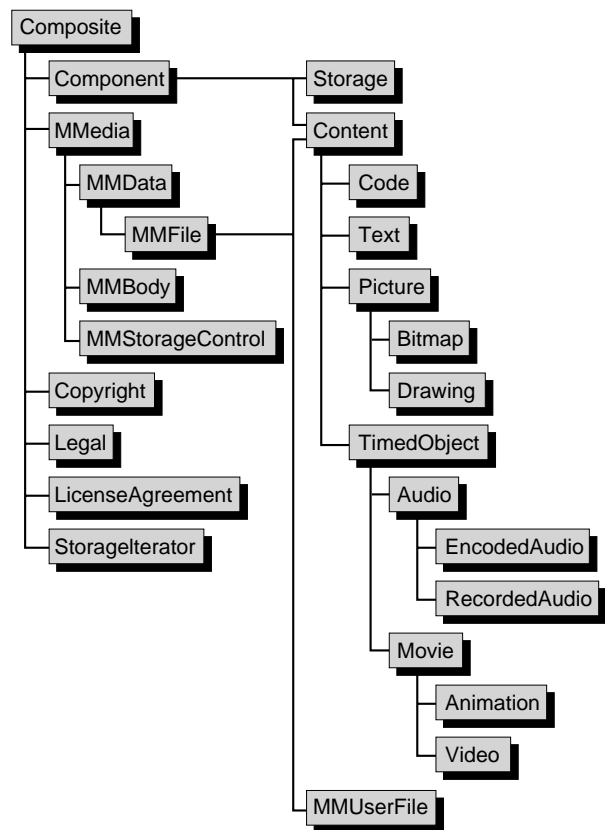


図-1 Jasmineのマルチメディアクラスライブラリ
Fig.1-Multimedia class library in Jasmine.

ライブラリについて説明する。

Jasmineでは、ある特定の機能や特定の分野のアプリケーション、機能部品セットを実現するためのクラス群(クラスツリー)をまとめてクラスファミリという単位でオブジェクトとメソッドの管理をしている。Jasmineが提供するクラスファミリには、先に登場したマルチメディアクラスライブラリ(mediaCF)以外にも、以下のものがある。

- (1) Jasmineのアプリケーション開発環境であるJasmine Studioが、その上で開発されたアプリケーションを格納するjadelibCF
- (2) JasmineのWebアクセスアプリケーション開発環境であるWebLinkが、そのテンプレートデータなどを格納するweblinkCF
- (3) 既存データベースにアクセスするためのSQLゲートウェイ機能を提供するsqlCF

もちろん、ユーザ自身がクラスファミリを定義し、その上にアプリケーションパッケージを開発し頒布することも可能である。XMLサポートなどもこのようなクラスファミリの一つとして拡張されることになる。またシステム提供のクラスファミリ以外にもいろいろな分野のソフトウェアベンダが、この技術の上にそれぞれ特徴的な

クラスライブラリを提供しており、それらの機能の中には特にマルチメディアの高度なハンドリングを容易にする技術が数多く含まれている。そこで次にJasmineにおける様々なISV技術との連携について説明する。

様々な ISV 技術との連携

今日、マルチメディアの分野のみならずあらゆるアプリケーション分野において、そのサポート機能はどんどん広がり、各分野において深い専門的なテクノロジーが要求される状況にある。このような現実の中で、一つのベンダがそれらすべての機能をカバーすることは不可能である。しかし同時に、このような付加価値のない裸のデータベースではその利用価値も半減してしまうことになる。JasmineではDPP(Development Partner Program)と呼ばれる戦略を採り、世界中の様々な分野の先進的なISV技術と連携することによって、エンドユーザやアプリケーション開発者に必要な機能を提供している。クラスライブラリで外部DB、あるいはファイルとアクセスするライブラリをラップ(wrap)するというやり方による連携イメージを図-2に示す。また、利用面から代表的な連携ソフトウェアを表-1に示す。

上記以外にも現在100社以上がJasmineのDPPとなっている。これらのISV技術を用いた様々なソリューションについては、本号掲載の「Jasmineによるソリューション」を参照されたい。

Jasmine とインターネット

今日のインターネット(あるいはイントラネットやエクストラネット)の発展・普及は、Webなくしてはあり得ない。そこでまず、Jasmineデータベースに対するWebからのアクセスを行うための代表的な機能であるWebLinkについて説明する。

WebLinkは、Webサーバ(HTTPサーバ)とJasmineデータベースサーバとの間でクライアント(Webブラウザ)からの検索・更新などの要求をサーバに渡し、その結果をHTMLページの中に埋め込んでクライアント側に返すというゲートウェイサーバとしての働きをするものである。WebLinkアプリケーションは一連のHTMLページからなり、ユーザはその中にWebLinkが提供する拡張タグを利用して、Jasmineサーバにアクセスするコードを埋め込む。Jasmineサーバに対するアクセスは、Jasmineが提供するODQLを用いる。このようにして作成されたHTMLテンプレートは前述したサーバ側のweblinkCFに格納される。データアクセス時は、WebLinkが登録されたテンプレートをたどることによって、そこにデータの処理結果を埋め込んで表示する。また、WebLinkはユーザのセッションの制御も行う。WebLinkの管理するセッションには個々のユーザ認証を行うプライベートセッションと不特定多数からのアクセスを許すパブリックセッションとがある。WebLinkのアーキテクチャを図-3に示す。

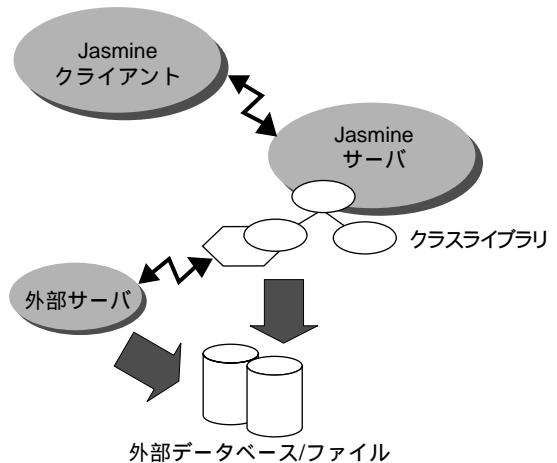


図-2 クラスライブラリによる連携

Fig.2-External system connectivity using class library.

表-1 Jasmineと連携する代表的なISV技術

ソフトウェア名	分野	ベンダ名
MapDK	地図情報システム	IncrementP社
ImageServer	FlashPix画像, VRMLパノラマ, 3Dオブジェクト	LivePicture社
Gen-trix Studio	簡易3Dオブジェクト	ゲン・テック社
ConceptBase	テキスト検索	ジャストシステム社
ISGNavigator	OLE DB	ISG社
ImageCABINET	イメージ文書管理	富士通大分ソフトウェアラボ

一般的に、Webからデータベースにアクセスするためには、CGIプログラムであれ、アプレットであれ何らかのプログラムを書く必要がある。その意味では、WebLinkはHTMLに関する知識だけで、テンプレートを作成することによって簡単にJasmineデータベースにアクセスするWebアプリケーションが開発できる、Jasmineの特長的な機能の一つ(かつエレガントなWebアクセスソリューションの一つ)である。さらに、分散オブジェクト運用に基づくアプリケーションモデルというようなことまで考えると、例えば、富士通製品であるINTERSTAGE^(注9)に代表されるようなCORBA^(注10)との連携にせよ、あるいはCOM

(注9) 富士通製CORBA製品。

(注10) Common Object Request Broker Architectureの略語: OMG (Object Management Group) が規定した異機種間アプリケーション連携のための国際標準仕様。

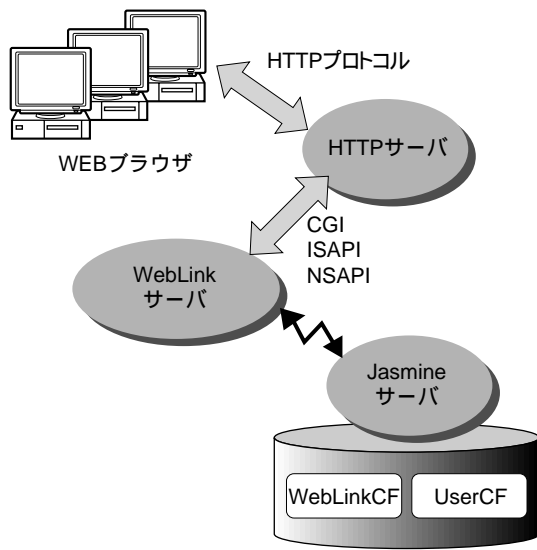


図-3 WebLinkのアーキテクチャ
Fig.3-Architecture of WebLink.

(Component Object Model)やJavaBeanなどのコンポーネントモデルをベースとしたアプリケーションの構築にせよ、その基本となってくるのはやはりC++やJavaのようなオブジェクト指向プログラミング言語とのバインディングである。Jasmineもまたこれら二つの言語インタフェースをサポートしているが、とくにJavaインタフェースについては次の2種類の方法を提供している。

(1) Jp(Java proxy)

Jpは、J APIというJasmineにアクセスするためのクラスライブラリと、jpcgというユーティリティコマンドから構成される。jpcgコマンドは既存のJasmineクラスに対するJavaクラス(プロキシクラス)を生成するコマンドであり、ユーザはこのようにして生成されたクラスとJ APIとを使うことによってJasmineデータベースにアクセスするアプリケーションやアプレットを記述することができる。

(2) pJ(persistent Java)

pJは、Jpとは逆にJavaの中に永続オブジェクトを導入するアプローチである。ユーザは永続化させたいJavaクラスに対してpJが提供するJPPコマンドを用いることでJasmine側に対応するJasmineクラスを定義することができる。このようにしてJasmine側に定義されたクラスは単に永続オブジェクトを格納するための器にすぎず、JavaプログラマはこのJasmineクラスを意識する必要はない。その意味でpJはpure Javaなアプローチであり、ODMG^(注11)

が提唱するJavaバインディングと同等のものである。

これらのJavaインタフェースや、Jasmineが提供するBeansを利用して、Jasmineデータベースにアクセスするためのアプリケーションやアプレット、さらにはより本格的な多階層(multi-tier)アプリケーションモデルの構築を行うことができる。

Jasmine Studio

最後にJasmineのアプリケーション開発環境であるJasmine Studioについて簡単に紹介する。Jasmine Studioの機能には大きく分けて、データベース管理機能とアプリケーション開発機能の2種類がある。

データベース管理機能

Windows Explorerライクのビューを持つクラスブラウザ^(注12)、プロパティインスペクタ^(注13)などの機能を用いることによって、データベースに格納されているクラスの定義の参照を行うことができる。また、クラスやメソッドの定義、またオブジェクト間の関係付けなど、いわゆるモデルの構築もすべてGUI操作で行うことができるため、データベース操作言語に不慣れなユーザでも容易にデータベースの構築や管理を可能にする。

アプリケーション開発機能

Jasmine Studioのアプリケーションは、GUIによる機能の貼り付けという大変簡単な方法で作成できる。つまりシーンと呼ばれるシートの上に、マルチメディアデータやある条件を満たすオブジェクトの集合(クエリオブジェクト)などのデータベースで管理されるオブジェクトや、ボタン、リストボックス、あるいはActiveXコントロールなどのローカルウィジェット(部品)を配置するというやり方で開発を行う。シーン上に配置されたオブジェクト(scene item)に対しては必要に応じて各種のアクションを指定・記述する。このようにして作成されたシーン、およびシーン上の各要素はすべてJasmineサーバ側のクラスライブラリ(jadelibCF)に格納されるため、ユーザはアプリケーションを構成する各種ファイルを個々に自分で管理することに煩わされることはない。

アプリケーションのdeployment

Jasmine Studioで開発されたアプリケーションは、そのままWindowsアプリケーションとして実行、Webブラウザ + Jasmine Plug-inでWebブラウザ上で実行、のい

(注11) オブジェクト指向データベースの標準化団体。

(注12) Class Browser : クラスの定義情報をツリービューで見せるJasmine Studioのツールの一つ。

(注13) Property Inspector : オブジェクトの属性値を参照するためのJasmine Studioのツールの一つ。

ずれもが変更なしに可能であり、クライアントサーバタイプおよびWebタイプの両運用タイプに対応したアプリケーションの頒布を行うことができる。

む す び

以上Jasmineの主な技術について説明してきたが、Jasmineにはここに紹介した以外にも、データベース定義言語、データベース操作言語、およびデータベース検索言語であるODQL、Jasmine ActiveXなどの持ち味があり、それらについては、Jasmineのホームページ(<http://www.fujitsu.co.jp/hypertext/softinfo/product/db/JASMINE>)を参照されたい。また、機能やアーキテクチャだけではなく、オブジェクト指向設計と、それをオブジェクトデータベースで実装した際の性能といった話題についてもまた別の機会に譲らせていただきたい。

Jasmineは富士通とCA社が全世界で協調ビジネスを推進しており、世界中のユーザやソフトウェアベンダの声、経験、技術を取り込みながら成長し続けている。日

本のお客様にも是非、このJasmineが持つ様々な最先端技術を体験・実感の上、ビジネスに役立てていただきたいと思う次第である。

参考文献

- (1) S. Khoshafian, S. Dasananda, and N. Minassian : The Jasmine Object Database . Morgan Kaufmann Publishers , 1998 .
- (2) S. Khoshafian : A Comparative Analysis of ObjectStore and Jasmine . revision 2.0 , 1998 .
- (3) R.G.G. Cattel , and D.K. Barry : Object Database Standard : ODMG 2.0 . Morgan Kaufmann Publishers , 1997 .
- (4) R. Orfali and D. Harkey : Client/Server Programming with JAVA and CORBA . 2nd Edition , John Wiley and Sons , 1998 .
- (5) D. Chang and D. Harkey : Client/Server Data Access with Java and XML . John Wiley and Sons , 1998 .