

BIGLOBEのインフラストラクチャ

# BizMAP フレームワークによるBIGLOBE 業務システムの プログラムレス開発

Development of BIGLOBE Business System by BizMAP Framework

檜垣清志\*  
Kiyoshi Higaki

藤田英利\*\*  
Hidetoshi Fujita

## 要旨

本稿では、BizMAP フレームワークによるBIGLOBEでの業務システム開発を紹介します。

BizMAPは、上流工程から下流工程までをトータルにサポートする独自フレームワークであり、標準アーキテクチャ、ビジネスルールエンジン、標準プロセス、標準設計書、および汎用コンポーネント/汎用サブシステムで構成されています。

このビジネスルールエンジンは、標準アーキテクチャで規定しているコンポーネントを、データマッピング定義をするだけで、プログラミングすることなく業務システムを実現します。

BIGLOBEでは、このフレームワークに基づき工場のような分業体制を構築し、多種多様な業務システムを早く、低コスト、高品質で提供しています。

This paper introduces the business systems development in BIGLOBE by the BizMAP framework.

BizMAP is an original framework which supports from upstream software development to downstream development in total, and consists of the standard architecture, business rule engine, a standard process, standard design specifications and a general-purpose component / a general-purpose subsystem.

This business rule engine realizes an business system, without only carrying out a data mapping definition and programming the component specified by the standard architecture.

In BIGLOBE, based on this framework, division-of-work organization like a factory is built, and a variety of business systems are early offered for low cost and high quality.

## 1. まえがき

ブロードバンドの普及でインターネットがより身近になるとともにプロバイダサービスが多様化し、それを支える業務システムの開発サイクルがより短くなっています。さらに、顧客へのサポートサービスは、より手厚くきめ細かいものへ向かっており、これらをサポートする業務システムの重要性は高まる一方です。

一方BIGLOBEの顧客管理などの業務システムは、立ち上げ当初、データベースを中心としたクライアント/サーバ型で構築しており、クライアントPC上で処理をプログラミングするという一般的な方式で構築していました。しかし、価格競争と変化の激しいインターネットプロバイダ事業にあって、新しいビジネスプロセスを、早く、安価に実現することが大変重要なテーマとなっていました。

そのようななかで、ビジネスプロセスの分析と要件確定といった上流工程から、設計、実装、出荷といった下流工程までをトータルに標準化したフレームワークであるBizMAP (Business rule & component interface data MAPping method) を独自開発しました。

まず第2章では、BIGLOBEの業務システムの特徴を説明します。続いて第3章では、業務システムの標準アーキテクチャと、プログラムレスを実現するビジネスルールエンジンを解説します。第4章では、これらを活用して実際に効果を上げるための開発プロセスと分業体制について説明します。

## 2. BIGLOBE の業務システムの事例

BIGLOBEでは、誰でもブロードバンドに入会できるよう、電話での入会サポートを実施しています。すなわち、まったくインターネットが分らないお客様に対して、入会条件やご自宅の回線などの環境状況を尋ね、申請代行から訪問セットアップまでをフルサポート対応するといったサポート業務が増えてきました（図1、図2）。

\* BIGLOBE構築運営本部  
BIGLOBE Design and Operations Division

\*\* NECシステムテクノロジー Webインテグレーション事業部  
NEC System Technologies, Ltd.

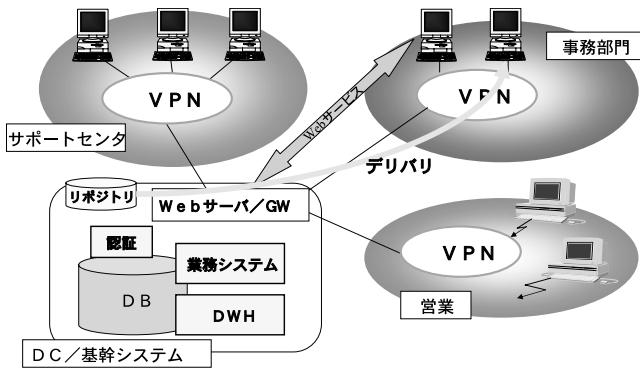


図1 BIGLOBE の業務システム例

Fig.1 Example of operating system of BIGLOBE.

## 【サポート業務のビジネスプロセス例】

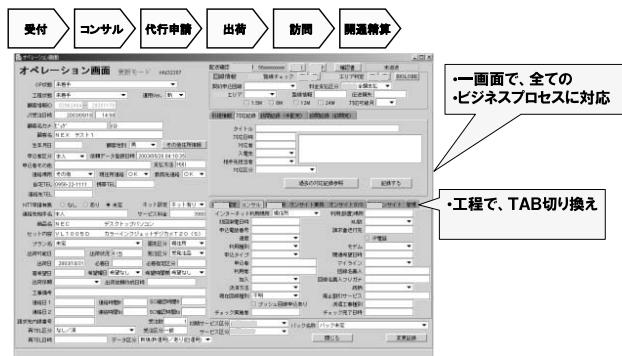


図2 ビジネスプロセスと画面の例

Fig.2 Example of business process and screen.

こういったサポート業務においては、オペレータの生産性を常に高いレベルで維持する必要があり、それを実現できる業務システムが重要となります。BIGLOBEでは、オペレータの生産性をチェックし改善するといった、BPM (Business Process Management) サイクルを実施しており、常に最適なビジネスプロセスへ改善させています。したがって、こういったビジネスプロセスの変更に対し、低コストで素早く対応できる業務システム開発手法が非常に重要です。

## 3 . BizMAP フレームワーク

BizMAP フレームワークは、DOA (Data Oriented Approach) の考え方をベースとしたデータ中心のフレームワークであり、データ項目間の関係性を厳密にデータマッピング表に記述することで、従来型のソフト開発で課題であるテキスト文章による要件定義（ビジネスルール）記述のあいまいさや抜けを極力排除します。

加えて、確立したアプリケーションアーキテクチャと、汎用コンポーネント/汎用サブシステムを持っているため、分析したビジネスルールをこれにあてはめて、組み立てただけで業務システムを完成させることができ、工場での製品生産のように種々業務システムを均一の品質と短納期で

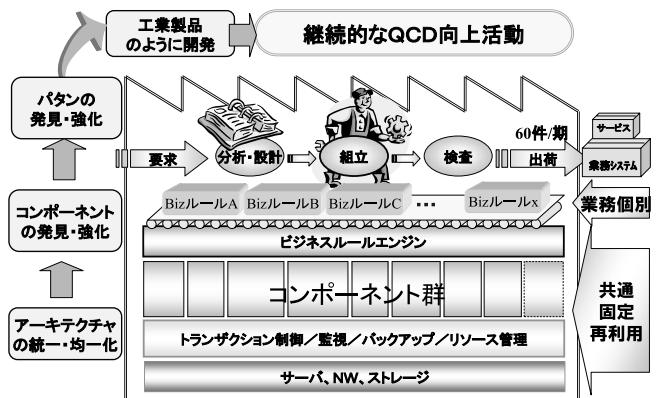


図3 フレームワークを活用した開発

Fig.3 Development which utilized framework.

開発できます（図3）。

このような特長を持つBizMAPは、業務システム開発の上流工程から下流工程までの全開発工程をサポートするために、標準開発プロセス、標準アーキテクチャ、標準設計書、ビジネスルールエンジン、汎用コンポーネント/汎用サブシステムを用意しています。

## 3.1 アーキテクチャ

BizMAPでは、業務要件（ビジネスルール）と、アプリケーション実装の2つの領域を完全分離し、それぞれ固有にビジネスルールアーキテクチャとアプリケーションアーキテクチャを規定します。この2領域のアーキテクチャはそれぞれ独立ですが、どちらもデータ中心であり、データ項目を最小単位としているため、業務要件であるビジネスルールを、容易にアプリケーション実装に変換することができます。

また、このように企業内システムのアーキテクチャをレイヤモデル化し確立するフレームワークとして、EA (Enterprise Architecture) がおなじみですが、BizMAP フレームワークは、それと同じレイヤ構造になっていることが分かります（図4）。

加えて、こういったアーキテクチャをレイヤ分割し独立させているので、それぞれのアーキテクチャを独自に改良、

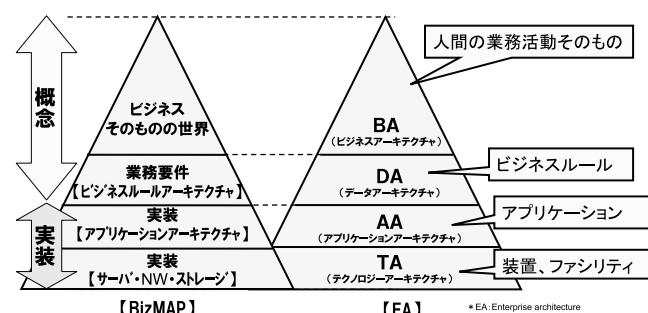


図4 BizMAP と Enterprise Architecture

Fig.4 BizMAP and Enterprise Architecture.

発展させることができ容易になりました。

### 3.2 ビジネスルールアーキテクチャ

BizMAPの業務システムを構成する最小単位はデータ項目であることから、これらデータ項目の関係性がどうなっているかをすべて表形式で記述できるようにしています。この記述をビジネスルールと呼び、アプリケーション実装とは無関係に記述します。

次に、このビジネスルールのアーキテクチャとして重要な「ビジネスルール種類」と「ビジネスルール領域」について説明します。

まずビジネスルール種類は、ビジネスルールを、「データ項目の意味定義」、「データの加工や導出の定義」、「データセット(DOAのエンティティに近い)の定義」、「ワークフローの定義」の4つに分類し、これらをうまく組み合わせることで、すべてのビジネスルールを表現します(表1)。

これによって従来文章などのテキスト文書であいまいになりがちであった記述を、正確で属人性のないものにします。

次にビジネスルール領域ですが、これはビジネスルールを記述する領域を、ユーザインターフェース・ビジネスルール（UIBiz ルール）と、トランザクション・ビジネスルール（TrBiz ルール）の2つの領域に分離し、それぞれの標準形式でビジネスルールを記述します。

UIBiz ルールは、入力イベントをトリガとし、画面状態とその遷移を中心にビジネスルールを記述します。一方 TrBiz ルールは、ユースケース (UC) をトリガとし、業務の状態とその遷移を中心にビジネスルールを記述します。

このように2つの領域にビジネスルールを分けて記述することにより、画面などのユーザインタフェースの複雑なビジネスルールとトランザクション処理に絡む複雑なビジネスルールを分離し、独立に分析、記述できるメリットがあります。さらに、状態遷移図やデータマッピング表などのチェックの容易な標準設計書により、属人性を極力排除し、品質向上と短納期化を実現します。

### 3.3 アプリケーションアーキテクチャ

ビジネスルールを表形式で正確に記述しても、アプリケーションの実装方式を毎回考えていたのでは、作りが属人的でバラバラになり、開発の長期化、品質低下およびコストアップになってしまいます。

BizMAPでは、業務システムのアプリケーションアーキ

表1 4種類のビジネスルール  
Table1 Four kinds of business rules.

ルール	内 容
L0Bizルール	データ項目の意味定義
L1Bizルール	データ加工・データ制約・データ導出定義
L2Bizルール	データセットの定義
L3Bizルール	ワークフローの定義

テクチャを規定しており、業務システムごとに新たなアーキテクチャを作ることなく、ビジネスルールをアプリケーションアーキテクチャに対応付け、組み立てるだけで業務システムを開発します。このうち2つの重要なアプリケーションアーキテクチャについて説明します。

1つ目はMVC+（Model View Control Plus）アーキテクチャといい、業務システムの構成コンポーネントを、V層、P層、C層、M層、D層、T層の6つのコンポーネントタイプに分割する規定です。このうちV層、C層は業務固有のコンポーネントとし、P層、M層、T層を汎用コンポーネントと位置づけています（図5、表2）。

我々は、業務システムを開発する時に、これら汎用コンポーネントを必ず利用することとしており、業務システムごとに独自に作ることを禁止しているため、100%の再利用を実現しています。またD層は、データベース上のデータモデルをアクセスするためのメソッドであり、M層コンポーネントの構成要素となります。両者の関係は、部品とアセンブル単位であるといえます。

2つ目はサブシステムタイプといい、サブシステムが保有すべきデータモデルを、リソース系、イベント系、在庫系、要約系といったDOAのデータ類型によってサブシス

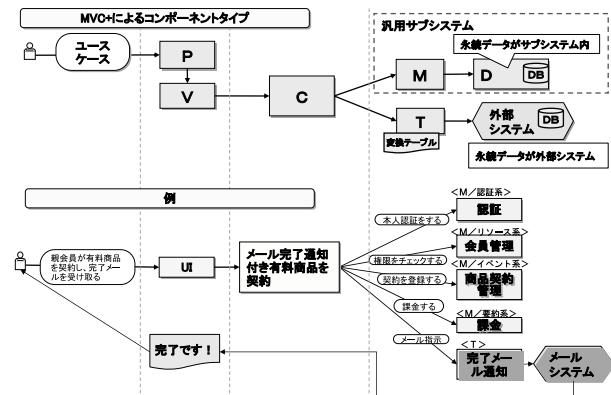


図5 MVC + アーキテクチャ

Fig.5 MVC+ architecture.

表2 MVC + アーキテクチャとコンポーネントタイプ  
Table2 MVC+ architecture and component type.

	種類	内容
業務固有	V層コンポーネント	業務システム固有の画面機能 入力チェックやトランザクション選択やガードチェック
	C層コンポーネント	業務システム固有のトランザクション機能
汎用	P層コンポーネント	画面や帳票の表示部品
	M層コンポーネント	汎用サブシステムのメソッド。C層コンポーネントにサブループを提供する。
	T層コンポーネント	外部システム等への接続アダプタ。外部システムとのプロトコル差異を吸収し、C層にサブループとして機能を提供する。
	D層コンポーネント	DBのデータモデルへのアクセス。M層コンポーネントにデータアクセスを提供する。

テムを規定、分類します。すなわち汎用サブシステムは、内部にデータ類型固有のデータモデルを持ち、M層コンポーネントをメソッドとして有します。

こういった大きな塊の汎用サブシステムを、業務システム開発時に毎回再利用するため、大幅な納期短縮、低コスト化および品質向上が実現できました。現在は、BIGLOBE の業務範囲での活用ですが、さらに強化し、対応可能な業務範囲を拡大していきます。

#### 3.4 ビジネスルールエンジン

BizMAPでは、コンポーネントを組み立てることで業務システムを実現するわけですが、我々は、これを「シナリオ」というビジネスルール定義のみで動作する独自のビジネスルールエンジンで実現しています。このビジネスルールエンジンは、「プローカー&プロトコル機能」、「ワークフロー機能」、「データマッピング機能」の独立した3レイヤで構成されています（図6）。

第一層目のプローカー&プロトコルは、MVC+で規定したコンポーネント間を繋ぐ仕組みですが、BOBIO (BIGLOBE BASIC IO) という軽くて高速な独自プロトコルで実現しており、認証処理でも利用できています。またこの部分は、必要に応じプロトコル変換アダプタの追加ができるので、今後デファクトへの対応も可能です。

第二層目のワークフローエンジンは、シナリオ定義でコンポーネントを順番に動作させ、1つのトランザクション処理を実現します。

第三層目のデータマッピングは、ワークフローエンジンがコンポーネントを順番に呼び出す時の条件判定や、インターフェース間のデータマッピング、データ加工およびデータ導出をデータマッピング定義で実現します。

これらのビジネスルールエンジンは、ビジネスルールアーキテクチャに合わせ、TrBiz ルール用と UIBiz ルール用があり、前者は独自開発の BLC (Business Logic Controller)，後者には独自開発の BizMAP/Web と市販の Visual Frame を使ってています。

BizMAP/Web は、Web で業務をする不特定多数向けのリッチクライアント開発用の Flash ベースのビジネスルールエンジンです。BizMAP/Web は、NEC システムテクノロジー製品の “iBizBlock” に、ビジネスルールエンジンとし

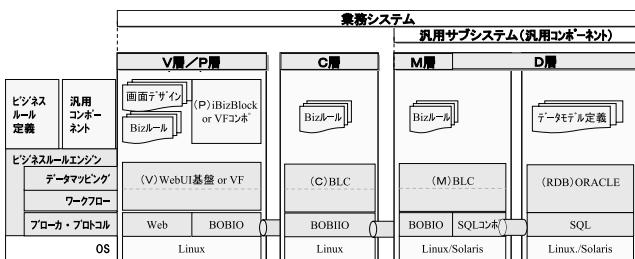


図6 業務システムとビジネスルールエンジン

Fig.6 A business system and business rule engine.

て追加する予定です。

もう一方の Visual Frame は、データ項目数が数百といった非常リッチな画面でかつ、タイトなオペレーション業務をこなす専用チーム向けのリッチクライアントを、開発するためのビジネスルールエンジンです。

#### 4. 標準開発プロセスと分業体制

このフレームワークによって、種々の業務システムを同時に工業製品のように開発するためには、工場のような標準生産プロセスと、それを実行する分業体制が不可欠です。BIGLOBE では、標準化と分業により、従来のソフト開発では実現が難しかった、工業製品のような生産体制を確立しつつあります。

##### 4.1 標準開発プロセス

BizMAP の開発プロセスは、実業務の分析（ビジネスモデリング工程）、システム化する業務要件の整理（要求工程）、ビジネスルールを所定のアプリケーションアーキテクチャに対応付ける作業（分析工程、設計工程）、対応付けたビジネスルールを実行形式に実装する作業（実装工程）、品質保証テスト（テスト工程）、切り換え（移行工程）という基本工程からなります（図7）。

ビジネスモデリング工程では、ビジネスプロセス、業務フロー、画面イメージなど業務要件を固めるベースとなる資料を作成します。要求工程ではこれをベースに、標準設計書でビジネスルール化しながら、要求部門に確認し補正していきます。

分析設計工程では、このビジネスルールをコンポーネントに対応付けしますが、過去の業務システム開発による蓄積から類似のパターンを選択して、非常に短期かつ高精度で設計することができます。

実装工程は、この設計情報を実装に変換しますが、ツールによる自動変換がかなりを占めており、将来的には、全自動変換を実現する予定です。

またテスト工程は、従来のようなブラックボックステストではなく、要求分析したビジネスルールに基づくシステムテストが中心となります。

##### 4.2 分業体制

この標準開発プロセスを使い、同時に多数の業務システムを開発するために、当該フレームワークに基づいた分業体制を敷いています。すなわち、ビジネスモデリングと現場への適用（移行）を指揮するビジネスエンジニア (BE)，要求工程、分析工程および導入工程といった業務システム

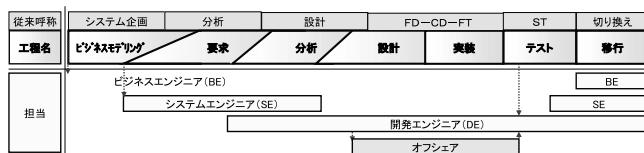


図7 開発フェーズと分業

Fig.7 A development phase and division of work.

開発の最も重要な部分を取り仕切るシステムエンジニア(SE), ビジネスルールをビジネスルールエンジンを使い設計実装する開発エンジニア(DE)の三者に分担しており, これらが, 各工程をオーバラップさせ主導権をバトンタッチしながらスムーズに開発を進めています。この分業方式と明確な成果物分担により, 無駄作業を排除し, さらに設計図を三者で共有することで, 最短1ヵ月での業務システム開発を成功させた事例もあります。

## 5. むすび

以上のようにBizMAPでは, ビジネスルールとアプリケーション実装の完全分離および, アーキテクチャの標準化により, 汎用サブシステムとコンポーネントをビジネスルールエンジンによるデータマッピングのみで業務システムを開発できるようにしています。

今後もBIGLOBEのBPMを支えるための業務システムを, 高い生産性と品質で素早く提供できるようBizMAPフレームワークの強化を図るとともに, 適用範囲を拡大していきます。

\* Visual Frameは, メディア情報開発(株)の登録商標です。

\* Flashは, Macromedia, Inc.の米国およびその他の国における商標または登録商標です。

## 参考文献

- 1) 椿正明, (株)データ総研, DRI通信85号.  
[http://www.drinet.co.jp/technical6\\_85.html](http://www.drinet.co.jp/technical6_85.html)
- 2) 椿正明, 「データ中心システムの概念データモデル」, オーム社,  
1997年1月.

## 筆者紹介



Kiyoshi Higaki

ひがき

きよし

**檜垣 清志** 1987年, NEC入社。現在, BIGLOBE事業本部BIGLOBE構築運営本部グループマネージャー。



Hidetoshi Fujita

ふじた

ひでとし

**藤田 英利** 1988年, NECソフトウェア中国入社。現在, NECシステムテクノロジー Webインテグレーション事業部マネージャー。