
Webアプリケーション入門コース

Webエンジンの思想と将来構想

ムラテック情報システム株式会社

目次

1. エンジンの特長
2. システム構成
3. データフロー例(検索系)
4. 将来構想(ロードマップ)

1. エンジンの特長

Webアプリケーションフレームワーク(以下、Webエンジン)の特長を挙げるとすれば、以下の3つに集約できます。

- (1). 短納期指向
- (2). 効果的な開発
- (3). RDBMSが中心

1. エンジンの特長

(1) 短納期指向

短納期指向とは、開発の初めから実稼動までのリードタイムの短縮を目指すというものです。

エクストリーム・プログラミング (**XP**) などのアジャイルな方法論が注目を集めていますが、考え方としましては、**TOC**(Theory of Constraints: 制約条件理論)による**システム開発の変革**による、**スループット向上**を中心に、XP的な考え方を取り込んでいます。

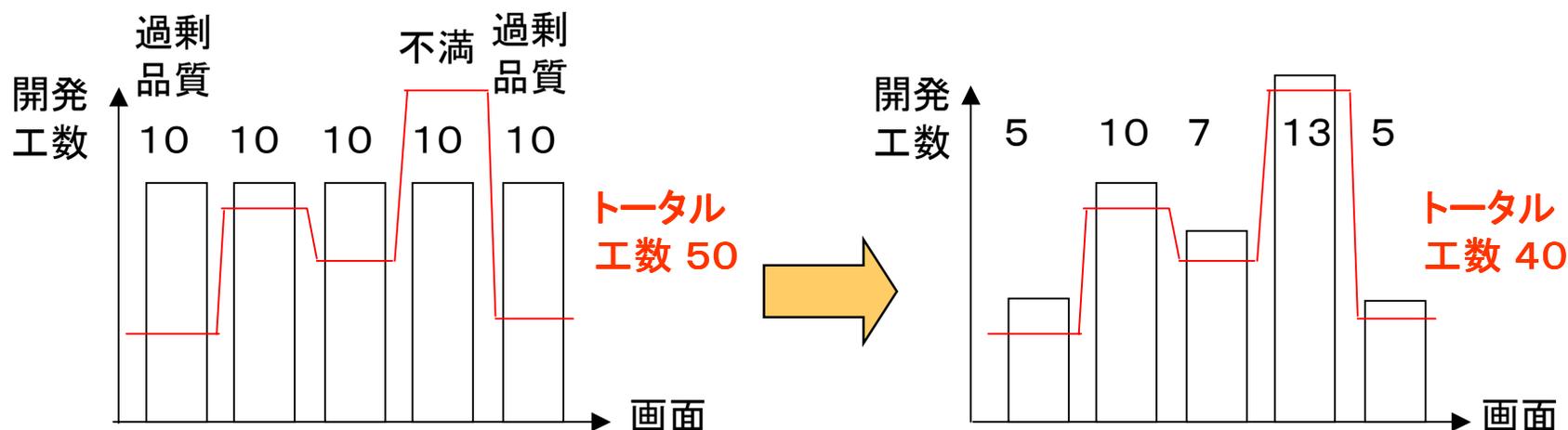


1. エンジンの特長

(2) 効果的な開発

効果的な開発とは、『**効率の追求から効果の追求**』に考え方をシフトし、『**今、必要な機能のみを実装する**』という開発手法です。

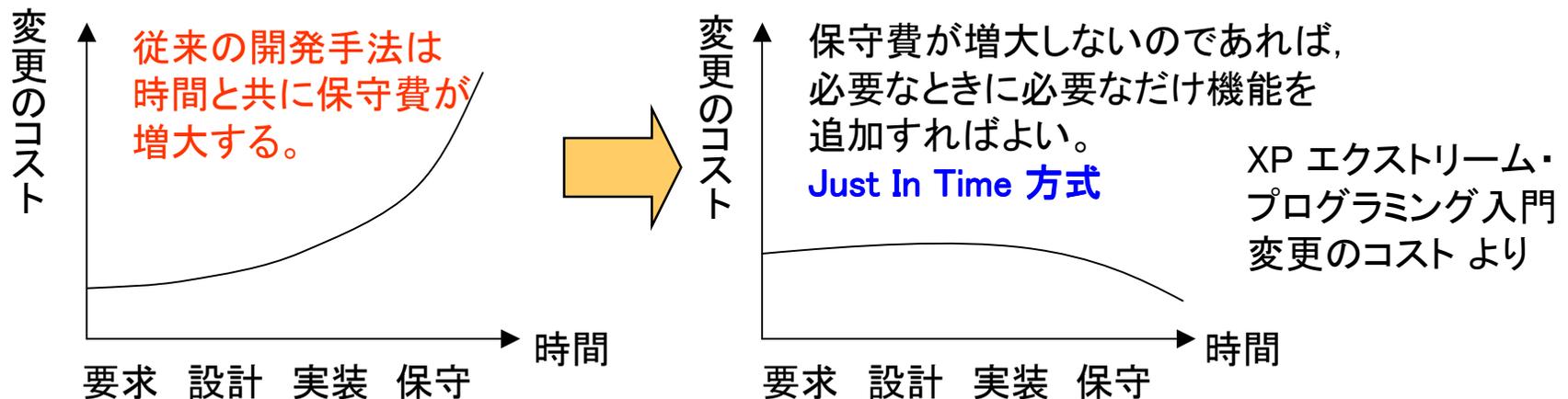
『**効率の追求から効果の追求**』では、設計-開発-顧客の**分業**による一律開発から脱却し、設計-開発-顧客が**一体**となって効果を追求することにより、**顧客満足度を向上**させる狙いがあります。



1. エンジンの特長

(2) 効果的な開発

現在のシステム開発手法では、システム設計時に「**将来に対する投資**」と称して、不要な設計や機能を組み込んでいます。技術進歩が早く、業務方法や市場環境の**変化が激しい**現代では、将来使われるかどうか判らない機能は**大きなリスク**であり、キャッシュフローや、スループットの観点からも、好ましいものではありません。『**今、必要な機能のみを実装する**』事が非常に重要です。

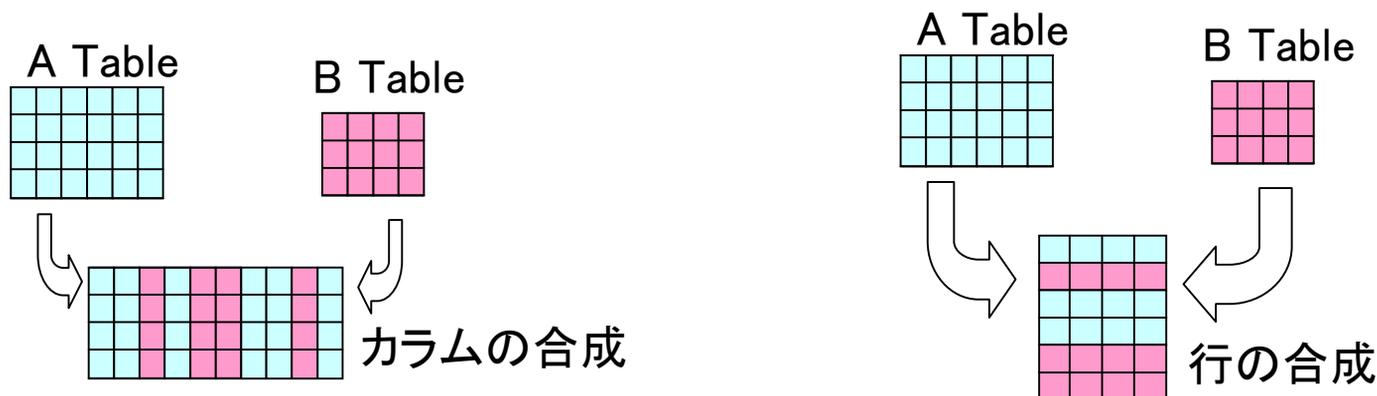


1. エンジンの特長

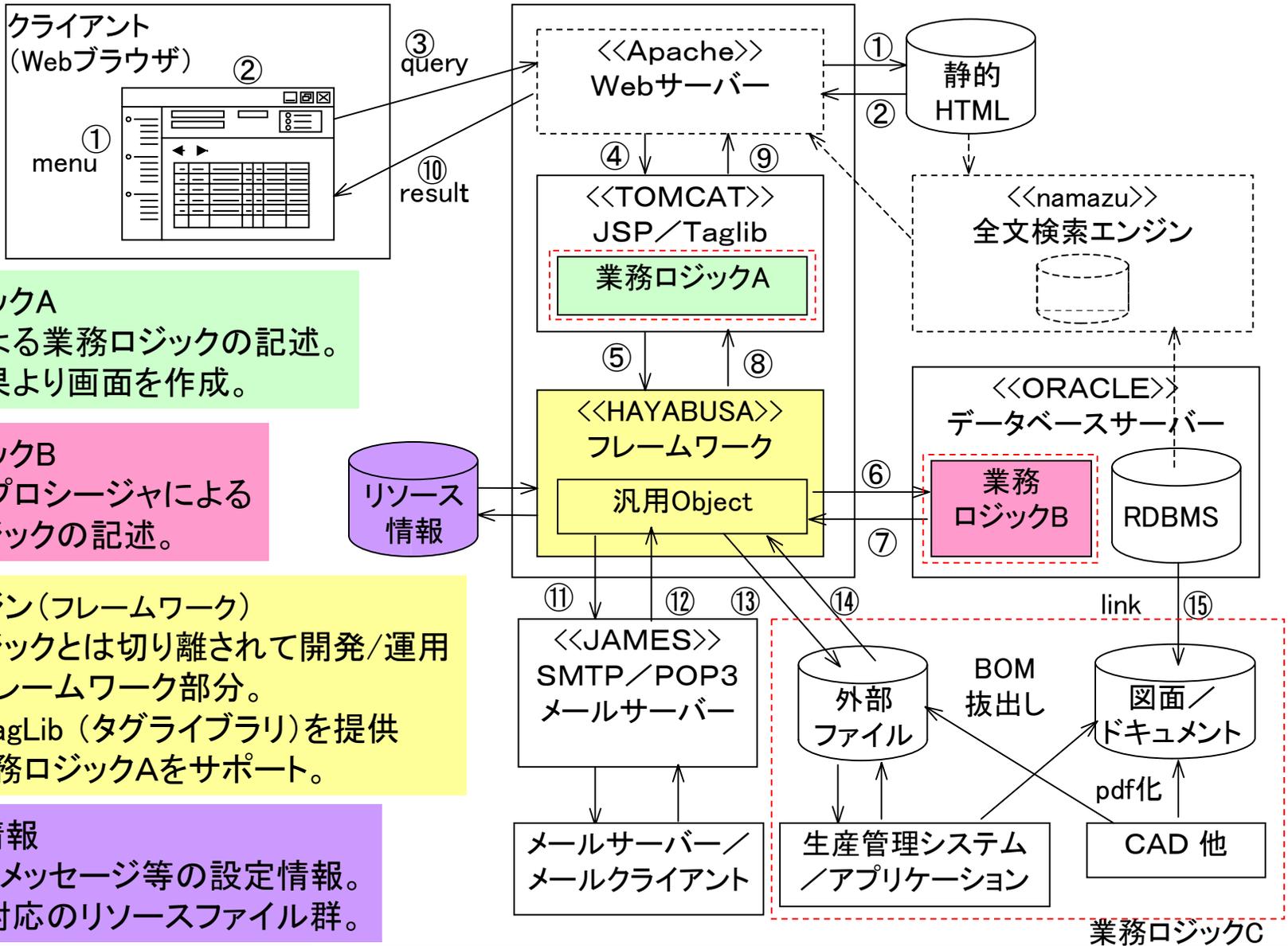
(3) RDBMSが中心

この中で、3番目の**RDBMSが中心**というキーワードが、最もこのエンジンを特徴付けているのかもしれませんが。

現在のWebアプリケーションの中心は**J2EE**です。**EJB**だ、**MVC**だと騒がしのですが、RDBMSで扱う表形式というのは、人が最も理解しやすい形式であり、かつ、非常に強力で柔軟だと思います。これを**最大限活用**する為、**RDBMS中心**の設計としました。



2. システム構成



業務ロジックA

SQLによる業務ロジックの記述。
検索結果より画面を作成。

業務ロジックB

ストアドプロシージャによる
業務ロジックの記述。

Webエンジン(フレームワーク)

業務ロジックとは切り離されて開発/運用
できるフレームワーク部分。
JSPのTagLib (タグライブラリ)を提供
して、業務ロジックAをサポート。

リソース情報

カラムやメッセージ等の設定情報。
国際化対応のリソースファイル群。

業務ロジックC

2. システム構成

システムを構成する4つのロジック(業務ロジックA, B, リソース, エンジン)は、下記方針を実現する為に、最適に配置されています。

- (1). リソース情報の分離
- (2). 多機能カラム情報
- (3). 画面／機能のモジュール化
- (4). 汎用オブジェクトによる拡張性

2. システム構成

(1) リソース情報の分離

ラベル、メッセージなどの国際化対応の項目や、**カラム**定義、**画面**定義、**ユーザー**定義、**システム**定義など、定義情報をリソースとして**画面開発と分離**しています。

これにより、基本設計時にデータベース定義情報を利用して基本的な画面を短期間で作成することが可能です。しかも、全画面において基本情報はすべて考慮されている状態になります。

(『**短納期志向**』)

その後、必要な画面を必要なだけ、機能アップしていけば、『**効果的な開発**』が、可能になります。

2. システム構成

(2) 多機能カラム情報

データベースの**カラム名**を**オブジェクト化**し、多機能情報を与えることにより、**select文**を記述する(『**業務ロジックA**』)だけで、基本的な画面を作成することが出来ます。

カラム情報は、先のリソース情報で与える為、画面設計時には、詳細指示を出す必要がありません。また、開発途中で、機能強化を行う場合も、リソース情報を書き換えるだけで、全画面に対してその機能を有効にする事が可能になります。

簡易的なカラムチェックは、多機能カラム情報を利用している為、**業務ロジックA**や**B**には、本流の流れのみを記述できます。

2. システム構成

(2) 多機能カラム情報

多機能カラム情報は、SQL文 (SELECT PN, **CDK**, NM FROM RK08) や、ColumnTag のname属性の『**CDK**』というカラム名より、カラム情報が自動的に集められて、多機能カラムオブジェクトを構築します。

DBカラムオブジェクトのラベルを使用。
これは、ラベルリソースより、構築される。

入力フィールドは、DBカラムオブジェクトの入力属性により指定され、その属性がコードリソースを使用する場合は、コードリソースを利用してプルダウンメニューを作成する。

品番 <input type="text"/>	工場C <input type="text"/>	<input type="button" value="検索"/>
品番	工場C	品名
D40A	O:大分	パネル
D40B	O:大分	プリンタ
D40C	A:海外	電源
Z06X	K:加賀	フレーム
Z06Y	S:犬山	ブラケット

ColumnTag の name属性に『**CDK**』を指定する。
全体は、DBカラムオブジェクトにより作成される。

QueryTag のSELECT文に、『**CDK**』を指定する。

表示は、ViewFormTag にて『**CDK**』をキーに表示する。

2. システム構成

(3) 画面／機能のモジュール化

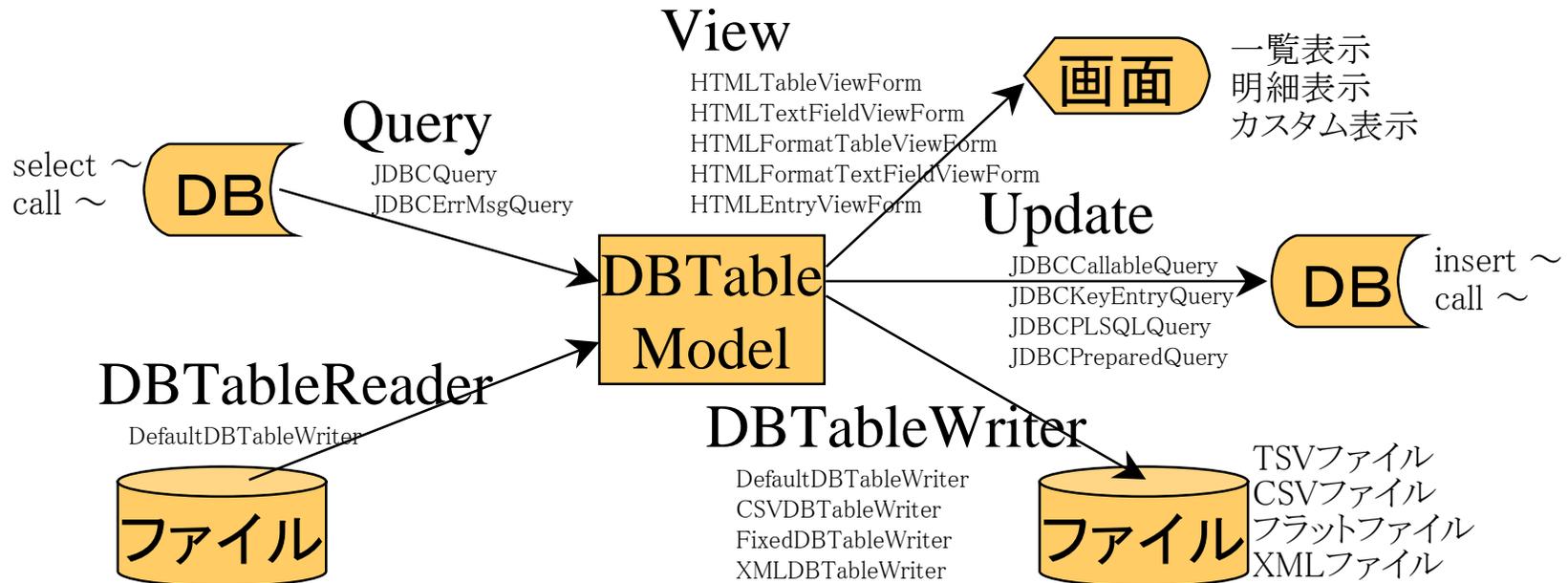
画面(『業務ロジックA』)は、画面IDと、実フォルダが仮想的に対応しています。そのため、フォルダ毎に画面を開発していくことで、フォルダのコピーによる機能追加や、他の画面を修正せずに、顧客対応のカスタマイズが可能になります。

また、画面系のロジック(『業務ロジックA』)と、業務系のロジック(『業務ロジックB』)も分かれているため、各々の機能をモジュール化しておくことで、カスタマイズが容易になり、基幹業務を流用することが可能になります。

2. システム構成

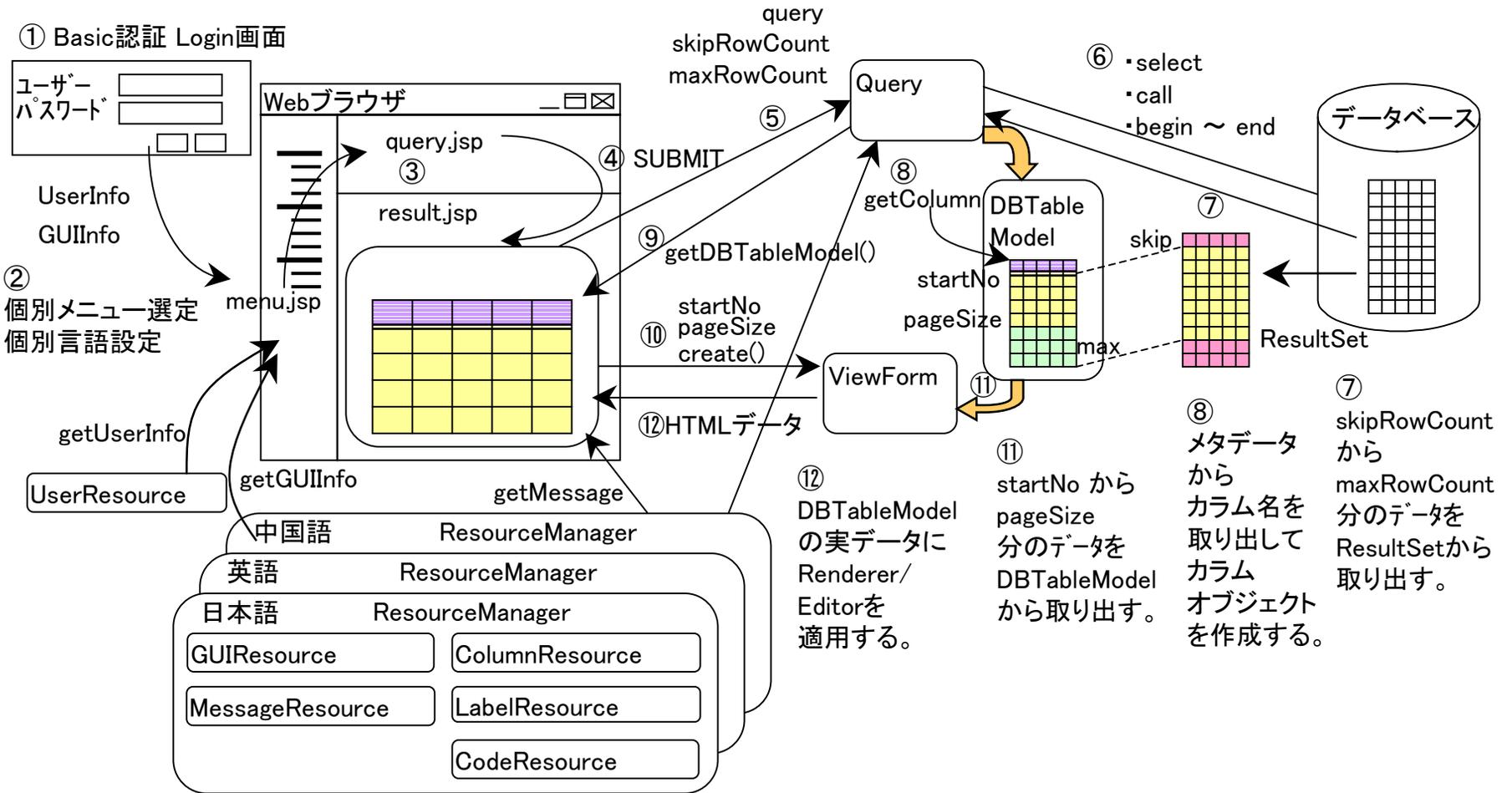
(4) 汎用オブジェクトによる拡張性

Webエンジンは、汎用オブジェクトインターフェース (Query、View、DBTableModel、Renderer、Editor、DBType など) を用いて開発しています。そのため、各インターフェースを継承したサブクラスを作成するだけで、容易に新機能を実装することが可能です。



3. データフロー例(検索系)

検索系のデータフロー例を示します。



4. 将来構想 (ロードマップ)

