



INTER-Mediatorの概要

INTER-Mediator Meet-up 2018

2018年11月8日

第2オカモトヤビル貸会議室

新居雅行 Masayuki Nii, 博士(工学)

 nii@msyk.net  [msyk_nii](https://twitter.com/msyk_nii)  [msyknii](https://facebook.com/msyknii)

Agenda

- FileMaker開発者が考慮すべきソリューションのWeb公開
- INTER-Mediatorを使ったWebアプリケーション開発
- INTER-Mediatorを使ったWebシステムの開発プロセス
- データベースにFileMaker Serverを使うメリット

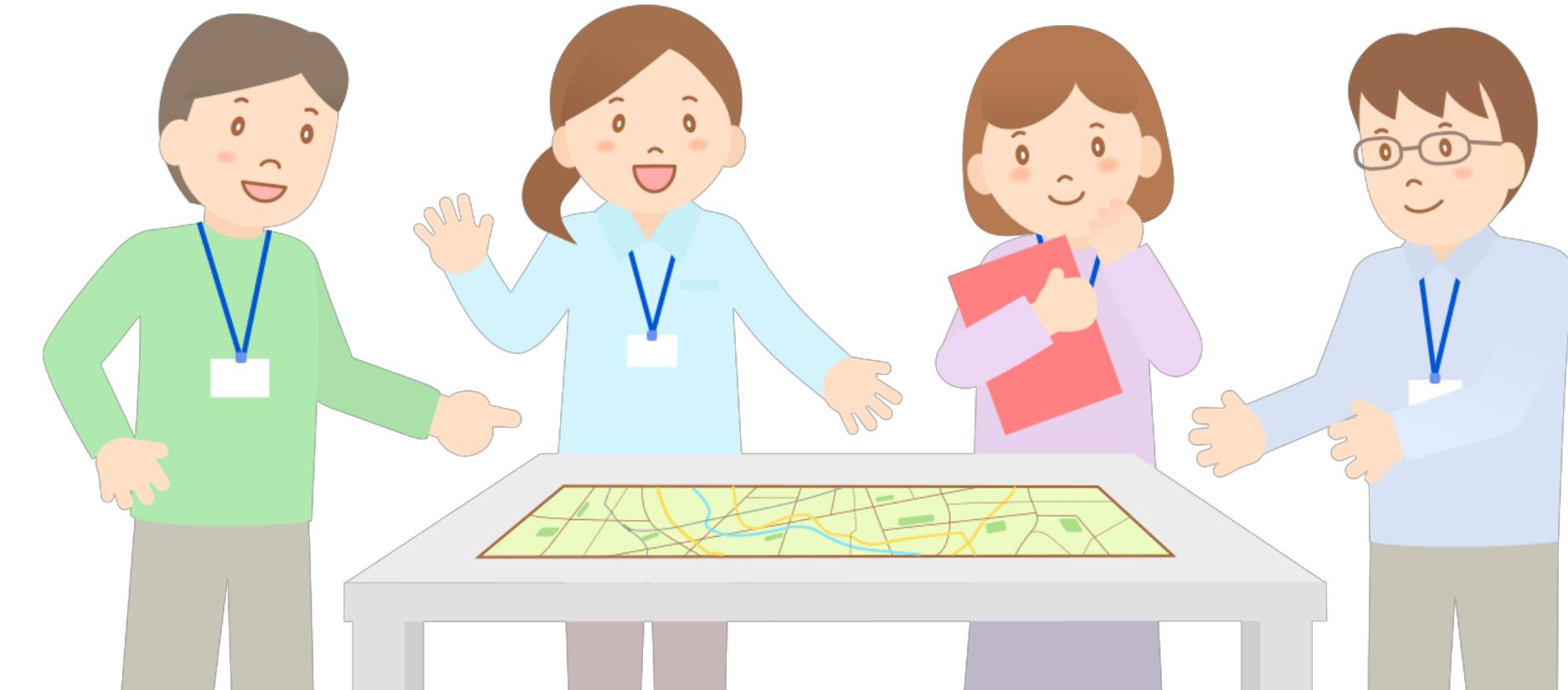
Talking by

- 講演者: 新居 雅行 (にい まさゆき) 博士 (工学)
- システム開発エンジニア、コンサルタント
 - ライフマティックス株式会社所属
 - FileMaker、LAMPベースでのWebアプリケーション、macOS/iOSネイティブアプリケーション開発、Androidアプリケーション
 - 開発や保守の効率が良いデータベース連動Webアプリケーション開発フレームワーク「INTER-Mediator」をオープンソースで開発、様々なソリューションで利用
- 教育、トレーニング
 - 国立情報学研究所特任研究員 トップエスイー講師・スタッフ
 - Apple認定トレーナー、Apple認定システムアドミニストレーター
 - 慶應義塾大学、明治大学、東京理科大学などで非常勤講師を務め、プログラミングやWeb開発に関する講義を受け持つ

FileMaker開発者が考慮すべき ソリューションのWeb公開

FileMakerでのWeb開発のニーズ

- FileMaker ProとWebでのハイブリッド運用
 - 例：社内で使う業務ソフトをFileMaker Proで運用しているが、一部の業務は外部からも実施できるようにしたい
 - 例：社外に向けてサービスを提供したいが、人数は多くてもアクセス頻度が低いので、FileMaker Serverで運用し、社内業務はFileMaker Proでやりたい
- FileMaker GOでの運用に適さない場合
 - 社外の不特定のユーザーがデータベースを利用する
 - PCは導入されたがiPadまで導入する予算がなかった



ニーズを満たすための方策と問題点

- FileMaker GO
 - 非常に良いソリューションだがライセンスに縛られるため、社員向け以外には難有
 - タブレットはもう一回再定義しないといけないのでは？
- Web Direct
 - だいぶんと性能は上がってきた。しかしライセンスに縛られる
 - JavaScriptのライブラリなどのWebエコシステムの解決策を適用できない
- Custom Web
 - PHPやRuby等でのプログラミング言語での開発が必要、FileMakerの手軽さが損なわれる
 - FileMaker社はいつまでこの機能を残すのかは非常に不安である
- FileMaker Data API
 - PHPなど何らかのプログラミング言語での開発が必要、FileMakerの手軽さが損なわれる
 - 実績が少なく、何となく心配 (FMDATAAPIがあるんですが→<https://github.com/msyk/FMDATAPI>)
- JDBC/ODBC経由での利用
 - やはりプログラミングが必要
 - 接続などレイヤーが増えるため管理が大変になる



昔はCDMLがあったのに…

- HTMLに記述を加えてWebページが作成できた
 - 独特な記述ながら、データベース連動できた
 - テーブルの内容を表示することは非常に簡単にできた
- なぜ、廃れたか？
 - データの更新のためにはフォームを組む。読み取ったデータで修正して書き込むというあたりの記述はかなり難しくなる
 - 複雑な処理が絡んでくると、データベース連動のための仕組みとの兼ね合いが困難
 - ポータルのない時代だけに、関連レコード処理が貧弱
 - 「パスワード」だけのデータベースセキュリティであり、十分なセキュリティかは疑問があった
- FileMakerのような手軽さがなかった
 - FileMakerであるにもかかわらず…

```
<html>
<head>
    <title>Store Search Results</title>
</head>
<body>
    Your search found [FMP-CurrentFoundCount] re
<table>
    <tr>
        <th>Item Name</th>
        <th>Category</th>
        <th>Description</th>
        <th>Price</th>
    </tr>
    [FMP-Record]
    <tr>
        <td>[FMP-Field:product_name]</td>
        <td>[FMP-Field:product_category]</td>
        <td>[FMP-Field:product_description]</td>
        <td>[FMP-Field:product_price]</td>
    </tr>
    [/FMP-Record]
</table>
</body>
</html>
```

そこで、INTER-Mediator

- FileMakerでの開発の快適さをWebに持ち込む
 - ▶ HTMLの属性追加、設定ファイルの作成で、データベースとテキストフィールドなどが自動的に連動し、表示、入力、編集を実現
- FileMaker開発の拡張性を凌ぐ
 - ▶ ユーザーインターフェースの高度化は、JavaScriptのプログラム
 - ▶ サーバー側のデータ処理の拡張は、PHPのプログラムで対応可能
- Web開発の最新の成果を利用できる
 - ▶ レスポンシブルページを作るための代表的なCSSフレームワークであるBootstrapと一緒に利用できるなど、Webページ向けの様々な素材と同時に利用できる
- メンテナンス開発の問題点を軽減
 - ▶ 記述がシンプルなので、メンテナンス開発も作成同様に手軽に進められる

The screenshot shows a website for 'Topse.jp' with a navigation bar at the top. The main content area is titled 'カリキュラム' (Curriculum) and contains a grid titled '講義時間割 (2018年度)' (Lecture Schedule for the 2018 Academic Year). The grid shows a weekly schedule from Monday to Friday for each month from April to December. Each cell in the grid contains a subject name. The subjects include: 基礎理論 (Foundation Theory), オブジェクト指向分析設計 (Object-oriented Analysis and Design), セキュリティ概論 (Security Fundamentals), ビッグデータIT基盤 (Big Data IT Foundation), 形式仕様記述 (基礎・VDM編) (Formal Specification (Basic - VDM Edition)), コンポーネントベース開発 (Component-based Development), 要求工学基礎 (Requirements Engineering Foundation), 要求工学先端 (Advanced Requirements Engineering), 安全要求分析 (Safety Requirements Analysis), 形式仕様記述 (Bメソッド編) (Formal Specification (B Method Edition)), 形式仕様記述 (実践編) (Formal Specification (Practical Edition)), 軟件アバターン (Software Avatar), 問題指向要求分析 (Problem-oriented Requirements Analysis), 性能モデル検証 (Performance Model Verification), アスペクト指向開発 (Aspect-oriented Development), 軟件再利用演習 (Software Reuse Practice), and 軟件開発実践演習 (Software Development Practice).

INTER-Mediatorを使った Webアプリケーション開発

Webページの稼働に必要なこと

- 前提条件
 - ▶ PHPが動作するWebサーバー
 - ▶ データベースには、FileMaker Serverにデータベースが設置されているなど、スキーマが適用されたデータベースがすでに存在する
- データベース連動ページに必須の2つのファイル
 - ▶ ページファイル
 - HTMLファイルで記述し、一部に独自の属性を記述
 - ▶ 定義ファイル
 - キーと値を与える形式の設定ファイル
 - PHPファイルで作成するが、配列による記述なので、宣言的な記述である
- ファイルをサーバー上に配備して、ブラウザーから開く

定義ファイルの例

定義ファイルエディタでの設定

The screenshot shows the 'INTER-Mediator - Definition File Editor' interface. In the 'Target Definition File' section, the file path is set to `.../def03.php`. The 'Contexts' section contains a table with the following rows:

name	person	table	person	view	person	
	key	id	sequence		paging	true
	repeat-control	confirm-insert confirm-d	navi-control		records	1
	maxrecords	100	cache		soft-delete	

Below the table, there are sections for 'Query', 'Sorting', and 'Relationship'. A green callout box points to the 'Contexts' table with the text: '定義ファイルを表示、編集するためのエディタ Webアプリも添付'.

定義ファイルの中身はPHPのコード

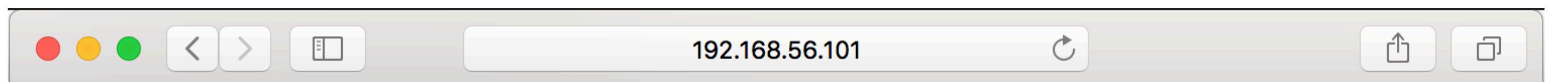
```
<?php
require_once('INTER-Mediator/INTER-Mediator.php');

IM_Entry(array (
  0 =>
  array (
    'name' => 'person',
    'table' => 'person',
    'view' => 'person',
    'records' => 1,
    'maxrecords' => 100,
    'paging' => true,
    'key' => 'id',
    'repeat-control' => 'confirm-insert confirm-delete',
    'validation' =>
    array (
      0 =>
      array (
        'field' => 'name',
        'rule' => 'length(address)>0',
        'message' => '空欄はダメです！',
        'notify' => 'inline',
      ),
    ),
  ),
),
array (
),
array (
),
array (
  'db-class' => 'PDO',
  'dsn' => 'mysql:host=localhost;dbname=test_db;charset=utf8mb4',
),
false);
```

プログラムを書くではなく、決められたキーに対する値をPHP言語で記述する

FileMakerのTOに相当する「コンテキスト」を定義し、ページファイルからはコンテキストを指定してデータベース利用する

ページファイルの例



```
1 <!DOCTYPE html>
2 |<html lang="ja">
3 |<head>
4 |  <meta charset="UTF-8">
5 |  <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 |  <title></title>
7 |  <script type="text/javascript" src="def03.php"></script>
8 |</head>
9 |<body>
10 |<div id="IM_NAVIGATOR"></div>
11 |<table>
12 |<tbody>
13 |<tr><th>id</th><td data-im="person@id"></td></tr>
14 |
15 |<tr>
16 |<th>name</th>
17 |<td><input type="text" data-im="person@name"></td>
18 |
19 |<tr>
20 |<th>mail</th>
21 |<td><input type="text" data-im="person@mail"></td>
22 |
23 |</tbody>
24 |</table>
25
26 |<tbody data-im-control="post">
27 |<tr>
28 |<th>name</th>
29 |<td><input type="text" data-im="person@name"></td>
30 |
31 |<tr>
```

純粋なHTMLファイル、独自拡張なし

定義ファイルをscriptタグで読み込み

ページネーションをこのIDの要素に生成

コンテキスト名@フィールド名でバインド
すなわち、データベースと連動する

様々なコントール、ここは入力専用モード

データベース連動アプリケーションの基本機能

- バインド
 - ▶ データベースのフィールドと、ページ上の要素を論理的に結合
 - ▶ フィールドの値が要素の値やその属性値として設定可能
 - ▶ フォーム要素で修正するとデータベース更新し、楽観的ロックが常に機能
- 繰り返し
 - ▶ 1レコード分のページ要素を、レコードの数だけ繰り返して、リスト表示
 - ▶ ページネーションの自動生成
- リレーションシップ
 - ▶ 内包する領域に、関連レコードをさらに展開（何段階でも内包可能）
 - ▶ FileMakerで言えば、ポータルの中にポータルを展開できるようなもの
- レコード作成、削除、複製
 - ▶ 設定により適切な位置に自動的にボタンが作成される

計算プロパティ、バリデーション

- ・ 計算プロパティ = FileMakerの計算フィールド
 - ▶ 計算式を与えて、計算結果が得られる新たなフィールドを定義する
 - ▶ 同じレコードだけでなく、ページ内に展開している他のテーブルも参照可能
 - ▶ 文字列、数値だけでなく、日付や時刻、リストの計算処理関数も用意
 - ▶ クライアント側での計算処理のため、フィールド更新が即座に反映できる
- ・ バリデーション
 - ▶ フィールド単位に式で判断し、ダイアログボックスやページ上にエラーメッセージを表示する
 - ▶ 複数のフィールドにまたがる判断は、JavaScriptを利用する
 - ▶ ユーザーインターフェース側のみで処理をしており、サーバーへの通信上では処理をしていない点に注意が必要／将来バージョンでサポート

高度なユーザーインターフェースを簡単に実現

- ページネーション
- マスター/ディテール形式のUI
 - ▶ 切り替わり、左右に配置
- テーマ機能
 - ▶ 既定のスタイルでも十分に視認性の高いページが作成される
 - ▶ 独自にテーマスタイルを定義可能
- ファイルアップロード
 - ▶ 独自UIによるドラッグ&ドロップや、ファイル選択
 - ▶ JQuery-FileUploadを利用したプログレス表示機能付きコンポーネント
- 同時参照時の即時更新
 - ▶ クライアント間の同期、Pusherを利用

検索フィールド、ソートボタン

- 検索フィールド
 - ▶ 入力してリターンを押すと…
 - ページ上に展開しているpostalcodeコンテキストを更新
 - その時に、指定したフィールドの値について、OR検索
 - テキストフィールドの値が含まれるレコードに絞り込む
- ソートボタン
 - ▶ 三角をクリックすると…
 - ページ上に展開しているpostalcodeコンテキストを更新
 - その時に、指定したフィールドでソートする設定を含めてデータベース検索
- ポップアップメニューから1ページのレコード数指定も可能

The screenshot shows a web browser window for 'demo.inter-mediator.com'. The title bar says 'INTER-Mediator For Database-driven Web Application'. The main content area is titled '検索機能のサンプル (設定のみで実現)'. It features a search form with '検索条件:' and '表示件数 : 4' dropdown, followed by a 'search' button. Below the form is a table with four columns: 郵便番号▲▼, 都道府県, 市区町村, and 町域名▲▼. The table contains five rows of data:

郵便番号▲▼	都道府県	市区町村	町域名▲▼
1000000	東京都	千代田区	以下に掲載がない場合
1020072	東京都	千代田区	飯田橋
1020082	東京都	千代田区	一番町
1010032	東京都	千代田区	岩本町

The screenshot shows a web browser window for 'demo.inter-mediator.com'. The title bar says 'INTER-Mediator For Database-driven Web Application'. The main content area is titled '検索機能のサンプル (設定のみで実現)'. It features a search form with '検索条件: 調布市' and '表示件数 : 4' dropdown, followed by a 'search' button. Below the form is a table with four columns: 郵便番号▲▼, 都道府県, 市区町村, and 町域名▲▼. The table contains five rows of data:

郵便番号▲▼	都道府県	市区町村	町域名▲▼
1820000	東京都	調布市	以下に掲載がない場合
1820004	東京都	調布市	入間町
1820035	東京都	調布市	上石原
1820007	東京都	調布市	菊野台

宣言的な記述で実現できる認証・認可

- データベースの内容をやりとする段階で認証を行う
 - ▶ ページファイル自体のやりとりでの認証はWebサーバーに依存
- カスタマイズ可能なログインパネル
 - ▶ ユーザーを管理するテーブルを利用、グループ管理も持つ
 - ▶ データベースシステム自体のアカウントを利用することができる
 - ▶ LDAP、OAuth2にも対応（Active DirectoryやGoogleアカウント対応実績あり）
- 認可の設定
 - ▶ コンテキスト／データベース操作／レコード単位での認可が可能
- その他の機能
 - ▶ 画像やPDFファイルなどのファイルに対するアクセス権の設定も可能
 - ▶ パスワードポリシーの設定
 - ▶ オンライン登録、オンラインでのパスワードリセット

そのほかの宣言的な記述で実現できる機能

- メール送信
 - ▶ データベース処理後にメール送信、テンプレートやフィールドの値の利用
- ブラウザー判別
 - ▶ 非対応ブラウザーの場合にメッセージを表示
- 集計処理を含むSQLステートメントの記述（FileMaker Server非対応）
 - ▶ GROUP-BYやJOIN等を含むSQLによるコンテキストの取得
- 論理削除
 - ▶ 特定のフラグとなるフィールドを利用して、レコード削除をしないで削除をしたかのように振る舞う手法（ソフトデリート）

クライアントサイドのプログラミング

- JavaScriptで実装
- APIの例
 - ▶ ページに展開したデータを取得、展開データを変更してサーバーに反映
 - ▶ データベース処理を直接行う
- 様々な実行タイミングの用意
 - ▶ ページ生成前後
 - ▶ 1テーブルの展開直後、1レコードの展開直後
 - ▶ バリデーションの前後
- ユーザーインターフェースコンポーネント
 - ▶ ファイルアップロードや、HTMLエディタなどを用意
 - ▶ 独自に開発可能

サーバーサイドのプログラミング

- PHPで実装する
- 既存実装機能の拡張や変更
 - ▶ フィールド単位の「データコンバータ」、DBアクセスのためのクラスの自作
- データベース処理の前後にプログラムを追加
 - ▶ アスペクト指向的な動作
 - ▶ データベース処理と絡めて追加の処理を行う
 - ▶ 例：APIを叩くシステムの場合
 - クライアントでボタンを押して、データベースに新規レコードを作るようとする
 - 新規レコード前の処理にAPI呼び出しを行う
 - API呼び出した結果をフィールドに追加することで、新規レコードにはAPI呼び出しの結果が残る

開発事例：ふち無しはがき印刷本舗

- FileMaker Server
 - ▶ 社内はFileMaker Pro
- オンデマンド印刷
 - ▶ 顧客ごとに変わる注文内容を管理
 - ▶ CRM機能により、修正の依頼や素材の受け渡しがスムーズに

ふち無しはがき印刷本舗

官製ハガキのフチ無し印刷専門サービス

ふち無しはがき印刷本舗

ホーム HOME 利用ガイド GUIDE フチ無しの特徴 SALES POINT データの作成 How to make

印刷メニュー

- ふち無し年賀状
- ふち無し暑中見舞い
- ビジネス挨拶はがき
- ふち無し挨拶はがき
- ふち無し喪中葉書・寒中見舞い

お助け情報

- 無料画像・テンプレート集
- 年賀状の書き方辞典
- 暑中見舞いの書き方辞典
- 喪中・寒中見舞いの書き方
- ふち無しはがきニュース

インフォメーション

- よくある質問

お電話は 048-813-3003 平日10:00~18:00

ふち無しはがき印刷本舗 MYページ

このサイトの通信は暗号化(SSL)に対応しています。

SECURED BY RapidSSL www.rapidssl.com

048-813-3003 平日10:00~18:00

新居雅行 様専用ページ

ご注文名：暑中見舞いフォト宛名（商品番号：00266）の詳細

進捗状況

STEP1 お申し込み STEP2 ご入稿 STEP3 ご入金 STEP4 校正作業 STEP5 印刷 STEP6 発送

発注日 2013年06月13日 入金日 枝了日

ご注文内容

ご注文名：暑中見舞いフォト宛名
商品番号：00266
会社名：3360923
お客様名：新居雅行
フリガナ：にいまさううき
郵便番号：222-0003
住所：神奈川県
横浜市港北区
大曾根
メール：msyk@msyk.net
選択商品：暑中見舞い_フォトフレーム
印刷枚数：200
宛名印刷枚数：100

ご請求金額

商品名	単価	個数	金額
暑中見舞い（かもめ～る）	50	200	10,000
暑中見舞いはがき印刷	60	200	12,000
宛名印刷	18	100	1,800
宛名データ処理費	2,000	1	2,000
佐川急便	840	1	840

合計金額：¥26,640
うち消費税：¥1,269 (5%)

請求書の表示と印刷

ブラウザの【プリント】から印刷してください。

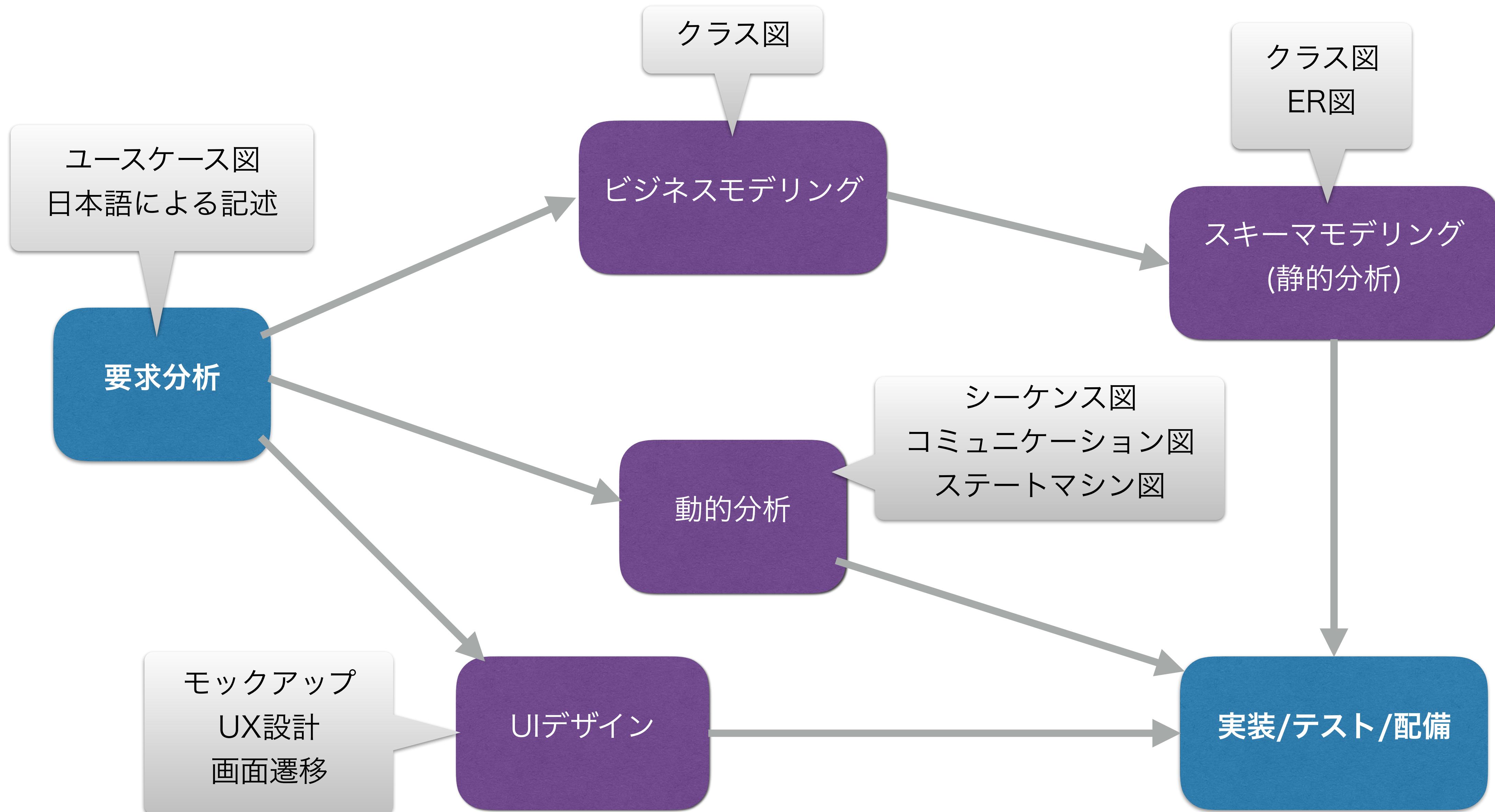
STEP2：ご入稿

1. お客様操作のご案内

ご注文では、はがきデザイン用のデータをご入稿ください。以下の作業が必要です。

INTER-Mediatorを使った Webシステムの開発プロセス

一般的なシステム開発の流れ



INTER-Mediatorでの実装作業

エンドユーザーとエンジニアの協業で効率的な開発を目指す

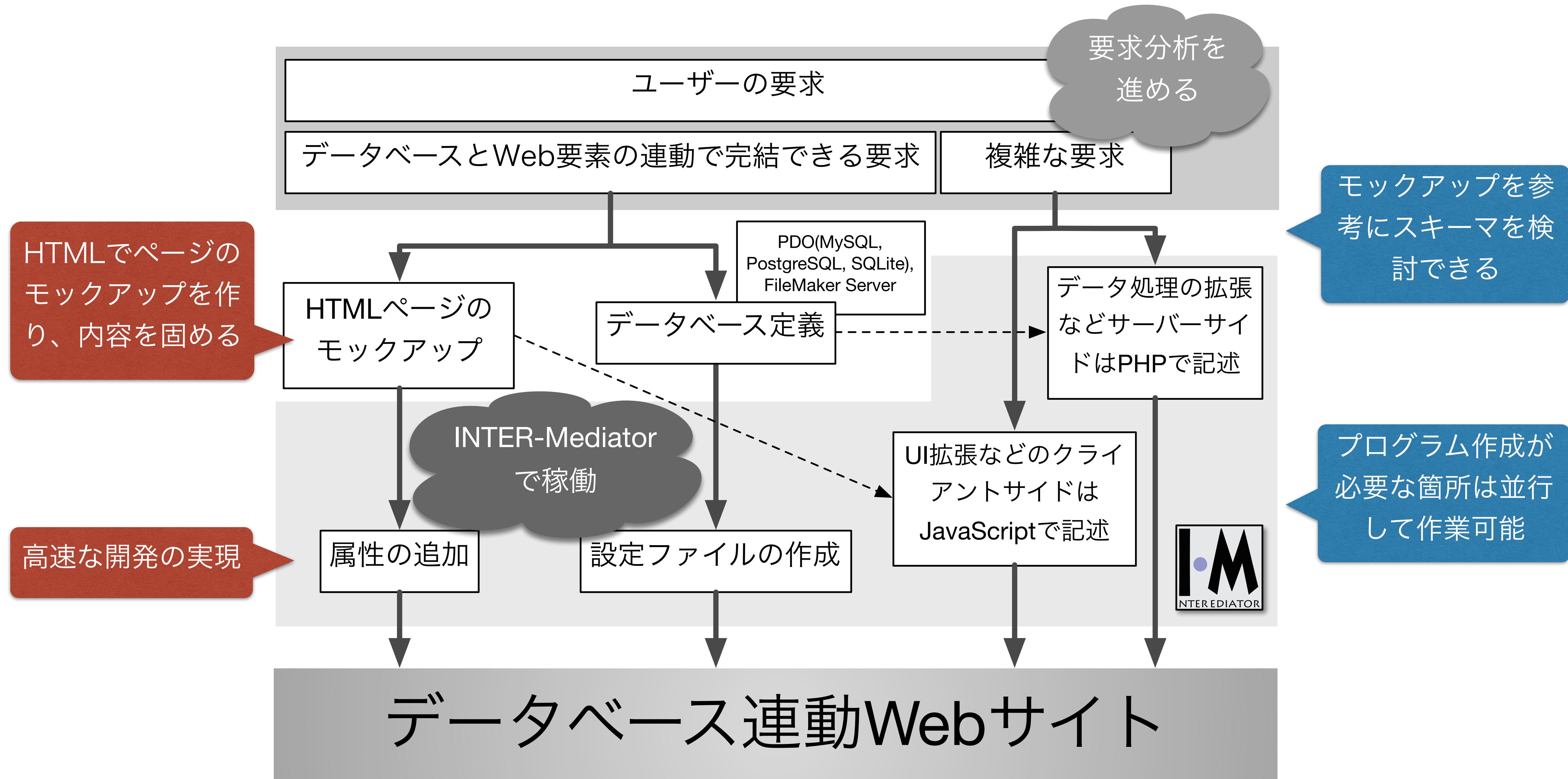
- ユーザーインターフェース
 - HTMLでUIデザインを行う（モックアップ駆動開発）
 - データベースと連動する部分を属性でマーキング
 - UIデザインに応じて定義記述を行い機能を呼び出す
- スキーマモデリング
 - データベース上に展開する
 - データベース利用できるように記述
- 動的分析結果
 - UIに関わる部分など、クライアント側での処理はJavaScriptで実装可能
 - サーバー処理に関わる部分はPHPで実装可能

エンドユーザーが
対応可能

複雑なものは
専門的な知識が必要

基本的に専門的な
知識が必要

INTER-Mediatorでの開発プロセス



開発プロセスに関するには？

ページのモックアップや
サイトデザインが
できるには

アプリケーションを
完成させるには

高度なユーザー
インターフェースや
ロジックの開発まで

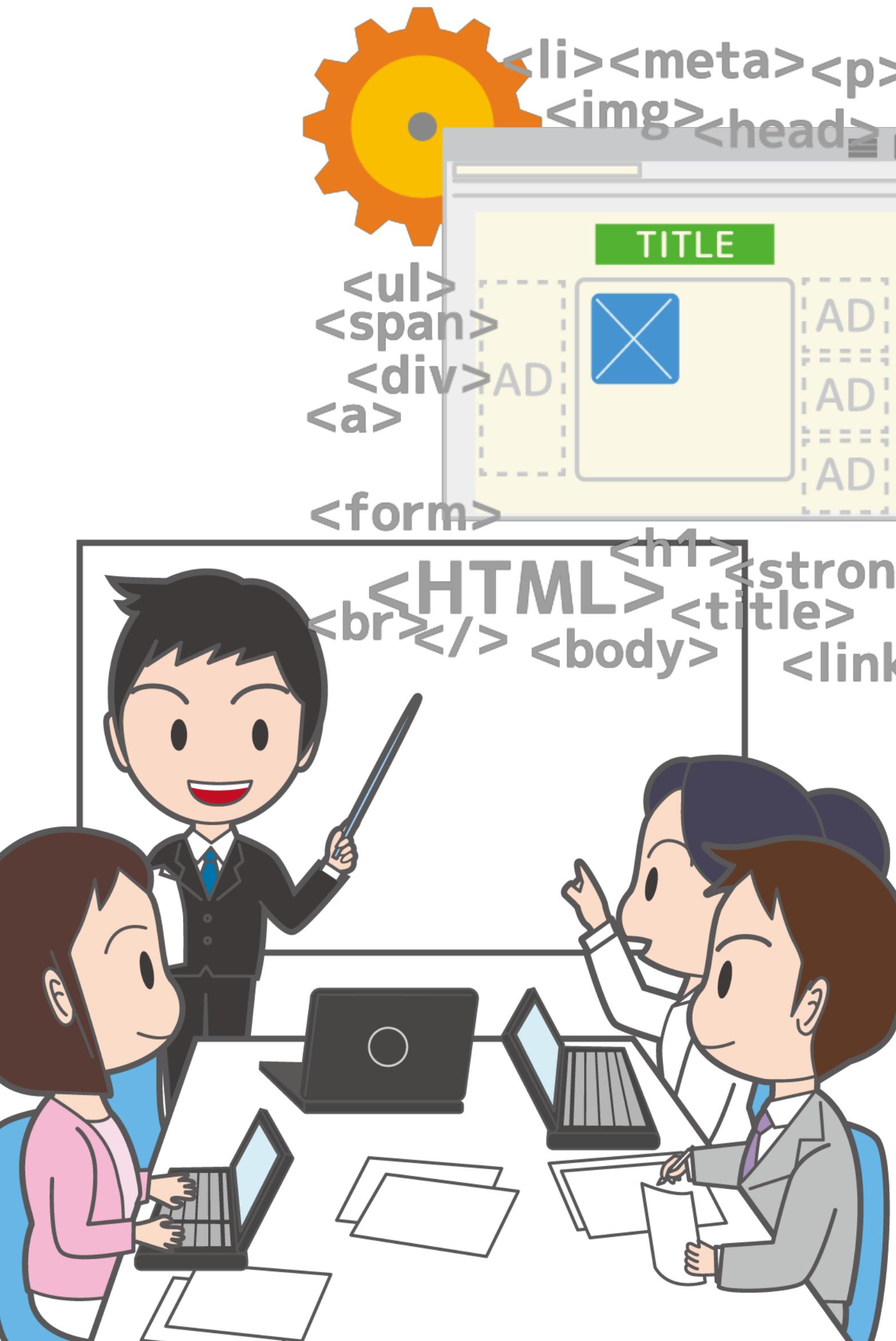
作成したWeb
アプリケーションを
安定して運用する

- HTML、CSSでのWebページの作成技術

- データベースに関して、設計までができる
- ページファイルや定義ファイルを自身で作成できる

- PHP、JavaScriptによる開発
- ネットワークやセキュリティの知識

- Windows ServerやLinuxを稼働させ、データベースを稼働させ、PHPを含むWebサーバーの稼働までができる



まとめ

- 宣言的な記述だけで、データベース連動したページの骨格を構築できる
 - 宣言的の反対は「手続き的」、例えばプログラミング言語は手続き的な記述が一般的
 - 学習時間が短く、メンテナンス性が良い傾向にある
- 複雑な仕組みはプログラミングで実現
- Web開発の素材やノウハウを利用できる
- FileMakerソリューションのWeb公開を効果的に実現：INTER-Mediator



<https://inter-mediator.com/>