

卒業研究

PHPによるGoogleMapAPI
を利用した
喜田村緑郎日記システムの
試作

平成19年2月10日
情報システム解析学科
谷研究室 新海 光

目次

- ◆はじめに・・
- ◆仕様
- ◆デモンストレーション
- ◆今後の課題

喜田村緑郎ってどんな人??

- ◆ 明治4年(1871)年-昭和36(1961)年
- ◆ 大正12(1924)年から昭和26(1951)年ほぼ毎年確認されている日記
- ◆ 女形の芸を完成した人物
- ◆ 食にうるさい人物

喜多村緑郎 を知ったきっかけ

- ◆ 学術フロンティアの一貫である文献をデジタルアーカイブ化する要求が国文学科からありました

喜多村緑郎 を知ったきっかけ

- ◆ **学術フロンティア**の一貫である文献をデジタルアーカイブ化する要求が国文学科からありました

学術フロンティア

- ◆ 文部科学省が、大学院研究科、研究所の中から、重点的研究領域ごとに、優れた研究実績をあげ、将来の研究発展が期待される卓越した研究組織のプロジェクトに対して選定し、総合的に支援するという事業のことである

喜多村緑郎 を知ったきっかけ

- ◆ 学術フロンティアの一貫である文献をデジタルアーカイブ化する要求が国文学科からありました

デジタルアーカイブとは??

- ◆ 歴史的・文化的資産や自然環境などを始め、様々な分野の情報をデジタル映像やデジタル文書として保存・蓄積したもの。

◆ 検索エンジンで有名な
Google社が地図WEBアプリ
ケーションGoogleMapAPIを
提供をしました

Google Map

特徴

ブラウザ上で地図を動かせる
航空写真、マップ、マップと航空写真の合成に切り替えることができる
デスクトップアプリケーションのように動く。

Google Map API

Google MAPを個人で扱えることができる関数集合



引用 <http://maps.google.co.jp/>

研究目的

喜田村緑郎日記



デジタルアーカイブ



Web上に地図を表示

では
実際試してみます

研究目的

喜田村 郵便

新しい発見や
日記を訂正したりするには
それなりの知識が
必要

Web上に地図を表示

研究目的

喜田村緑郎日記

喜田村緑郎が歩いた場所を
知識がない人でも簡単に登録、削除、修正
できる地図WEBアプリケーションシステムを
試作

編集可能な地図Webアプリケーションに
変更

目次

- ◆はじめに・・・
- ◆仕様
- ◆デモンストレーション
- ◆今後の課題

仕様

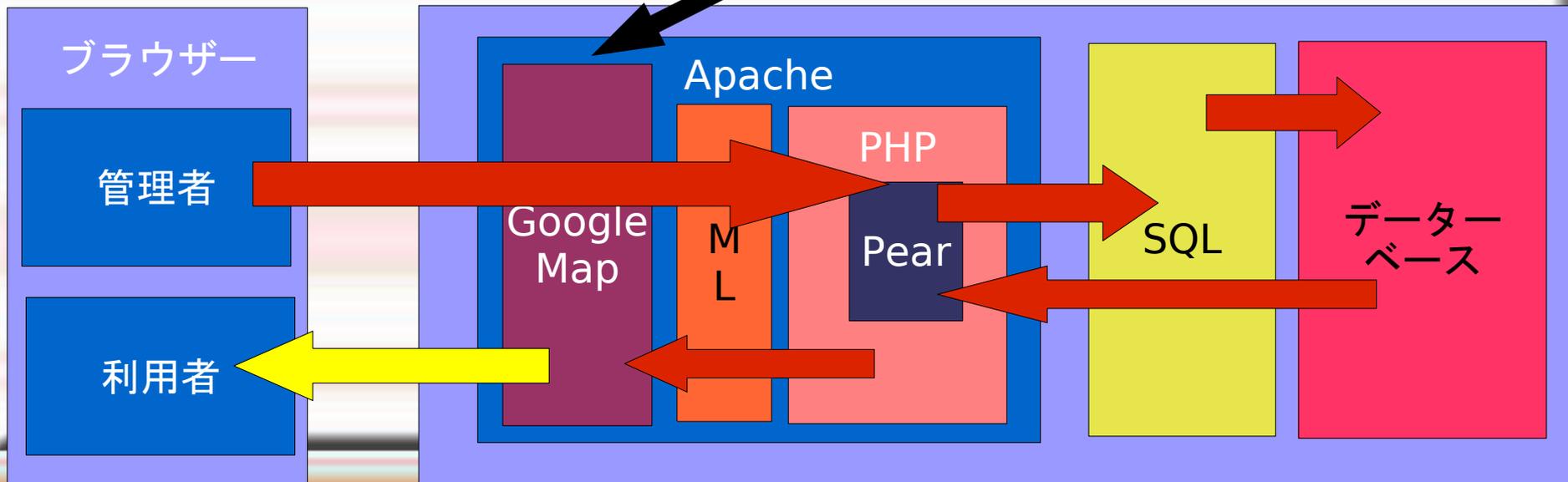
●今回試作する環境

- OS: Fedora Core 6
- Webサーバー : Apache
- HTML組込み言語 : PHP+Pear (拡張ライブラリ)
- データベース : MYSQL
- Web マップ : Google Map API

Google社

クライアント

サーバー



システム機能

◆ 管理ページ

- ◆ 用語

- ◆ 日記

- ◆ 場所

- ◆ ユーザー管理

◆ 公開ページ

- ◆ 用語

- ◆ 日記

- ◆ Google Map

目次

- ◆はじめに・・・
- ◆仕様
- ◆デモンストレーション
- ◆今後の課題

デモ内容について

- ◆ 喜多村緑朗が歩いた場所の登録、削除、修正
- ◆ 昭和6年4月26日データー 大野屋根
- ◆ 入力値
- ◆ 緯度、経度
- ◆ 場所
- ◆ 説明
- ◆
- ◆
- ◆ 出力
- ◆ 歩いた場所を地図上に表示し、歩いた場所を順に表示
- ◆
- ◆
- ◆

編集可能な喜多村緑郎日記
地図*WEB*アプリケーション

デモンストレーション
してみます

目次

- ◆はじめに・・
- ◆研究目的、背景
- ◆仕様
- ◆今後の課題

今後の課題

- ◆ 目的なデーターを探すために検索機能を付ける。
- ◆ 日付ごとにマップにポイントを表示できていない。

以上で
発表終わります
ありがとうございました

使われている技術

- GoogleMapAPI
 - Ajax
 - XML
 - DOM
 - 非同期通信
 - Javascript

Ajax

◆特徴

- ◆ ページの遷移なしで動的に変化してくれる
- ◆ デイスクトップアプリケーションの動作をインターネット上でも実現可能

XML

◆ 特徴

- ◆ 文書やデータの意味を構造的に記述可能
- ◆ 特定のソフトウェアに依存しないデータ形式である点
- ◆ ユーザーが独自のタグを指定できる
- ◆ 人間、ソフトウェア両方とも理解しやすい

XML

- ◆ 名前 新海 光
- ◆ 学籍番号 5 4 0 3 0 1 9
- ◆ クラス 3 8

属性

XML

<?xml version="1.0" encoding="UTF-8" ?>

<students id="5403019" >

<名前>新海 光</名前>

<クラス>38</クラス>

</students>

開始タグ

終了タグ

要素

DOM(Document Object Model)

◆特徴

- ◆XMLにアクセスするためのAPI
- ◆**DOMTree**という木を作りXML文章内のデータの順番に関係なく自由にアクセスできる

DOM(Document Object Model)

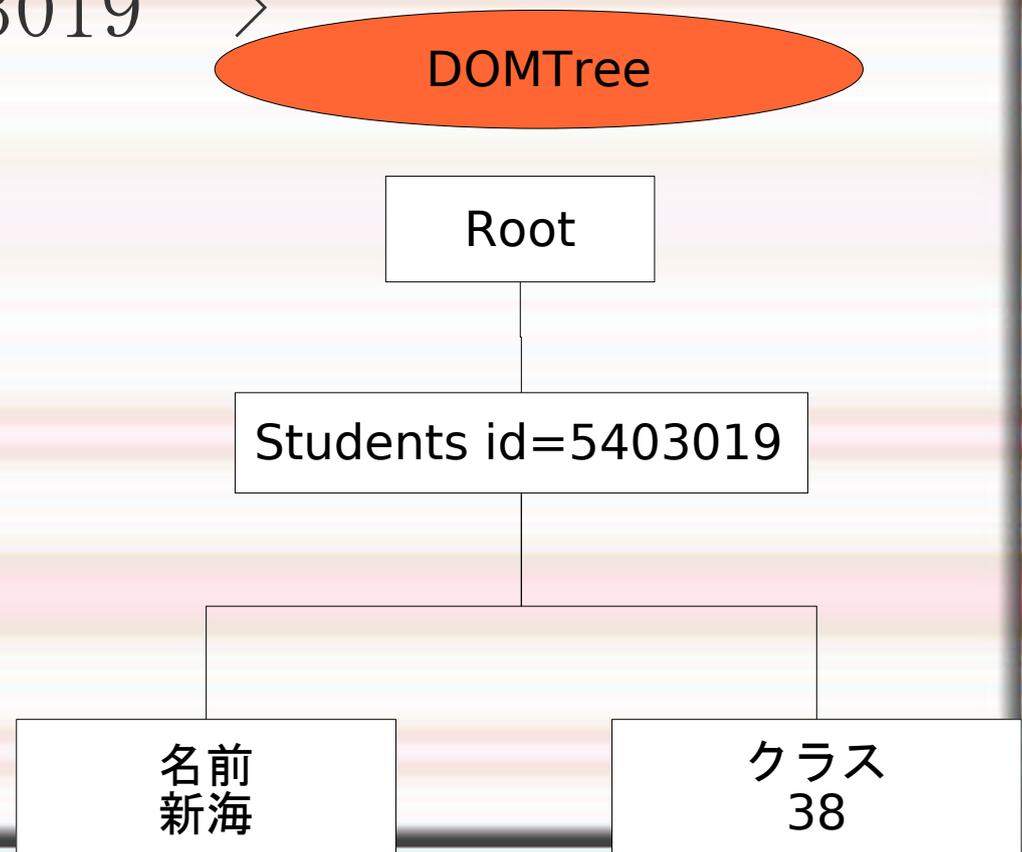
```
<?xml version=" 1.0" encoding=" UTF-8" ?>
```

```
<students id=" 5403019" >
```

```
<名前>新海 光</名前>
```

```
<クラス>38</クラス>
```

```
</students>
```



DOM(Document Object Model)

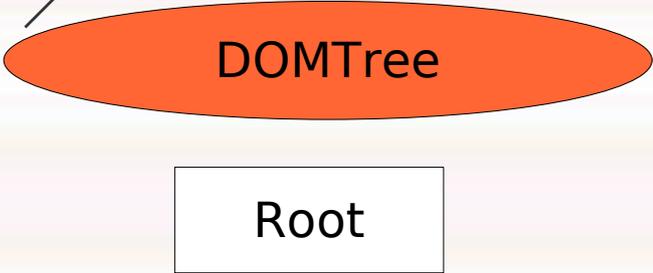
```
<?xml version=" 1.0" encoding=" UTF-  
8" ?>
```

```
<students id=" 5403019" >
```

```
<名前>新海 光</名前>
```

```
<クラス>38</クラス>
```

```
</students>
```



DOMTree

Root

DOM(Document Object Model)

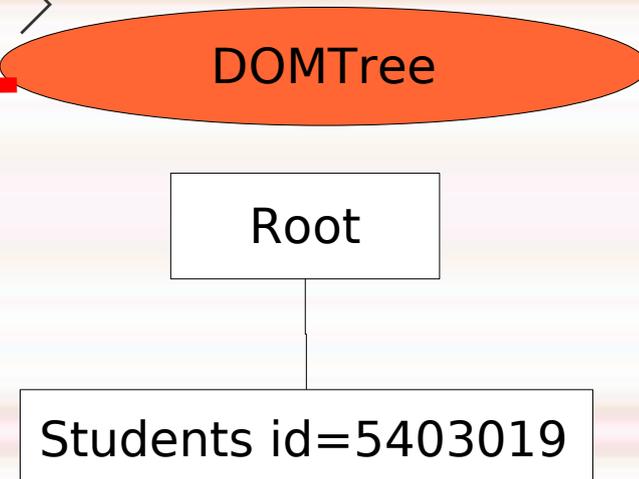
```
<?xml version=" 1.0" encoding=" UTF-8" ?>
```

```
<students id=" 5403019" >
```

```
<名前>新海 光</名前>
```

```
<クラス>38</クラス>
```

```
</students>
```



DOMTree

The diagram illustrates the DOM tree structure. It starts with a root node labeled 'Root' in a box. A vertical line connects the root to a child node labeled 'Students id=5403019' in a box. A red oval labeled 'DOMTree' is positioned to the right of the root node, with a red line pointing to the root node. Another red line points from the 'DOMTree' oval to the opening tag of the 'students' element in the XML code on the left.

Root

Students id=5403019

DOM(Document Object Model)

```
<?xml version=" 1.0" encoding=" UTF-8" ?>
```

```
<students id=" 5403019" >
```

```
<名前>新海 光</名前>
```

```
<クラス>38</クラス>
```

```
</students>
```

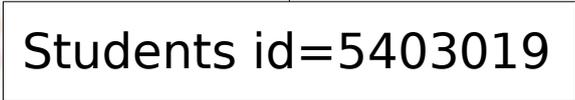


Root



A rectangular box labeled "Root" representing the root node of the tree.

Students id=5403019



A rectangular box labeled "Students id=5403019" representing the root's child node.

名前
新海



A rectangular box labeled "名前 新海" representing the child of the "Students" node.

DOM(Document Object Model)

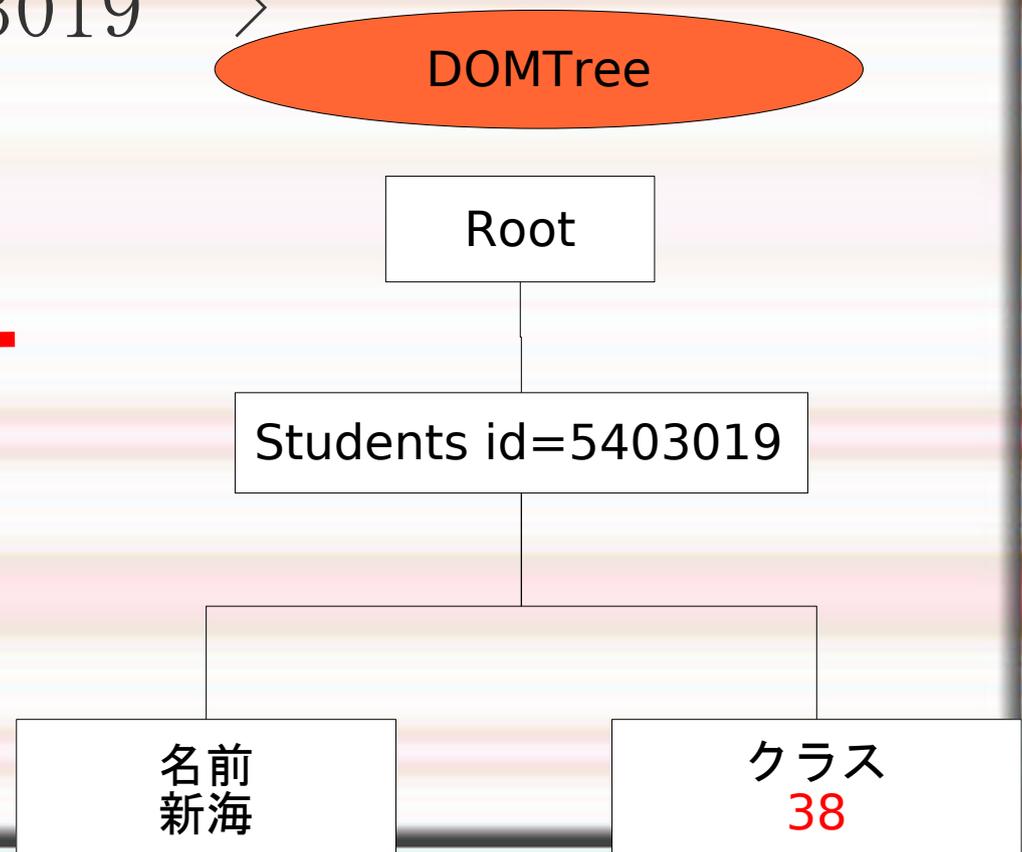
```
<?xml version=" 1.0" encoding=" UTF-8" ?>
```

```
<students id=" 5403019" >
```

```
<名前>新海 光</名前>
```

```
<クラス>38</クラス>
```

```
</students>
```



非同期通信

◆特徴

- ◆ Web ブラウザーはリクエストを送信したら返信を待たない
- ◆ サーバーは必要なデータのみを送信

クライアント

Webサーバー

アプリケーションサーバー

DBサーバー

XMLHttpRequest
送信

クライアント

Webサーバー

アプリケーションサーバー

DBサーバー

XMLHttpRequest
送信

リクエスト受信



クライアント

Webサーバー

アプリケーションサーバー

DBサーバー

XMLHttpRequest
送信

リクエスト受信

ビジネスロック
処理



クライアント

Webサーバー

アプリケーションサーバー

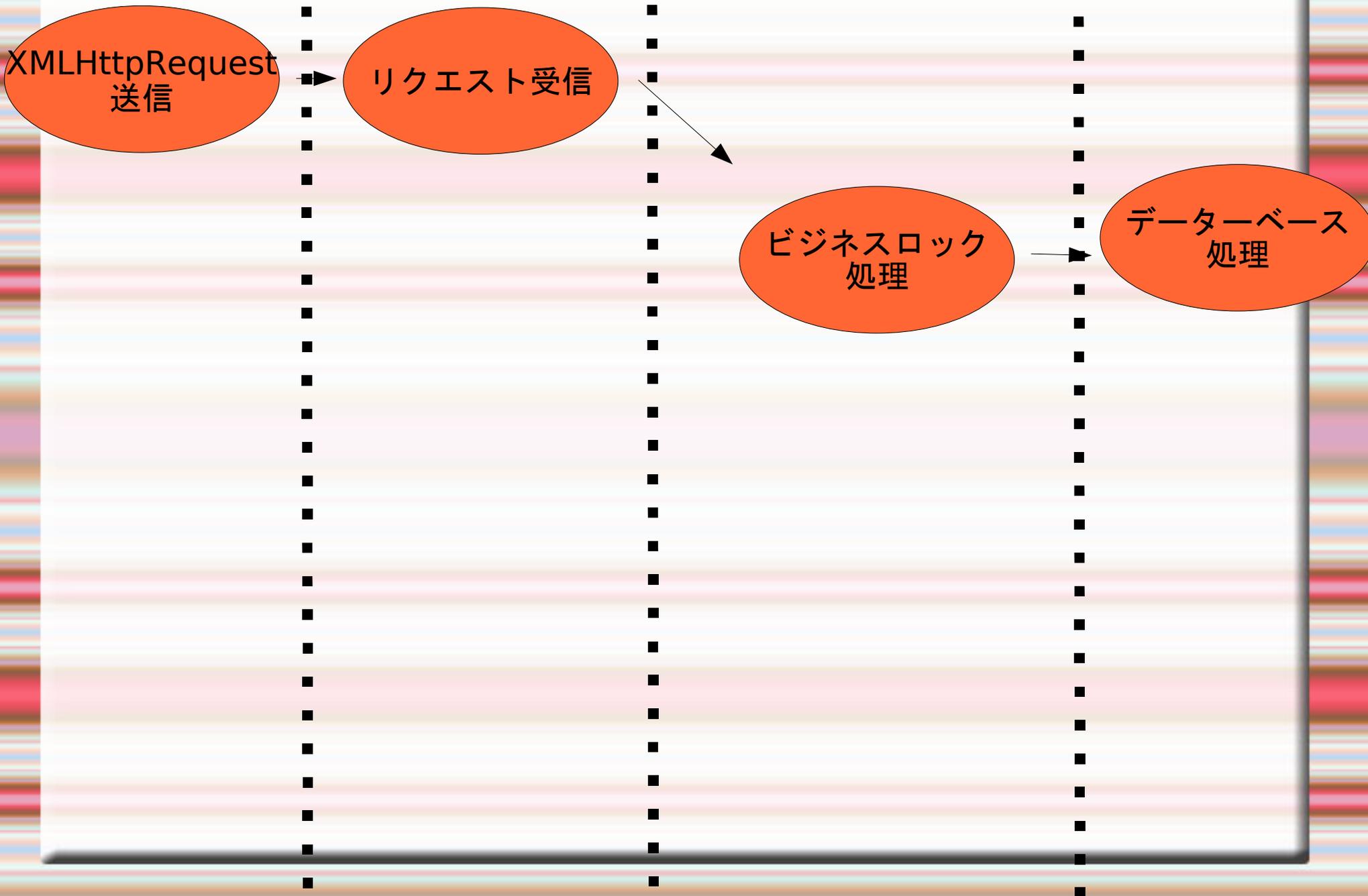
DBサーバー

XMLHttpRequest
送信

リクエスト受信

ビジネスロック
処理

データベース
処理



クライアント

Webサーバー

アプリケーションサーバー

DBサーバー

XMLHttpRequest
送信

リクエスト受信

ビジネスロック
処理

データベース
処理

データ返信

クライアント

Webサーバー

アプリケーションサーバー

DBサーバー

XMLHttpRequest
送信

リクエスト受信

ビジネスロック
処理

データベース
処理

ページの一部の
データ

データ返信

クライアント

Webサーバー

アプリケーションサーバー

DBサーバー

XMLHttpRequest
送信

リクエスト受信

ビジネスロック
処理

データベース
処理

ページの一部の
データ

データ返信

結果の返信

クライアント

Webサーバー

アプリケーションサーバー

DBサーバー

XMLHttpRequest
送信

リクエスト受信

ビジネスロック
処理

データベース
処理

ページの一部の
データ

データ返信

結果の受信

結果の返信

クライアント

Webサーバー

アプリケーションサーバー

DBサーバー

XMLHttpRequest
送信

リクエスト受信

ビジネスロック
処理

データベース
処理

ページの一部の
データ

データ返信

結果の受信

結果の返信

ページの一部
を更新

システム機能

- ◆ 管理ページ
 - ◆ 用語 追加、削除、修正
 - ◆ 日記 追加、削除、修正
 - ◆ 地域 追加、削除、修正
 - ◆ 場所 追加、削除、修正
 - ◆ ユーザー管理 追加、削除
- ◆ 公開ページ
 - ◆ 用語
 - ◆ 日記
 - ◆ Google Map

Pear

- ◆ PHPの拡張ライブラリー
- ◆ DB : データベースとの接続、追加、削除、修正をして
くれる。
- ◆ Auth : 認証
- ◆ HTML QuickForm
- ◆ フォーム
- ◆ HTML QuickForm Controller
- ◆ : フォームの遷移

◆ データベース名

◆ Kitamuradb

◆ テーブル名

◆ Auth

認証テーブル

◆ Youhotb

用語テーブル

◆ Nikkitb

日記テーブル

◆ placetb

場所テーブル

データベース設計

◆ Auth	認証テーブル	
◆ Userid	整数	ユーザーID
◆ Username	可変長文字	ユーザ
◆ Password	可変長文字	パスワード

データベース設計

- | | | |
|-----------|--------|------|
| ◆ Youhotb | 用語テーブル | |
| ◆ Yougoid | 整数 | 用語ID |
| ◆ Yougo | 可変長文字 | 用語 |
| ◆ Kaisetu | 可変長文字 | 解説 |

データベース設計

◆ Nikkitb	日記テーブル	
◆ nikkiid	整数	日記ID
◆ datename	date型	日付
◆ weather	可変長文字	天気
◆ Nikki	可変長文字	日記
◆ Hosoku	可変長文字	捕捉
◆		

説明

データベース設計

◆ placetb	場所テーブル	
◆ Plcid	整数	場所ID
◆ Locid	整数	地域ID
◆ Dateid	整数	日付ID
◆ Place	可変長文字	場所
◆ Description	可変長文字	説明
◆ Link	可変長文字	リンク
◆ Image	可変長文字	イメージファイル
◆ Lat	浮動小数点	緯度
◆ Lng	浮動少数点	経度