



# Ruby on Railsが注目される理由

---

株式会社ネットワーク応用通信研究所  
前田 修吾



# 自己紹介

---

- 前田 修吾 (まえだ しゅうご)
- (株)ネットワーク応用通信研究所  
基盤研究グループ 主任研究員
- プログラマ
- 愛知県出身
- 1975年11月24日生れ



# Ruby on Railsとは

---

- Webアプリケーションフレームワーク
- アプリケーション開発言語としてRubyを採用
  - Ruby on Rails自体もRubyで実装
- RailsやRoRと略されることが多い
- 作者はDavid Heinemeier Hansson
  - DHHと呼ばれることが多い

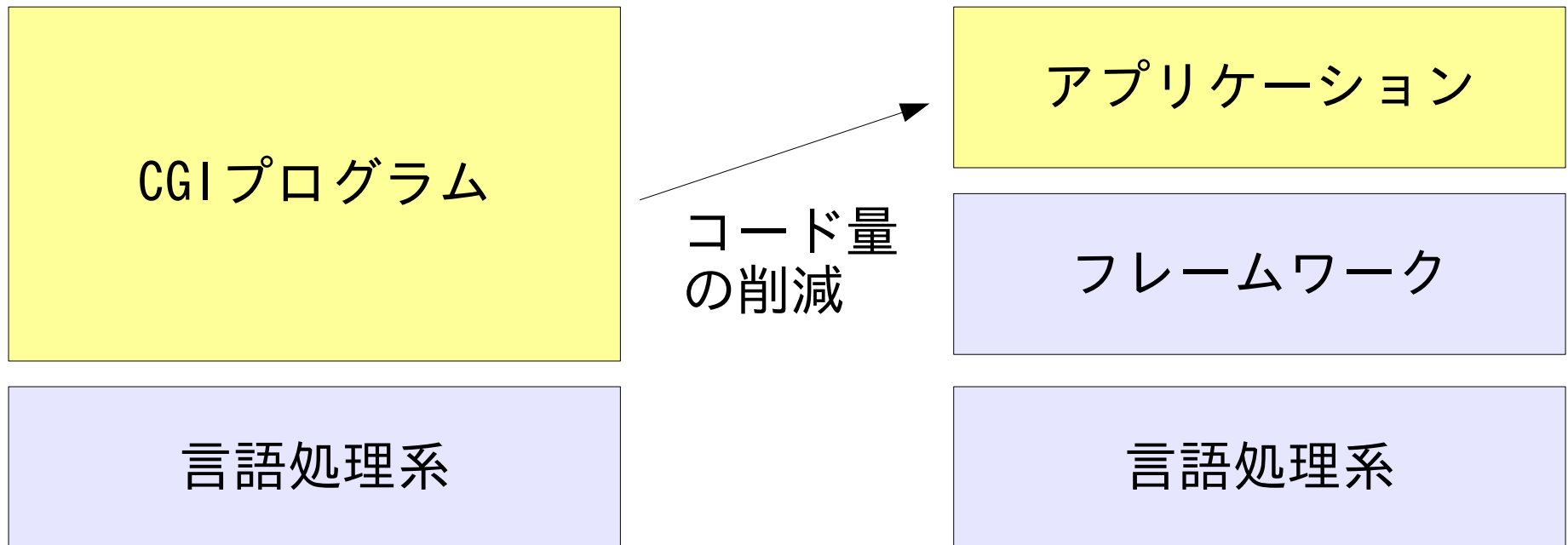


# Webアプリケーションフレームワーク

- Webアプリケーションの開発を簡単にするもの

古典的CGIプログラムの場合

フレームワークを使う場合





# Ruby on Railsは注目されている？

---

- Railsを知っている？
- Railsを使っている？
- まわりにRailsを知っている人がいる？
- まわりにRailsを使っている人がいる？



# Googleの検索結果



[ウェブ](#) [イメージ](#) [ニュース](#) [マップ](#) [グループ](#) [more »](#)

"Ruby on Rails"

Google 検索

ウェブ全体から検索  日本語のページを検索

ウェブ

"Ruby on Rails" の検索結果 約 21,500,000 件中 1 - 10 件目 (0.14 秒)



[ウェブ](#) [イメージ](#) [ニュース](#) [マップ](#) [グループ](#) [more »](#)

Struts

Google 検索

ウェブ全体から検索  日本語のページを検索

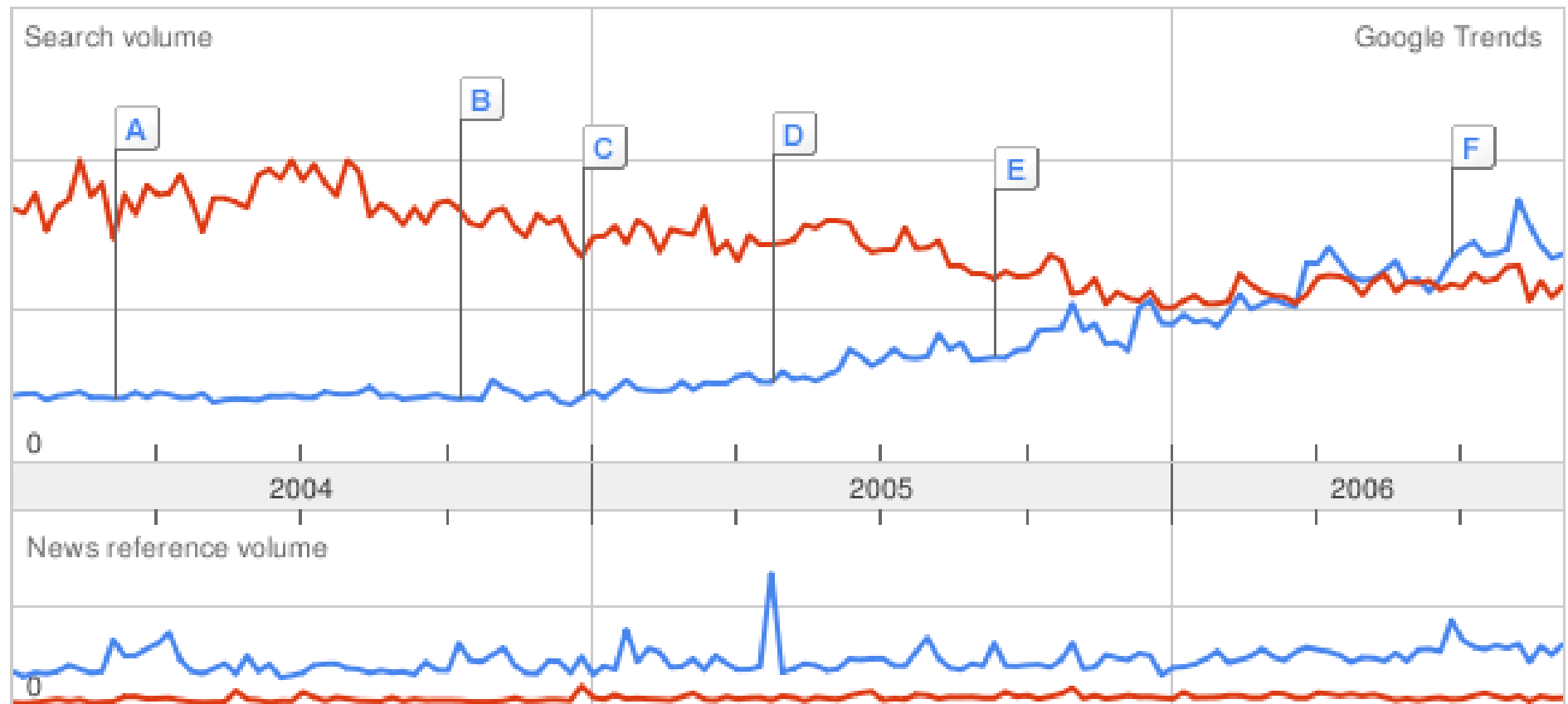
ウェブ

Struts の検索結果 約 17,700,000 件中 1 - 10 件目 (0.04 秒)



# Google Trends

● rails ● struts





# Google AdSense

Rails の検索結果 約 55,000,000 件中 1 - 10 件目 (0.04 秒)

を誇るWebア  
短時間で高機

[シ](#)

ails(以下、Rails)」  
ニアの目線から見

[連ページ](#)

アプリケーション  
築しています。

介します。...  
方ならともか

対応できる

スポンサー

## [Ruby on Rails トレーニン](#)

話題の Ruby on Rails の基本スキル  
を3日間で学ぶチャンスです!  
[www.school.ctc-g.co.jp/ruby/](http://www.school.ctc-g.co.jp/ruby/)

## [Railsエンジニア募集](#)

いま話題のRuby on Railsを使って  
イノベーションに加担しませんか?  
[zerobase.jp](http://zerobase.jp)

## [統合開発・運用環境](#)

Rails Platform - 開発環境、教育、  
サポートをパッケージ化、好評販売中  
[www.rakuten.co.jp/osj-online/](http://www.rakuten.co.jp/osj-online/)

## [Rubyエンジニア急募](#)

Railsによる自社サービス構築業務  
先端の開発環境がキミを待つ  
[tech.feedforce.jp/recruit.html](http://tech.feedforce.jp/recruit.html)

## [Ruby on Rails 講習会](#)

RubyのNaClだからできる豪華講師陣  
によるRailsトレーニングを開催中!  
[rails.netlab.jp](http://rails.netlab.jp)

## [Rails](#)

1500円以上国内配送無料。代引OK  
5000円以上購入は最大500円還元!  
[Amazon.co.jp](http://Amazon.co.jp)

## [Ruby on Rails入門](#)

Ruby on Railsを基礎から解説  
iPodnanoプレゼントキャンペーン中  
[www.thinkit.co.jp/](http://www.thinkit.co.jp/)

Struts の検索結果 約 17,700,000 件中 1 - 10 件目 (0.04 秒)

スポンサー

れ2と呼ばれる  
アプリケーション開

2 source distribution  
バイナリディストリ

:ml - 18k -

ーション・フ  
ークの中でも

## [変化に強いITシステム基盤](#)

フレームワークのサポートにより  
Javaアプリケーション開発を支援  
[www.hitachi.co.jp/cosminexus/](http://www.hitachi.co.jp/cosminexus/)

## [NRI OpenStandia](#)

高品質システム基盤の構築を実現  
オープンソース・サポートサービス  
[www.nri-aitd.com/openstandia](http://www.nri-aitd.com/openstandia)

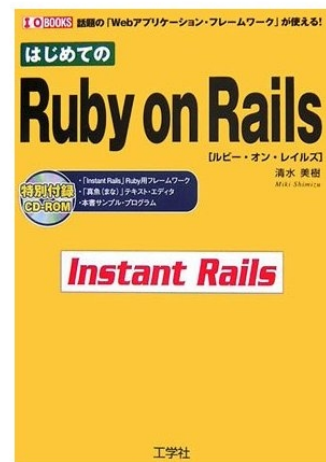
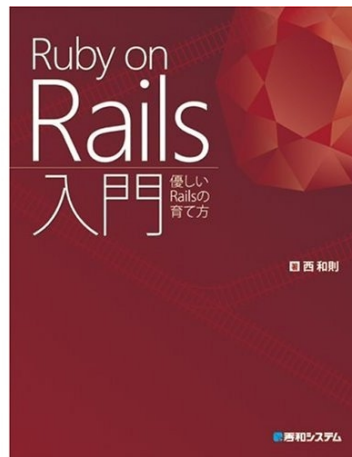
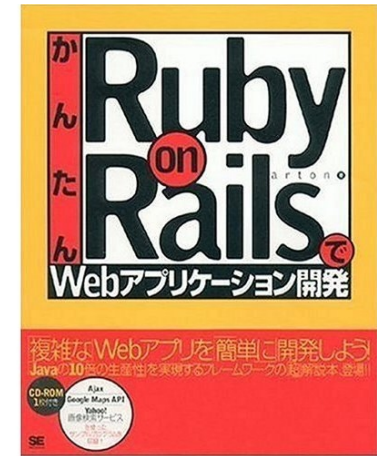
## [Development Tools](#)

Oracle JDeveloper: A comprehensive  
Java and Web services IDE.  
[www.Oracle.com/tools](http://www.Oracle.com/tools)





# 今年発売された書籍





# フォロワー

Trails

Biscuit

MonoRail

Sails

PHP on Trax

Grails

Catalyst

Django

Cake

Turbo Gear

■ Java ■ Perl ■ PHP ■ Python ■ .NET ■ Groovy



## その他

---

- Google/O'Reillyの2005 Open Source Awardで Railsの作者がBest Hacker of the yearに
- RailsConf 2006に400人を超える開発者が参加
  - ヨーロッパでも開催
- Sun MicrosystemsがJRubyの開発者を雇用



# RailsConf 2006(1)







# RailsConf 2006(2)

---





Ruby on Railsは注目されている？

---

注目されている



# Railsが注目される理由

---

- 巧みなマーケティング
- 高い生産性
- プログラマを惹きつける魅力



# 巧みなマーケティング

---

- わかりやすいキャッチフレーズ
  - Don't Repeat Yourself
  - Convention over Configuration
  - Javaの10倍の生産性
- 動画の活用
  - 15分でできるブログ
- 洗練されたデザイン
- DHHのプレゼンテクニック





# 洗練されたデザイン

www.rubyonrails.org

basecamp

Overview | [Screencasts](#) | [Download](#) | [Documentation](#) | [Weblog](#) | [Community](#) | [Source](#)



## Web development that doesn't hurt

Ruby on Rails is an open-source web framework that's optimized for programmer happiness and sustainable productivity. It lets you write beautiful code by favoring convention over configuration.

---

[Rails 1.2.0: Release Candidate 1](#), [New book: Agile Web Development with Rails \(2nd Ed\)](#), [Jobs: Find or pitch work](#)

### Get Excited



[Screencasts & presentations](#)

### Get Started



Released Aug 10, 2006

### Get Better



[API, Books, tutorials, samples](#)

### Get Involved



[Join the community](#)

"It is impossible not to notice Ruby on Rails. It has had a huge effect both in and outside the Ruby community... Rails has become a standard to which even well-established tools are comparing themselves to."

-Martin Fowler, Author of *Refactoring*, *PoEAA*, *XP Explained*

[Read more quotes](#)

Ruby Development Ruby Developers [My account](#) | [Log-out](#)

Overview | Messages | To-Do | Milestones | Writeboards | People | Search

### Project overview

**Late Milestones**

101 days late: [1.8.5 release](#) (Shyouhei URABE is responsible)

**What's fresh?**

**Today**  
No activity today

**Yesterday**  
No activity yesterday

**Before Yesterday**

- WRITEBOARD** **Stabilizing Sandbox**  
updated by Why t. 8 Sep
- MILESTONE** **Done : 1.8.5 preview2**  
Shyouhei U. was responsible
- MILESTONE** **Done : 1.8.5 preview1**  
Shyouhei U. was responsible
- MILESTONE** **Added : 1.8.5 release Due 16 Aug**  
Shyouhei U. is responsible
- TO-DO** **Added :** [ruby-core:07806] Ruby fails to parse valid wsdl to [Fix 1.8 bugs](#)
- MESSAGE** **Ruby releases and SEGV**  
by Why t. in [Release](#), 5 Apr
- TO-DO** **Added :** [ruby-talk:168731] "disallow sticky world writable directory in PATH": Why? to [Fix 1.8 bugs](#)
- TO-DO** **Done :** [ruby-dev:28053][ruby-core:6983] test/drb hangs on Solaris 64bit binary in [Porting 1.8](#)
- TO-DO** **Added :** [ruby-dev:27729] Thread deadlock: when sianale

**Subscribe to project's RSS feed**

Want to be notified any time someone posts a message, comment or file, or adds or completes a to-do item or milestone in this project? Just [subscribe to your project RSS feed](#) [RSS](#) ([What is RSS?](#))

**People on this project**

**matz**

Yukihiko Matsumoto  
Last login 3 days ago

**Ruby Developers**

Shugo Maeda

Akira Tanaka  
Last login 18 minutes ago

Why the lucky stiff  
Last login 19 minutes ago

akira yamada  
Last login 19 days ago

Minero Aoki  
Last login 27 days ago

Yui NARUSE  
Last login 29 days ago

Nathaniel Talbott  
Last login 60 days ago

Shyouhei URABE  
Last login 113 days ago

Ryan Davis  
Last login 129 days ago

Yuuzou GOTOU  
Last login 135 days ago



# DHHのプレゼンテクニック





# 高い生産性

---

- Javaの10倍?
  - 10倍は極端なケース
  - 生産性はかなり高い
    - NaClもCGIKitから乗り換え
- Rubyによるコンパクトな記述
- 強力な自動生成機能



# プログラマを惹きつける魅力

---

- Rubyを仕事に使える!
- 言語内DSL
  - XML不要
- 書いたコードをすぐ試せる
  - コンパイル不要
  - サーバの再起動不要
- プラグインによる拡張
  - フレームワークの機能を拡張できる



# Rails誕生の経緯

---

- DHHのバックグラウンド
  - PHP
    - 早いけど汚い
  - Java
    - 遅いけどきれい
- 「早くてきれい」な開発環境 → Ruby on Rails



# Railsの設計方針

---

- Don't Repeat Yourself (DRY)
  - 重複を避ける
- Convention over Configuration (CoC)
  - 設定より規約(決まり事)



# Don't Repeat Yourself (DRY)

---

- 同じことを何カ所にも記述すると…
  - 記述量が増える
  - メンテナンス性が下がる
- DRY = 重複の排除
  - by Dave Thomas / Andy Hunt
  - ソフトウェア開発の基本中の基本
  - でも守れていないことが多い
- Once And Only Once
  - XPで使われるフレーズ(意味は同じ)



# Convention over Configuration (CoC)

---

- 設定は面倒
  - とくにXMLは…
- 規約(決まり事)を守れば、設定は不要に
  - たとえば…
    - テーブル名は複数形で小文字・アンダーバー区切り
    - テーブルに対応するクラス名はCamelCase
- 柔軟性を犠牲に開発コストを低減
  - 設定で規約から外れた書き方もできる
    - ただし、記述量は増える





# Railsの特徴

---

- オールインワン
- 密結合
- 自動生成の多用
- 言語内DSL
- プラグインにより拡張可能



# オールインワン

---

- 必要なものがすべて揃う
  - O/Rマッパー
  - ビューテンプレート
  - Webサーバー
  - テスト支援ツール
- 他の選択肢がない
  - 柔軟性はない
  - 準備・習得にかかるコストが低い



# 密結合

- 
- MVCの各コンポーネントが密接に結合
  - 柔軟性を犠牲に開発コストを低減



# 自動生成の多用

---

- アプリケーションの雛型
- モデル
- コントローラ
- CRUD(scaffold)
- テスト
- テーブル定義(migration)



# 言語内DSL

---

- DSL = Domain Specific Language
  - 特定のドメイン(分野)に特化した言語
    - 設定ファイル
    - テンプレート
    - エディタのマクロ
- 言語内DSL
  - 汎用のプログラミング言語によりDSLを定義
  - 新しい文法を覚えなくてもよい
  - 記述力の高い言語のみで利用可能なテクニック
    - Lisp  $\leftrightarrow$  Java



# プラグインにより拡張可能

---

- フレームワーク自体をプラグインにより拡張可能
  - <http://www.agilewebdevelopment.com/plugins>
  - 433個のプラグインが登録
- プラグインもRubyで記述
- 出来のよいプラグインがRailsに取り込まれることも
  - Harvested Framework ⇔ Foundation Framework



# Railsの概要

---

- Railsのアーキテクチャ
- Railsのコンポーネント



# Railsのアーキテクチャ

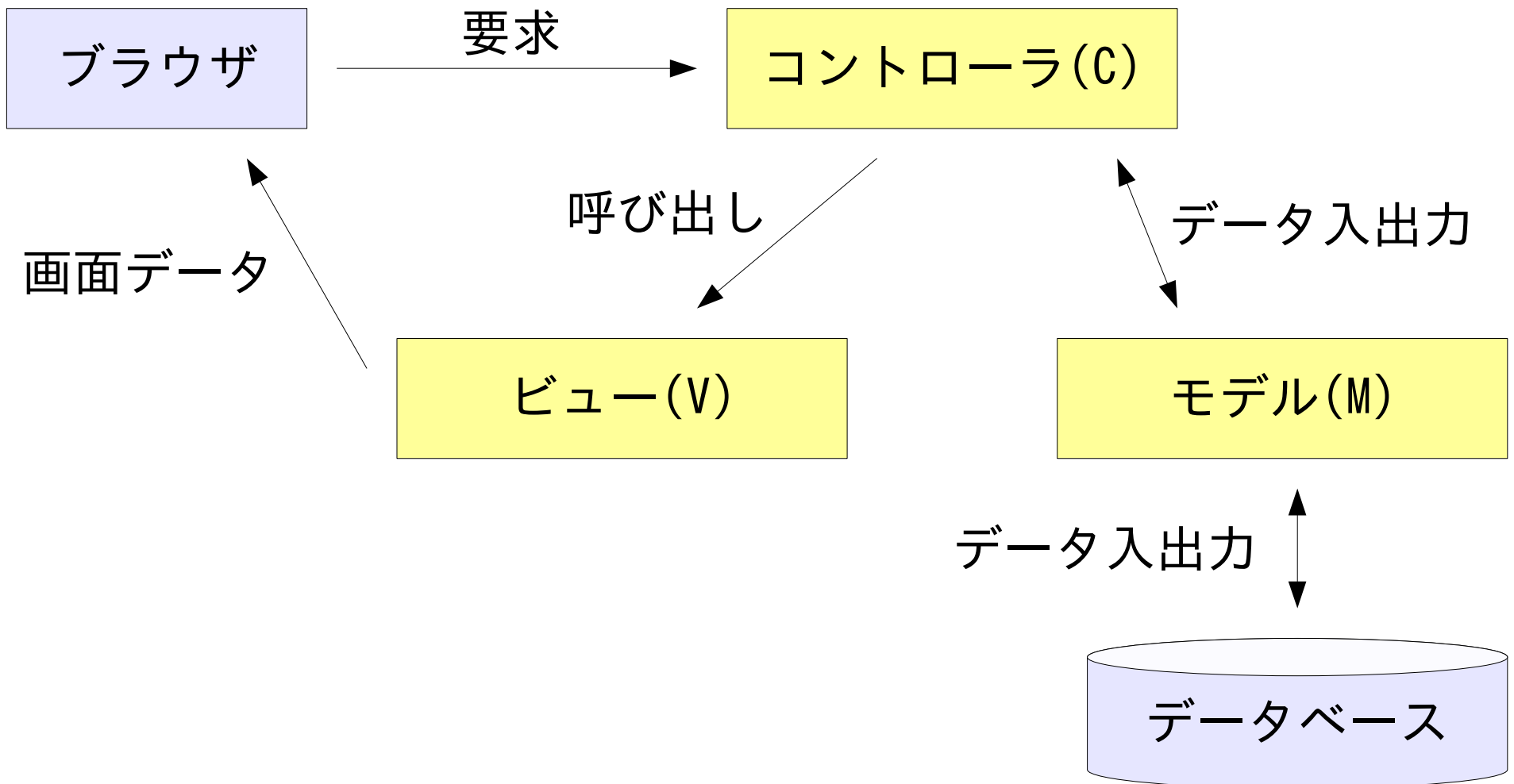
---

- MVCアーキテクチャ
  - 最近ではWebでもMVCが一般的
- アプリケーションを3つのパートに分けて記述
  - モデル(M)
    - データを扱う部分
  - ビュー(V)
    - 画面表示を行う部分
  - コントローラ(C)
    - 全体の制御を行う部分





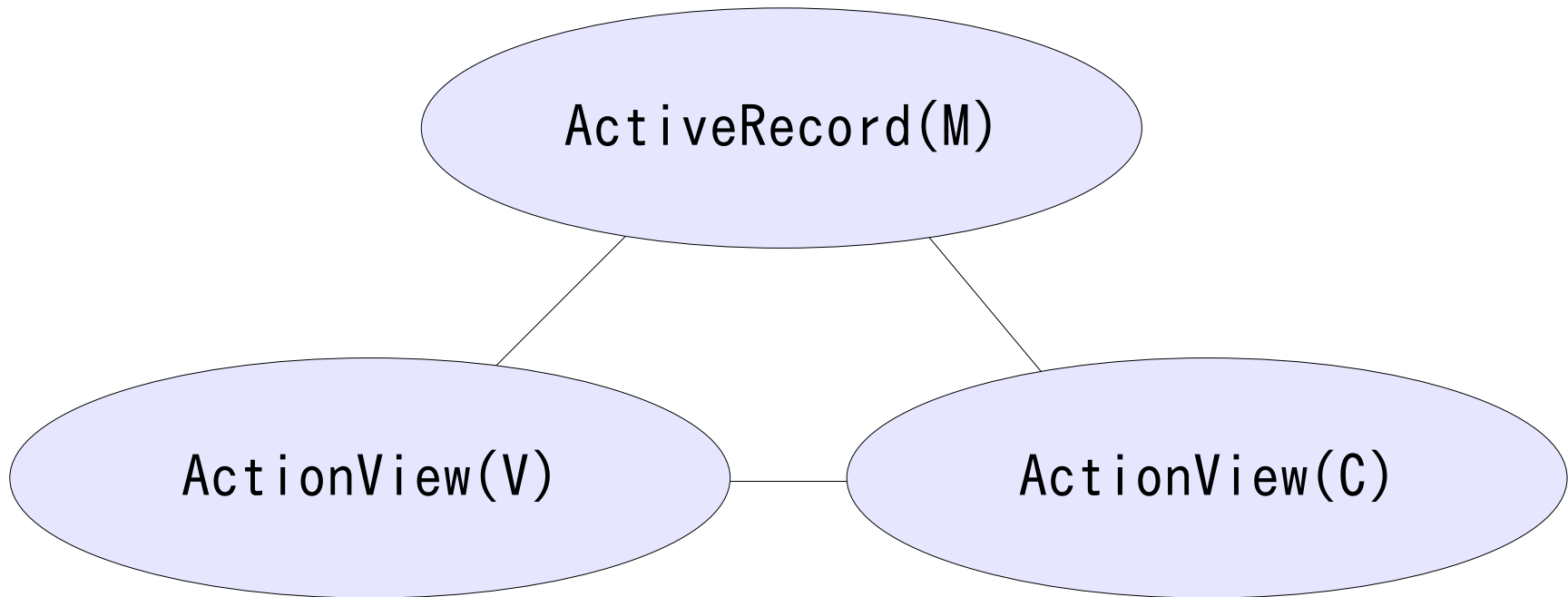
# MVCアーキテクチャ





# Railsのコンポーネント

---





# ActiveRecord

---

- モデル(M)を担当するコンポーネント
- O/Rマッパー
  - オブジェクトの世界とリレーショナルの世界を仲介
- クラス名とテーブル名の対応は規約ベース(CoC)
- データベーススキーマから列情報を自動取得
- リレーションシップは言語内DSLで定義
- スキーマ定義に一定の制約



# クラス名とテーブル名の対応

---

- クラス名はCamelCase
- テーブル名はすべて小文字・アンダーバー区切り
- 規約から外れる場合は、明示的に設定

クラス名	テーブル名
Product	products
Person	people
LineItem	line_items



# 列情報の自動取得

---

```
CREATE TABLE products (  
  id integer,  
  name varchar(255),  
  price integer  
);
```

```
class Product < ActiveRecord::Base  
end
```

```
product = Product.new  
product.name = "Rails Platform"  
product.price = 50000  
product.save
```



# リレーションシップの定義

---

```
CREATE TABLE categories (...);
CREATE TABLE products (
  ...
  category_id integer
);
CREATE TABLE coupons (
  ...
  product_id integer
);
```

```
class Product < ActiveRecord::Base
  belongs_to :category
  has_many :coupons
end
```



# スキーマ定義の制約

---

- integer型のプライマリキーが必要
  - デフォルトではidという名前
  - プライマリキーを複合キーにはできない
- テーブル名は小文字・アンダーバー区切りで複数形
- 外部キーのサポートがない
  - ActiveRecordによるリレーションシップの定義を推奨
  - 外部キーを使えないわけではない



# ActionController

---

- コントローラ(C)を担当するコンポーネント
- 関連するアクション(ブラウザからの要求に対する処理)を一つのコントローラにまとめる
- アクションはコントローラのメソッドとして記述
- 対応するビューは規約で決まる
  - listアクションならビューはlist.rhtml
  - 規約から外れる場合は明示的に表示するビューを指定





# コントローラの例

---

```
class StoreController < ApplicationController
  def list
    @products = Product.find(:all)
    # 明示的な指定がなければ自動的にlist.rhtmlが
    # 呼び出される
  end
end
```



# ActionView

---

- ビュー(V)を担当するコンポーネント
- デフォルトではeRuby(HTMLにRubyを埋め込む書式)で記述
- コントローラのインスタンス変数を参照(密結合)
- MVCの3コンポーネント中もっとも汚い部分
  - 理想よりも現実



# ビューの記述例

---

```
<ul>  
  <% for product in @products %>  
    <li><%=h product.name %></li>  
  <% end %>  
</li>
```



# 他の言語でRailsは作れる?

---

- YesでもありNoでもある
- 機能の有無○×表で示せば、他の言語でも同じになるかもしれない
  - 実際にRailsを参考にした他言語のフレームワークは多数ある
- でもそれは今のRailsとは違うもの
  - 定量的に計れない魅力
- もちろん、Railsとは違ったやり方でRailsにはない魅力を持ったフレームワークもあり得る



# まとめ: Railsが注目される理由

---

- 巧みなマーケティング
- 高い生産性
  - 柔軟性を犠牲に実現
  - 思いきった割り切り
    - Railsの機能の20%で問題の80%を解決する
  - 用途によっては向いていない
    - レガシーなスキーマを扱う必要がある場合など
- プログラマを惹きつける魅力
  - Rubyによるところが大きい



# Railsの問題点

---

- 導入に手間がかかる
- 日本語の対応が弱い
- サポートがない



# 導入に手間がかかる

---

- 必要なソフトウェアがいろいろある
  - Ruby
  - MySQL
  - RubyGems
  - その他各種ライブラリなど
- しかもインストール方法がソフトウェアによって違う



# 日本語の対応が弱い

---

- 日本語を扱うためにいくつかの設定が必要
- scaffoldは英語のビューを生成
- validationのメッセージは英語
- IDE(RadRails)のインタフェイスも英語





# サポートがない

- 
- Rails自体は商用ではないため、開発元のサポートを期待できない
  - バグ報告なども英語で行う必要がある



# 解決策

- Rails Platform





# Rails Platform

---

- Windows用のRails開発環境パッケージ
- 開発元
  - 株式会社ネットワーク応用通信研究所(NaCl)
- 販売元
  - オープンソース・ジャパン株式会社(OSJ)



# 製品ラインナップ

---

## ● Rails Platform Basic

- 開発環境パッケージ
- アップデートサービス(1年間)
- OSJによるメールサポート(問合せ回数無制限/1年間)
- 価格 5万円

## ● Rails Platform Premium

- Rails Platform Basic相当の内容
- + NaClによる3日間のトレーニングプログラム
- 価格 20万円



# Rails Platformの特徴

---

- 使いやすいインストーラ
- 日本語対応
- 充実したサポート



# 使いやすいインストーラ

---

- 一つのインストーラで必要なソフトウェアをインストール
- Windowsで一般的なウィザード形式を採用
- 「プログラムの追加と削除」でアンインストール可能



# 日本語対応

---

- 日本語を扱うための設定を自動化
- scaffoldの日本語化
- validationの日本語化
- もちろんインストーラの表示も日本語



# 充実したサポート

---

- OSJによる問合せ回数無制限のメールサポート
- NaClによるトレーニングプログラム





# デモ

---

- 実際の開発環境をご覧ください



# 質疑応答

---



# Ruby on Railsが注目される理由

---

株式会社ネットワーク応用通信研究所  
前田 修吾