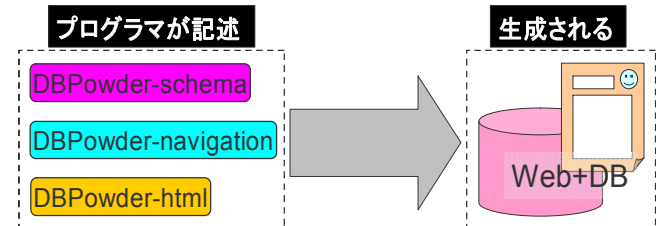
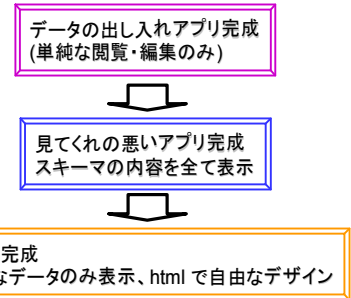


DBPowder: RDBMS活用フレームワーク

- 複雑なスキーマがすぐに動く
 - DBPowder-schema の記述のみで、まずは動く
- スキーマ定義を DBPowder-schema に集約
 - データを扱いやすい
 - スキーマの変更に強い
- 定型的で煩雑な記述を減らせる
 - DBスキーマ作成・DBへ値格納・DBから値取り出し・ブラウザへ値表示・ブラウザから入力受付・DB/ブラウザ値チェック・セキュリティ対策・複数ユーザ対応
- 柔軟な記述を損なわない
 - ページは html で書ける

DBPowder による開発 ~DBPowderの3ステップ

1. DBPowder-schema
 - データ構造(スキーマ)を書く
2. DBPowder-navigation
 - ウェブページの動きを書く
3. DBPowder-html
 - ウェブページを書く



2006年度下期 未踏ソフトウェア (並木PM)
開発者: 村上 直 (高エネルギー加速器研究機構)
info@dbpowder.org
http://www.dbpowder.org/ (公開準備中)

<http://www.ipa.go.jp/jinzai/esp/2006mito2/gaiyou/5-18.html>

DBPowder-schema

- データ構造を階層構造で書く
 - テーブル名は []
 - 親との関係を 1:1, 1:n, n:1, n:n で
 - テーブルの項目を書き下す
- スキーマ構築に必要なキーを補ってくれる
 - 主キー、外部キー
 - 付加機能に活かされる属性
 - 詳細なスキーマ理論を知らずに複雑なスキーマを構築できる
 - 特に複数テーブル連携

DBPowder-navigation

- ページ全体
 - `name`: ページ名
 - `base`: ページの機能
- リンク・送信ボタン
 - `submit`: 送信ボタンの動作
 - `link`: リンクの動作
- 項目内の定義
 - `action`: 動作内容
 - `pageto`: 動作後のページ
 - `name`: 項目の名前
 - `pageto`, `html` で引用
- Yaml 形式で記述

```
common: {schema: blog}
pages:
- # 記事追加
  name: Add
  base: InputAdd
  submit: {name: next, pageto: ConfirmAdd}
- # 追加確認
  name: ConfirmAdd
  base: Item
  submit: {name: back, pageto: Add}
  submit: {name: submit, action: Insert, pageto: Item}
- # 記事修正
  name: Edit
  base: Item
  submit: {name: back, pageto: Add}
- # 修正確認
  name: ConfirmEdit
  base: Item
  submit: {name: back, pageto: Edit}
- # 一覧
  name: List
  base: Item
  submit: {name: back, pageto: Add}
```

プログラマが記述

DBPowder-schema

```
[user addon="user"]
[blog 1:n]
  blog_title text64
  blog_body text8192
```

DBPowder-navigation

```
common: {schema: blog}
pages:
- # 記事追加
  name: Add
  base: InputAdd
  submit: {name: next, pageto: ConfirmAdd}
```

DBPowder-html

```
<html> <body>
<error />
<form> 題名:
<input type="text" name="blog title">
<br> 本文:<br>
<textarea name="blog body">
</textarea>
<input type="submit" name="next"
value="確認">
</form> </body> </html>
```

Add.html

DBスキーマ作成
DBへ値格納
DBから値取り出し
ブラウザへ値表示
ブラウザから入力受付
DB・ブラウザ値チェック
セキュリティ対策
複数ユーザ対応

DBPowderが
やってくれる

生成されるもの



DBスキーマ

schema, navigation で
定義したものが html で
引用され、ウェブページ
になる