

Seasar Conference 2007 Autumn



うるま
Urūma による

簡単！ リッチアプリケーション開発

2007.11.11

エスエムジー株式会社

小森 裕介 (komori@smg.co.jp)



はじめまして！

- **名前:** 小森 裕介
- **Blog:** <http://d.hatena.ne.jp/y-komori/> (こもりん日記)
- **所属:** エスエムジー株式会社 (<http://www.smg.co.jp>)
- **主な仕事:**
 - Javaによる集中監視制御システム設計・開発
 - Webアプリケーションシステムの設計・開発
 - 教育・各種執筆活動
 - 日経ソフトウェア「とことん作って覚える! Java入門」
 - 「なぜ、あなたはJavaでオブジェクト指向開発ができないのか」
- **Seasar2とのかかわり**
 - S2Containerコミッタ、Urumaコミッタ、S2JMSコミッタ





プロダクト名を変えました

- 「S2JFace」から「^{うるま}Uruma」へ
 - SWT/JFace のみのサポートから、
Eclipse RCPアプリケーションへサポートを拡大
 - より親しみやすく、覚えやすい名前へ
 - うるま: 琉球語で「**珊瑚の島**」の意
 - 2007年11月2日 **version 0.1.0** リリース



Uruma のコンセプト(1)

「つらいGUI」

から

「楽しいGUI」

へ!



Uruma のコンセプト(2)

- Webアプリでは実現しにくい、
軽快なリッチGUIアプリケーションを
簡単に開発できるフレームワーク
 - Javaの世界では、このようなフレームワークは
少ない/存在しても高価で手が出しにくい

なんでもかんでも Webアプリで作るのは間違いでしょ！
世の中には適材適所ってものがあるんだ！





なぜ、Webアプリケーションが
圧倒的多数を占めるのか？

それは…

OSSのすぐれた
GUIフレームワークが
存在しないから



Uruma のコンセプト(4)

ならば
創ってしまえ!!!



Uruma の技術コンセプト

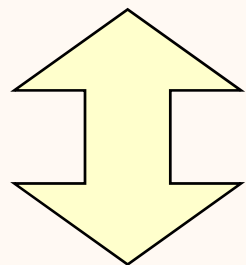
- XML によるGUI画面の作成
- 強力なデータバインディング
- Eclipseプラグイン/Seasar の知識がなくてもアプリを開発できる
- 必要なのはUrumaの規約とJavaAPIの知識だけ



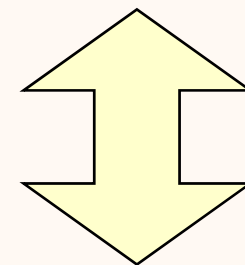
フレームワークのジレンマ

○ 簡単に使える

× 痒いところに手が届かない



Uruma



× 習得が難しい

○ 豊富な機能



「わかりやすさ・使いやすさ」の追求

- 自明の設定は書かなくて済むようにする
- 必要な部分のみを記述するようにする
- メソッドやフィールド名だけの暗黙規約は避けて、
アノテーションによる明示的な指定を採用

最小限の機能から、
実際に動かしながら
必要な機能を増やしていく、
アジャイルな開発が可能に



多孔質 (?) フレームワーク

Urumaは見かけ上、
下位レイヤを隠蔽している

Eclipse
RCP

UserApp

Uruma

Workbench

JFace

SWT

OSGi Runtime

必要に応じて
下位レイヤを
使うこともできる

下位レイヤを完全に隠蔽しないことで、
GUIの細かい制御が可能になる



ライブで作る！Urumaアプリケーション

「開く」メニューで
選択したファイルを実行

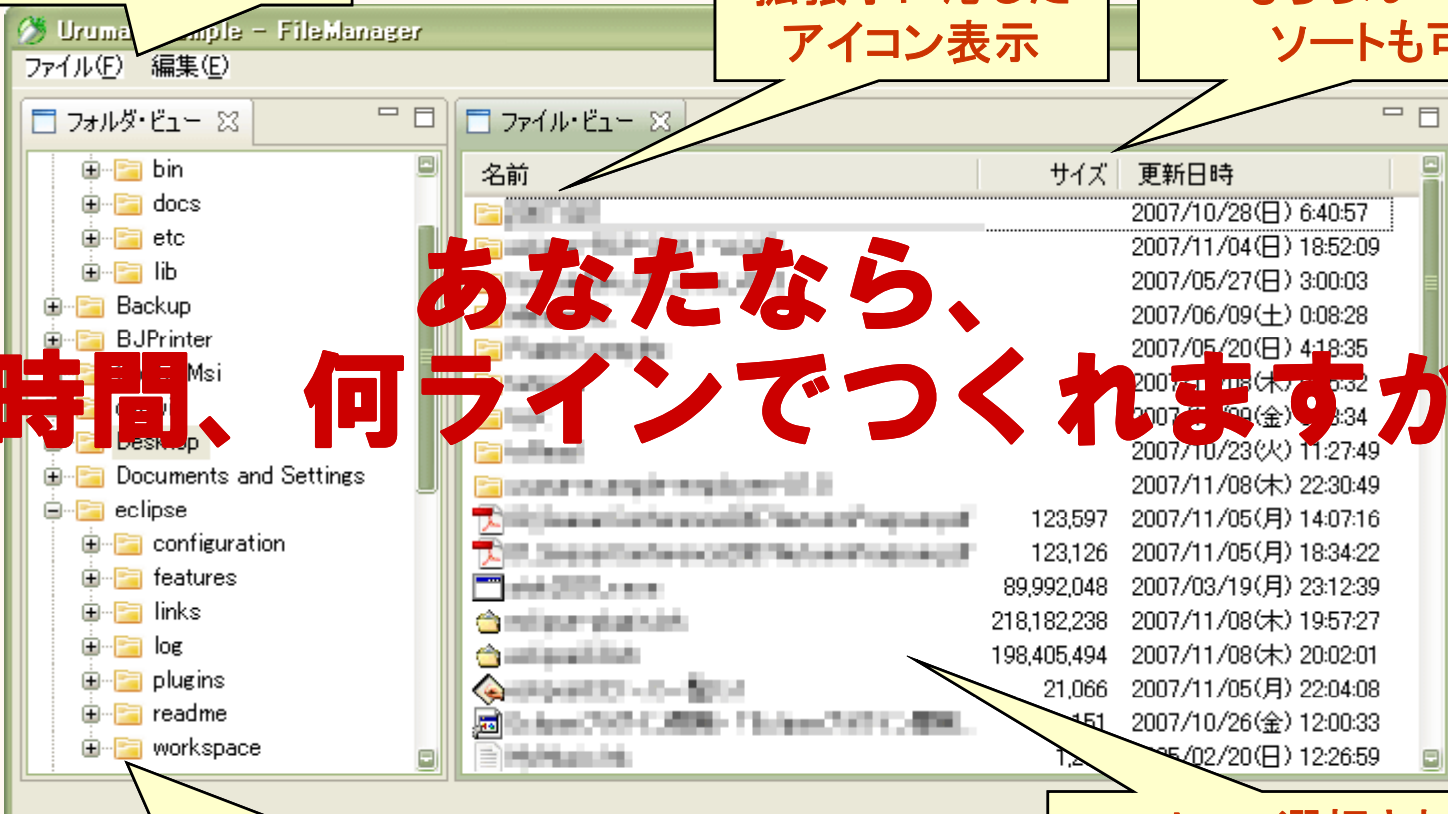
拡張子に応じた
アイコン表示

もちろん一覧の
ソートも可能

あなたなら、
何時間、何ラインでつくれますか？

Cドライブをツリー表示

ツリーで選択された
フォルダの中を表示

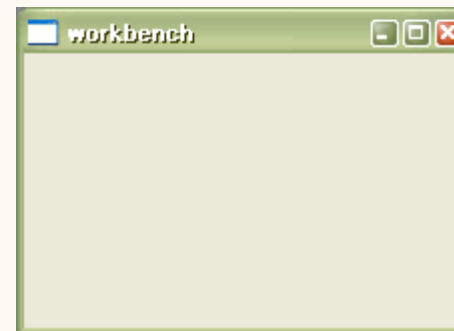




ライブでつくる! Urumaアプリケーション(1) workbench.xml でウィンドウを開く

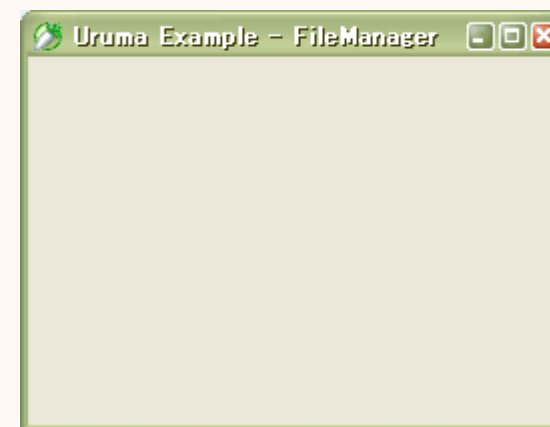
■ workbench.xml

```
<?xml version="1.0" encoding="UTF-8"?>  
<template xmlns="http://uruma.sandbox.seasar.org">  
  <workbench>  
  </workbench>  
</template>
```



■ workbench.xml

```
<?xml version="1.0" encoding="UTF-8"?>  
<template xmlns="http://uruma.sandbox.seasar.org">  
  <workbench title="Uruma Example - FileManager"  
    initWidth="50%" initHeight="50%"  
    image="/images/container.gif" statusLine="true">  
  </workbench>  
</template>
```

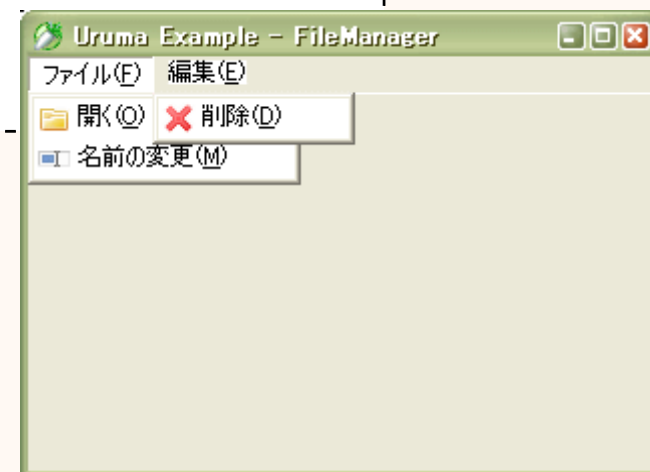




ライブでつくる! Urumaアプリケーション (2) メニューバーを追加する

■ workbench.xml

```
<workbench title="Uruma Example - FileManager"  
  initWidth="20%" initHeight="20%"  
  image="/images/container.gif" statusLine="true">  
  <menu>  
    <menu text="ファイル(&F)">  
      <menuItem text="開く (&O)" image="folder" />  
      <menuItem text="名前の変更 (&M)" image="rename" />  
    </menu>  
    <menu text="編集 (&E)">  
      <menuItem text="削除 (&D)" image="cross" />  
    </menu>  
  </menu>  
</workbench>
```





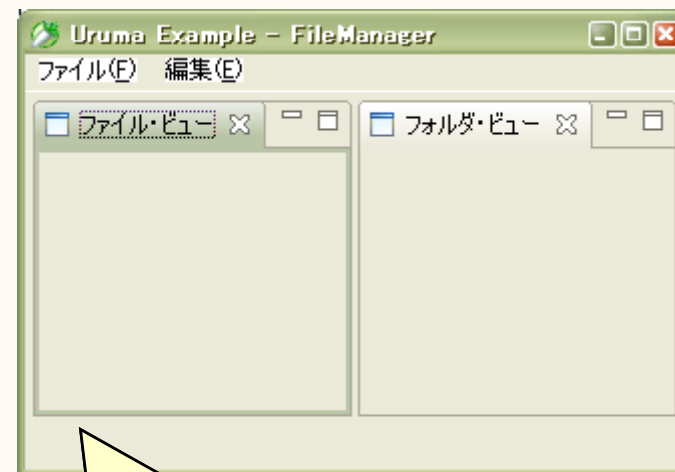
ライブでつくる! Urumaアプリケーション (3) ビューパートをつくる

■ /views/folderView.xml

```
<?xml version="1.0" encoding="UTF-8"?>  
<template xmlns="http://uruma.sandbox.seasar.org">  
  <viewPart title="フォルダ・ビュー">  
  </viewPart>  
</template>
```

■ /views/fileView.xml

```
<?xml version="1.0" encoding="UTF-8"?>  
<template xmlns="http://uruma.sandbox.seasar.org">  
  <viewPart title="ファイル・ビュー">  
  </viewPart>  
</template>
```



/views 配下にXMLファイルを
置くだけでビューパートができる!



ライブでつくる! Urumaアプリケーション(4) パースペクティブを設定する

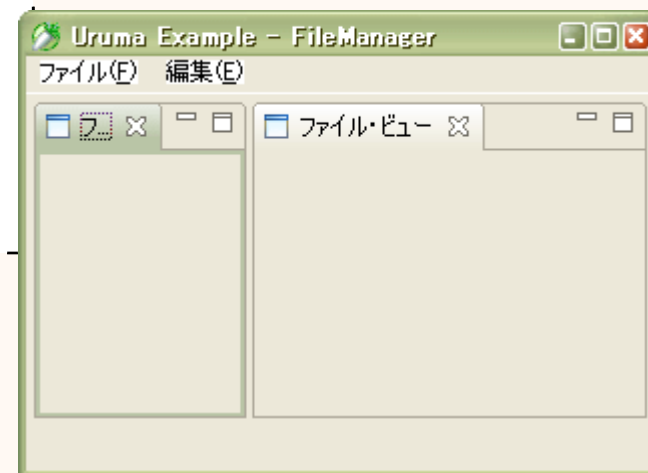
workbench.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<template xmlns="http://uruma.sandbox.seasar.org">
  <workbench title="Uruma Example - FileManager" initWidth="30%"
    initHeight="20%" image="/images/container.gif" statusLine="true">
    <menu>
      <menu text="ファイル (&F)">
        <menuItem text="開く (&O)" image="folder" />
        <menuItem text="名前の変更 (&M)" image="rename" />
      </menu>
      <menu text="編集 (&E)">
        <menuItem id="fileDelete" text="削除 (&D)" image="cross" />
      </menu>
    </menu>
    <perspective>
      <part ref="folderView" ratio="30" />
      <part ref="fileView" ratio="70" />
    </perspective>
  </workbench>
</template>
```



Id未指定の時は
ファイル名がidになる

パートはidで参照

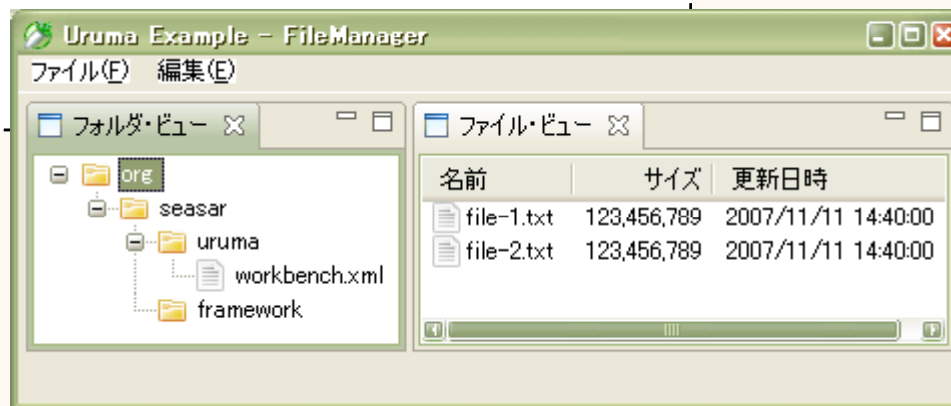




ライブでつくる! Urumaアプリケーション(5) フォルダ・ビューを作る

■ /views/folderView.xml

```
<viewPart title="フォルダ・ビュー">
  <tree>
    <treeItem text="org" image="folder">
      <treeItem text="seasar" image="folder">
        <treeItem text="uruma" image="folder">
          <treeItem text="workbench.xml" image="file" />
        </treeItem>
      </treeItem>
    </treeItem>
  </treeItem>
</tree>
</viewPart>
```





ライブでつくる! Urumaアプリケーション(6) ファイル・ビューを作る

■ /views/folderView.xml

```
<viewPart title="ファイル・ビュー">
  <table linesVisible="false">
    <tableColumn width="70" text="名前" />
    <tableColumn width="70" text="サイズ" alignment="RIGHT" />
    <tableColumn width="80" text="更新日時" />
    <tableItem>
      <tableCell text="file-1.txt" image="file" />
      <tableCell text="123,456,789" />
      <tableCell text="2007/11/11 14:40:00" />
    </tableItem>
    <tableItem>
      <tableCell text="file-2.txt" image="file" />
      <tableCell text="123,456,789" />
      <tableCell text="2007/11/11 14:40:00" />
    </tableItem>
  </table>
</viewPart>
```





ライブでつくる! Urumaアプリケーション (7) フォルダツリーを表示する

■ FolderTreeContentProvider.java (抜粋)

```
public class FolderTreeContentProvider
    implements ITreeContentProvider {
    public Object[] getChildren(final Object parentElement) {
        File parentFolder = (File) parentElement;
        File[] children = parentFolder.listFiles();
        List<File> temp = new ArrayList<File>();
        for (File child : children) {
            if (child.isDirectory()) {
                temp.add(child);
            }
        }
        return temp.toArray(new File[temp.size()]);
    }
    public boolean hasChildren(final Object element) {
        File file = (File) element;
        return file.isDirectory();
    }
}
```

■ /views/folderView.xml (抜粋)

```
<tree id="folderTree">
```

Id の一致する
コンテンツプロバイダ
ラベルプロバイダが
自動設定される

■ FolderTreeLabelProvider.java (抜粋)

```
public class FolderTreeLabelProvider
    extends LabelProvider {
    public Image getImage(final Object element) {
        Image folderImage = ImageManager.loadImage("folder");
        return folderImage;
    }
    public String getText(final Object element) {
        File folder = (File) element;
        return folder.getName();
    }
}
```



ライブでつくる! Urumaアプリケーション (8) フォルダビューのアクションクラスを作る

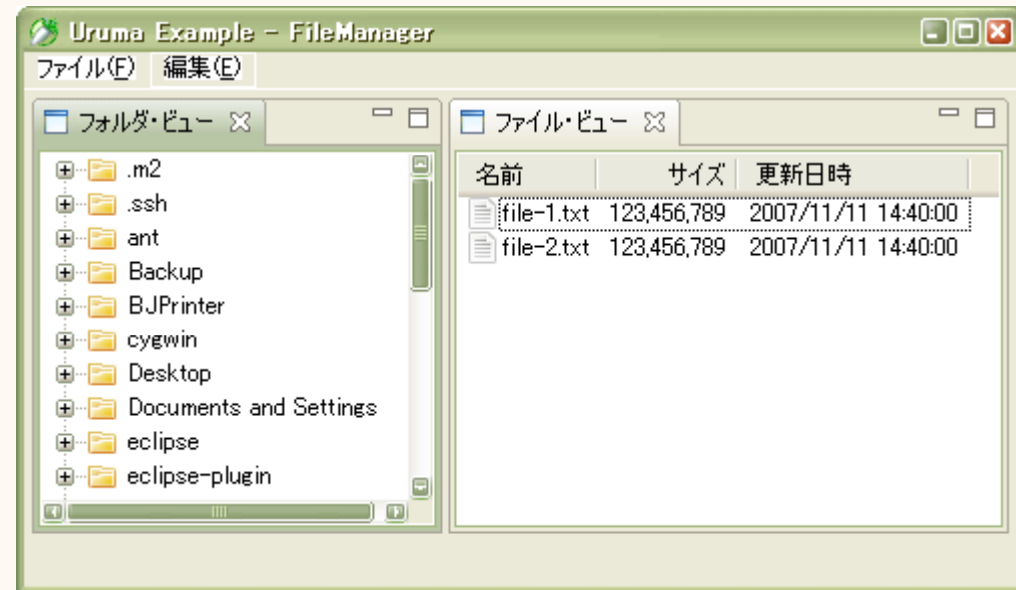
■ FolderViewAction.java (これだけ!)

```
@Form(FolderViewAction.class)
public class FolderViewAction {
    @ExportValue(id = "folderTree")
    public File root = new File("c:/");
}
```



@ExportValue

アノテートされたフィールドが、
画面表示時にバインドされる





ライブでつくる! Urumaアプリケーション (9) ファイルビューにフォルダの内容を表示する

■ FileViewAction.java

```
@Form(FileViewAction.class)
public class FileViewAction {
    @ExportValue(id = "fileDetailTable")
    public List<FileDto> fileList = new ArrayList<FileDto>();
    @SelectionListener(partId = "folderView")
    public void selectionChanged(final File parentFolder) {
        fileList.clear();
        File[] children = parentFolder.listFiles();
        (...途中省略...)
        fileList.add(fileDto);
    }
}
```

@SelectionListener
アノテートされたメソッドが、
ツリー選択時に呼び出される

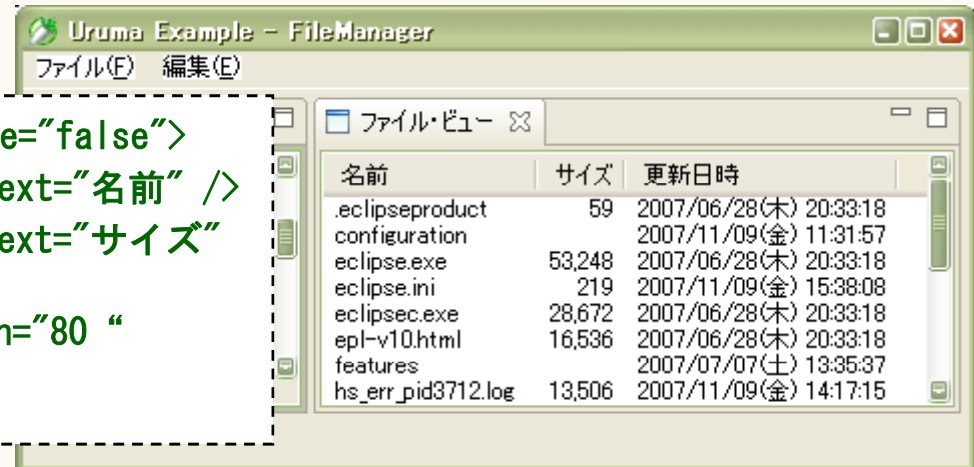
■ FileDto.java

```
public class FileDto {
    public String absolutePath;
    @BindingLabel
    public String fileName;
    @BindingLabel
    public String fileSize;
    @BindingLabel
    public String fileUpdateTime;
}
```

@BindingLabel
アノテートされたフィールドが、
同じ名前のカラムに表示される

■ /view/fileView.xml

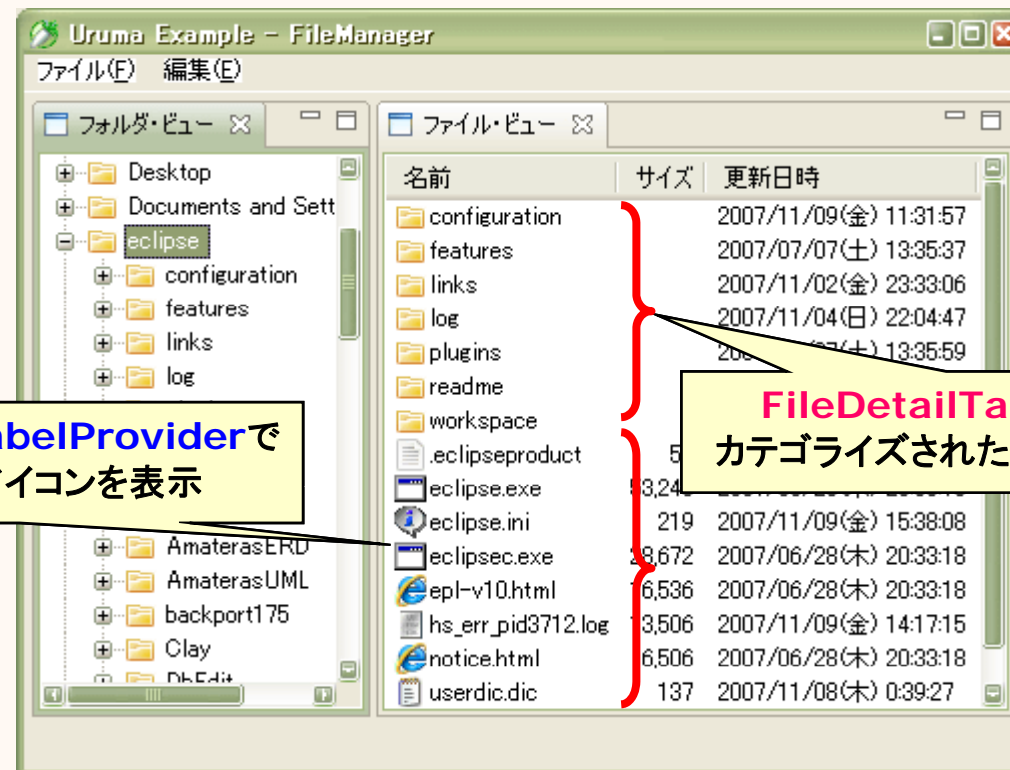
```
<table id="fileDetailTable" linesVisible="false">
    <tableColumn id="fileName" width="70" text="名前" />
    <tableColumn id="fileSize" width="70" text="サイズ"
        alignment="RIGHT" />
    <tableColumn id="fileUpdateTime" width="80"
        text="更新日時" />
</table>
```





ライブでつくる! Urumaアプリケーション(10) アイコン表示とソート機能を強化する

- ビューアの内容表示は決まった名前のクラスを用意するだけ!
 - (画面コンポーネントのid)ContentProvider
 - (画面コンポーネントのid)LabelProvider
 - (画面コンポーネントのid)Sorter



FileDetailTableLabelProviderで
拡張子に応じたアイコンを表示

FileDetailTableSorterで
カテゴリ化されたソート機能を追加



ライブでつくる! Urumaアプリケーション (11) メニューからアクションを実行する

■ workbench.xml

```
<menu text="ファイル(&F)">  
  <menuItem id="fileOpen" text="開く(&O)" image="folder" />  
  <menuItem text="名前の変更(&M)" image="rename" />  
</menu>
```

■ fileViewAction.java

```
@ImportSelection(id = "fileDataTable")  
public List<FileDto> selectedFile;  
  
@EventListener(id = "fileOpen")  
public void onFileOpenMenu() {  
  if (selectedFile.size() == 1) {  
    FileDto dto = selectedFile.get(0);  
    File file = new File(dto.absolutePath);  
    if (file.isFile()) {  
      Program program = Program.findProgram(StringUtils  
        .substringToLast(dto.fileName, "."));  
      if (program != null) {  
        Program.launch(dto.absolutePath);  
      }  
    }  
  }  
}
```

テーブルで選択中の
オブジェクトが代入される

対応する id のメニューが
選択されると呼び出される

名前	サイズ	更新日時
フォルダ .settings		2007/07/07(土) 13:39:27
フォルダ org.eclipse.core.runtime		2007/11/09(金) 11:35:05
フォルダ org.eclipse.equinox.app		2007/07/07(土) 13:39:25
フォルダ org.eclipse.help.base		2007/11/08(木) 23:12:13
フォルダ org.eclipse.osgi		2007/11/09(金) 11:32:21
フォルダ org.eclipse.ui.intro.universal		2007/07/07(土) 13:39:31
フォルダ org.eclipse.update		2007/11/10(土) 7:14:30
ファイル config.ini	3,287	2007/06/28(木) 20:33:18



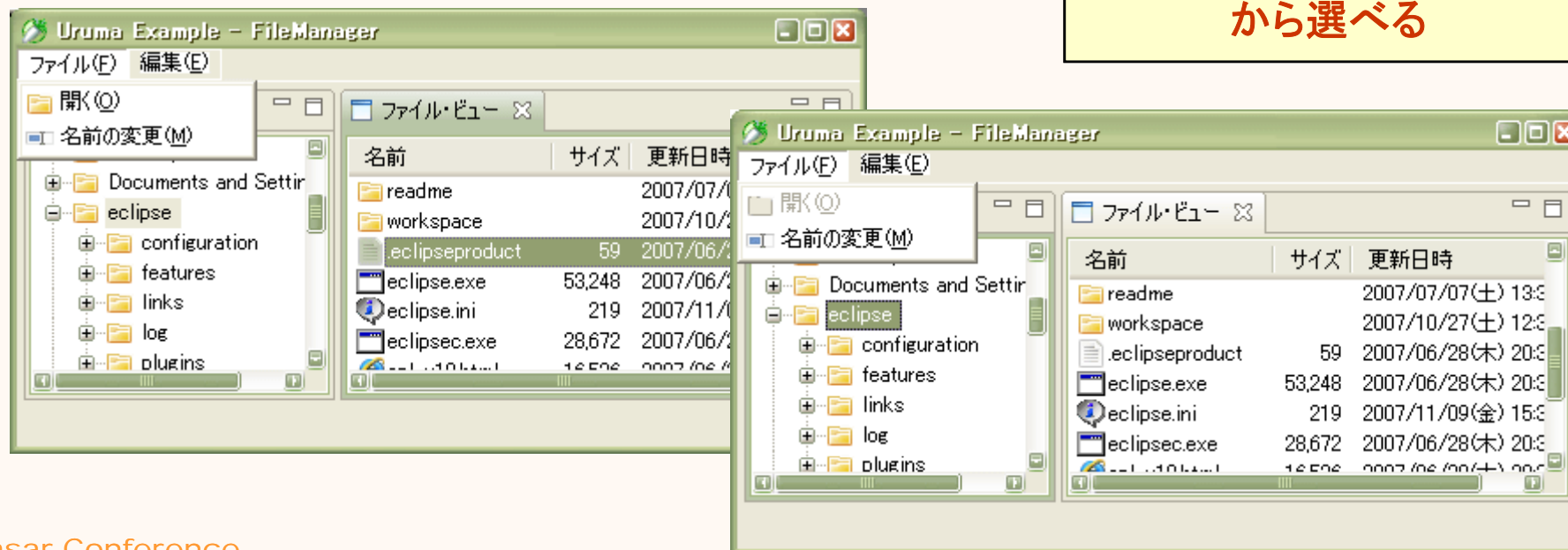
ライブでつくる! Urumaアプリケーション(12) 項目選択時のみメニューを有効にする

workbench.xml

```
<menuItem id="fileOpen" text="開く (&O)" image="folder"  
enablesDependingId="fileDetailTable" enablesFor="SINGLE"/>
```

指定されたテーブルの
選択状態によって
選択可/不可が変化

SELECTION(1項目以上),
SINGLE(1項目のみ),
PAIR(2項目のみ),
MULTI(2項目以上)
から選べる





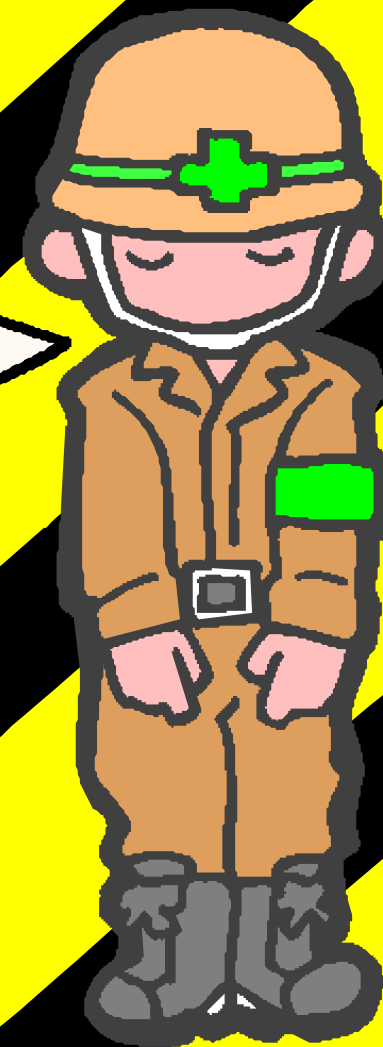
- **開発環境の整備**
 - Uruma Studio(仮称)
- **機能向上**
 - RCPサポート機能追加
 - エディタ対応、プリファレンス、EclipseForms ...
 - バリデーション機能の提供
 - 画面定義テンプレートの簡素化(継承、コンポーネント機能、etc...)
 - GUIテストのサポート
- **品質向上**
 - テスト網羅性向上
 - サンプルの充実と、不具合修正
 - リファレンスマニュアルの整備



- **Webサイト**
 - <http://uruma.sandbox.seasar.org/ja/>
- **ダウンロードサイト(サンプルアプリケーション有)**
 - <http://uruma.sandbox.seasar.org/ja/download.html>
- **開発者ML**
 - uruma-dev@ml.seasar.org
- **Subversion リポジトリ**
 - <https://www.seasar.org/svn/sandbox/uruma/>
- **トラッキング(JIRA)**
 - <https://www.seasar.org/issues/browse/URUMA>

Urumaは現在
Sandboxプロジェクトにて
鋭意開発中です

みなさまからのご意見、
ご要望をお待ちしております





質問コーナーで
お待ちしております
気軽にお越しく下さい



ご静聴
ありがとうございました