What is Wazaabi 2.0?

Declarative UI framework

based on

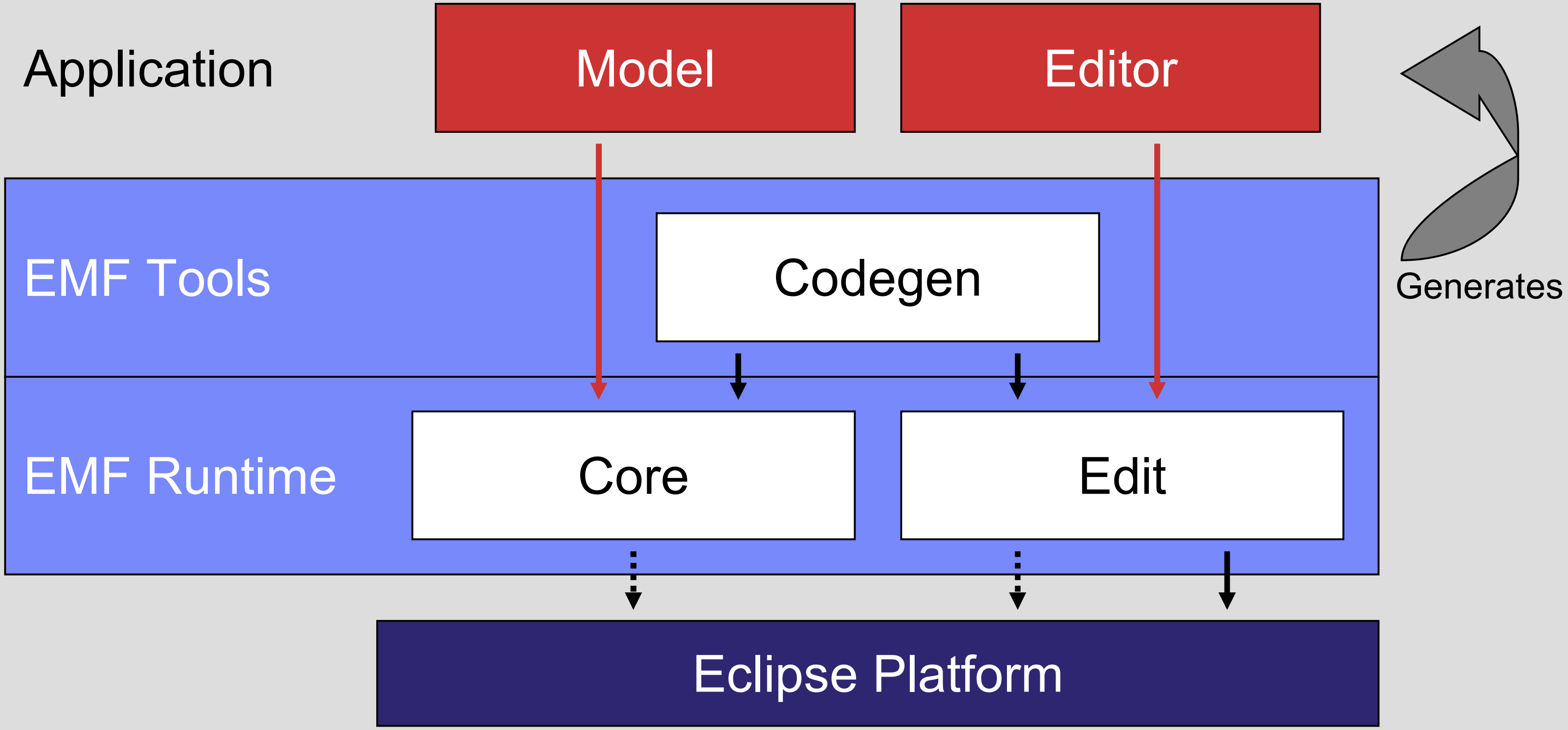
live EMF model(s)

What is an EMF “Model”?

- Specification of an application’s data Object attributes
- Relationships (associations) between objects
- Operations available on each object
- Simple constraints (e.g. multiplicity) on objects and relationships
- Essentially, the Class Diagram subset of UML

Extract from « Fundamentals of the Eclipse Modeling Framework » © IBM Corp. 2006-2008

EMF Architecture



Application

Model

Editor

Generates

EMF Tools

Codegen

EMF Runtime

Core

Edit

Eclipse Platform

Extract from « Fundamentals of the Eclipse Modeling Framework » © IBM Corp. 2006-2008

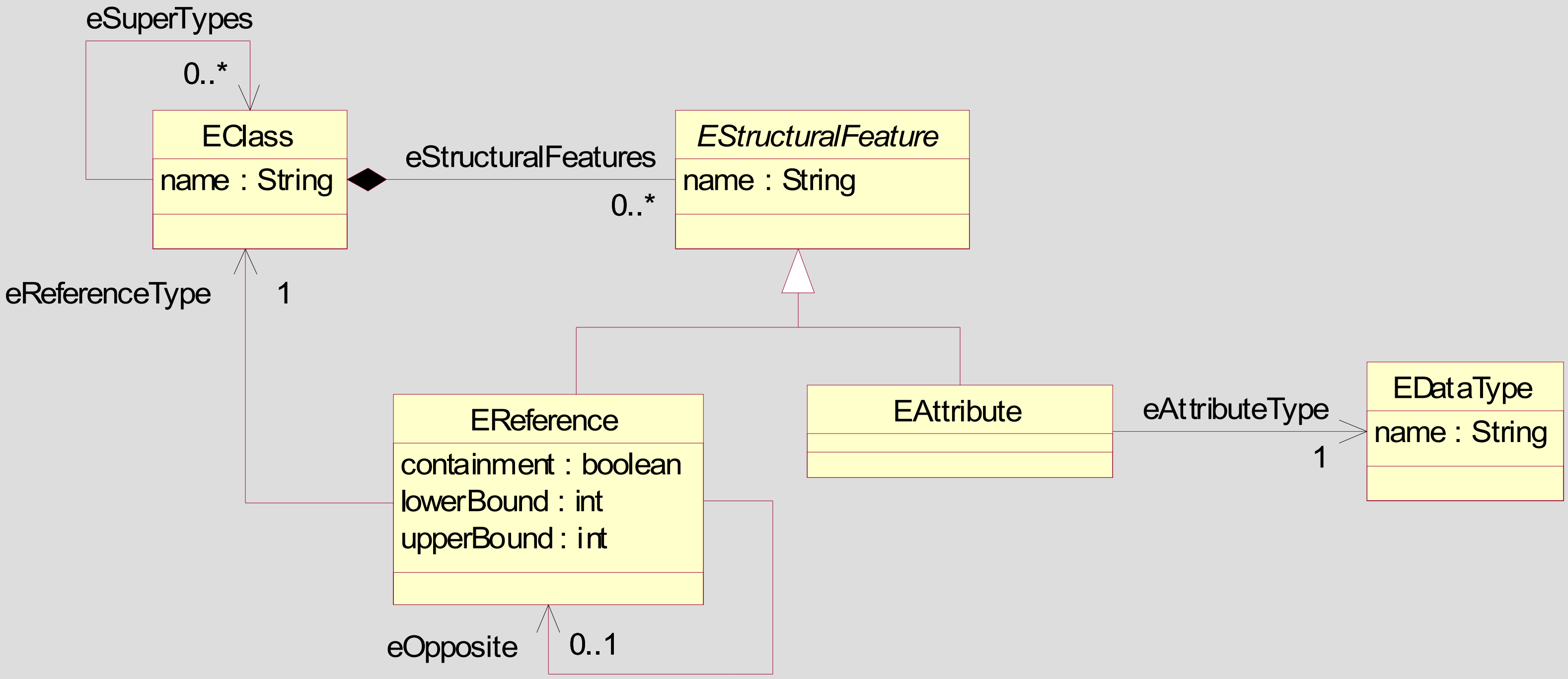
EMF Components

- Core Runtime
  - Notification framework
  - Ecore metamodel
  - Persistence (XML/XMI), validation, change model
- EMF.Edit
  - Support for model-based editors and viewers
  - Default reflective editor
- Codegen
  - Code generator for application models and editors
  - Extensible model importer/exporter framework

Extract from « Fundamentals of the Eclipse Modeling Framework » © IBM Corp. 2006-2008

Ecore

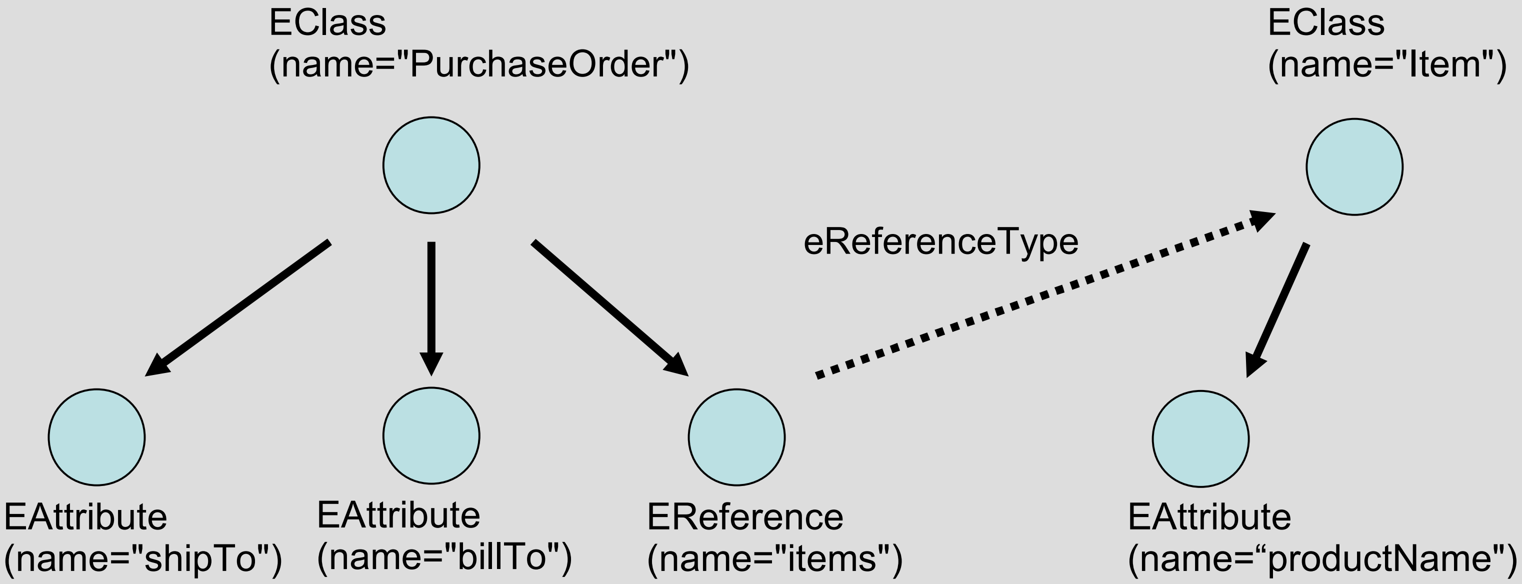
- EMF’s metamodel (model of a model)



Extract from « Fundamentals of the Eclipse Modeling Framework » © IBM Corp. 2006-2008

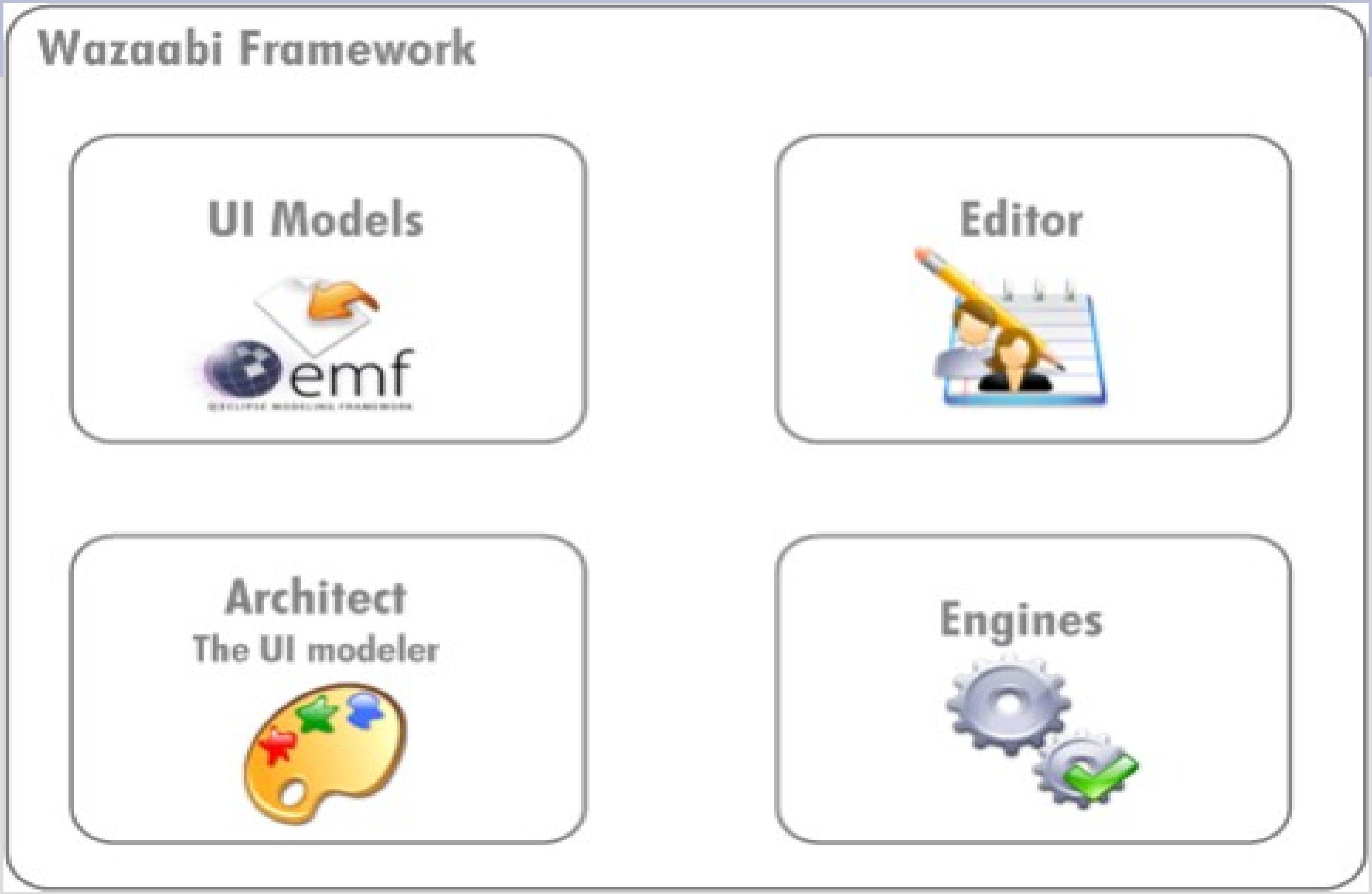
Ecore

- Application models (e.g. purchase order model) are instances of Ecore

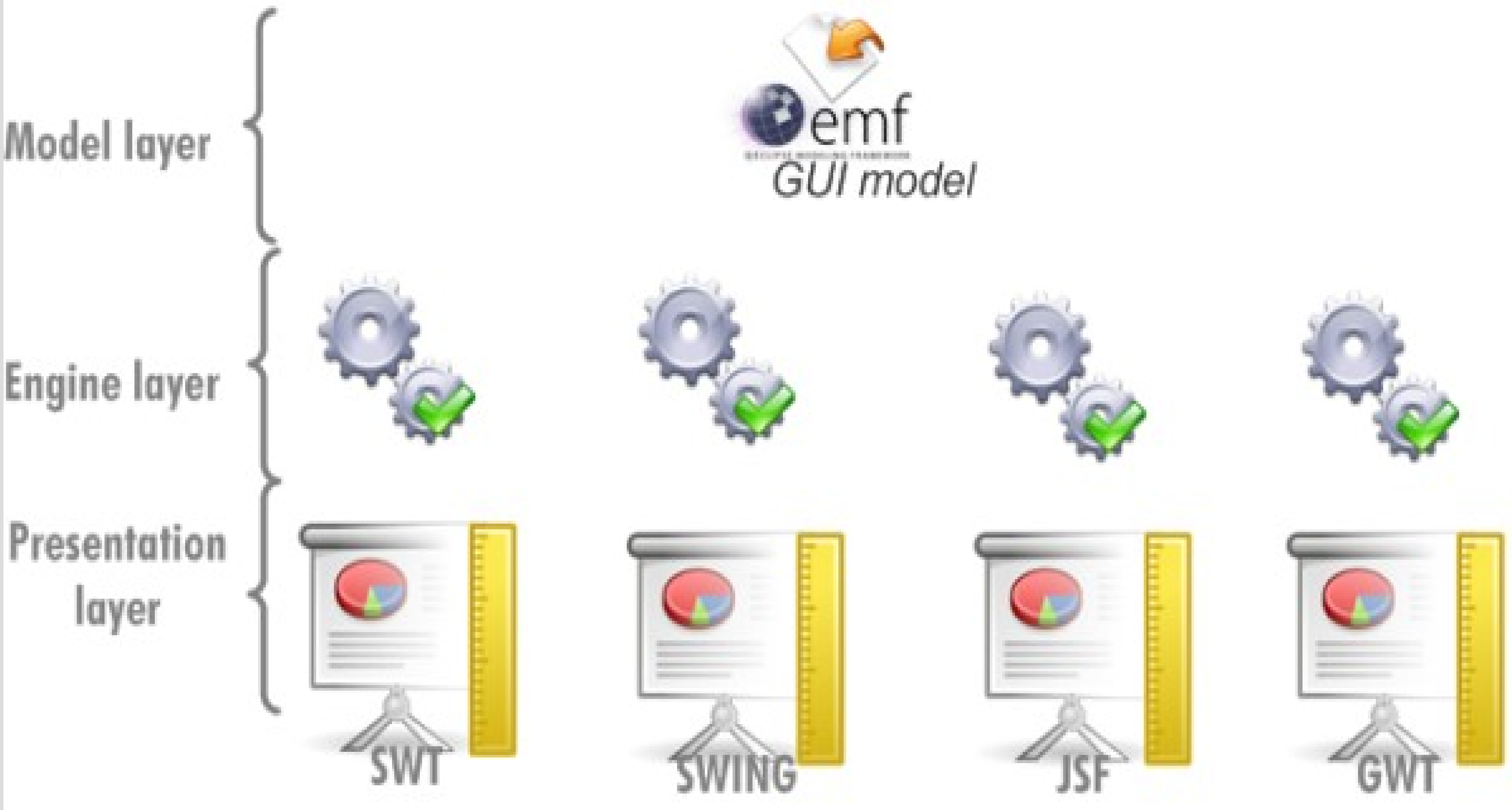


Extract from « Fundamentals of the Eclipse Modeling Framework » © IBM Corp. 2006-2008

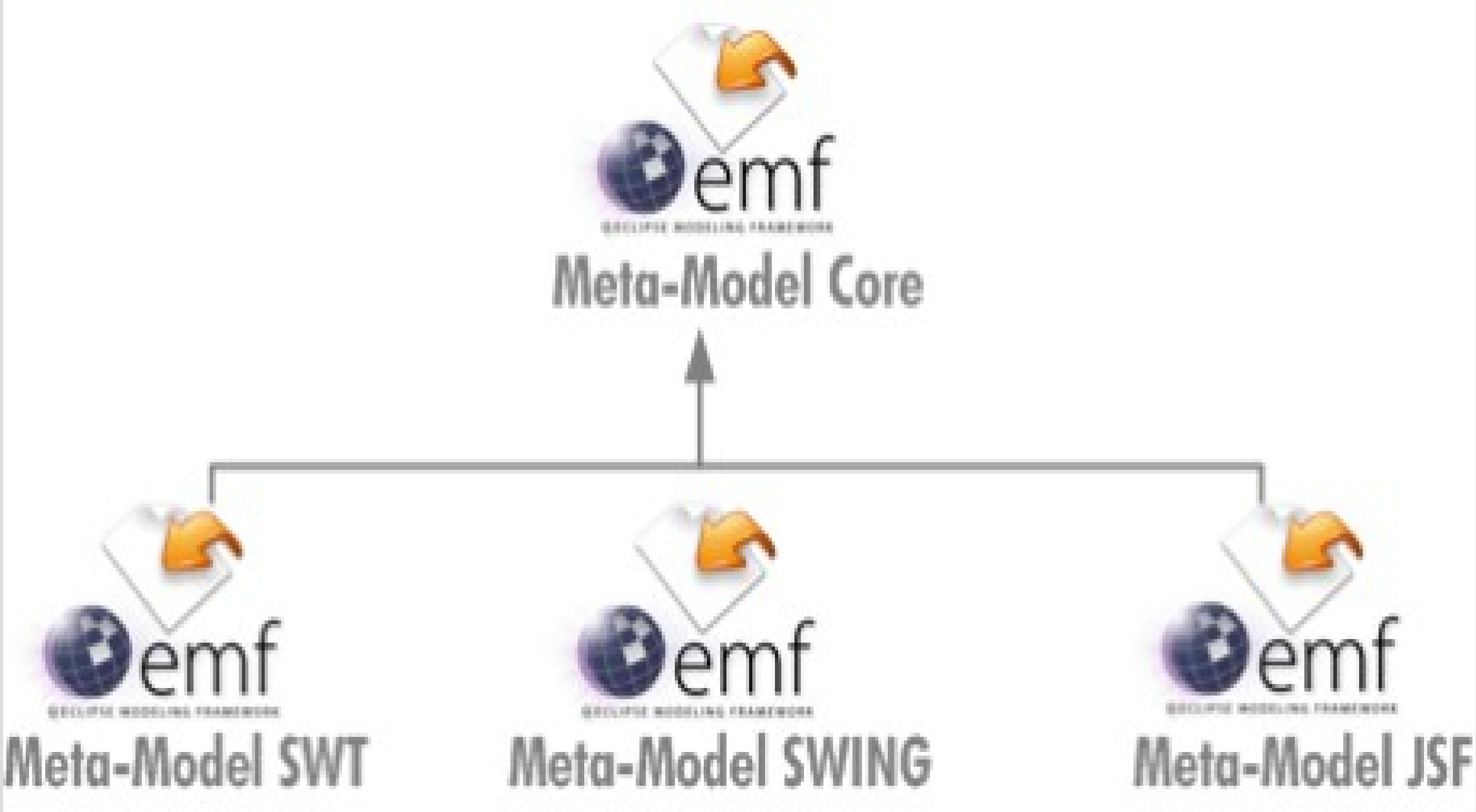
Wazaabi overview



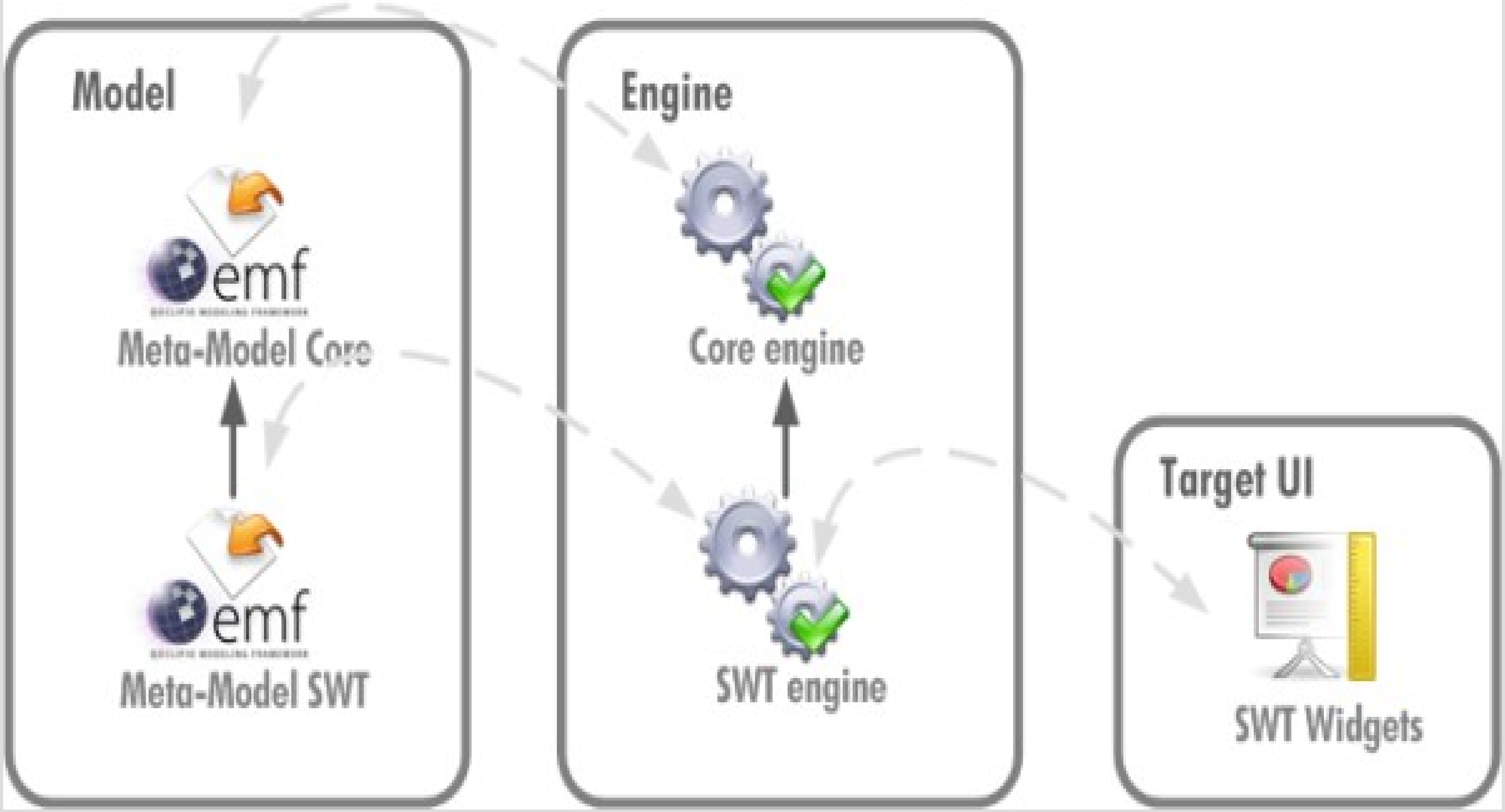
Wazaabi overview



UI's metamodels


emf
ECLIPSE MODELING FRAMEWORK
Meta-Model Core
emf
ECLIPSE MODELING FRAMEWORK
Meta-Model SWT
emf
ECLIPSE MODELING FRAMEWORK
Meta-Model SWING
emf
ECLIPSE MODELING FRAMEWORK
Meta-Model JSF

Project's architecture



Wazaabi 2.0 key concepts

Postulate :

In Wazaabi 2.0 everything is a (live) model

- Widgets & properties
- Layouts & Layout data
- Binding, workflow & data flow

Live UI models, what does it mean ?

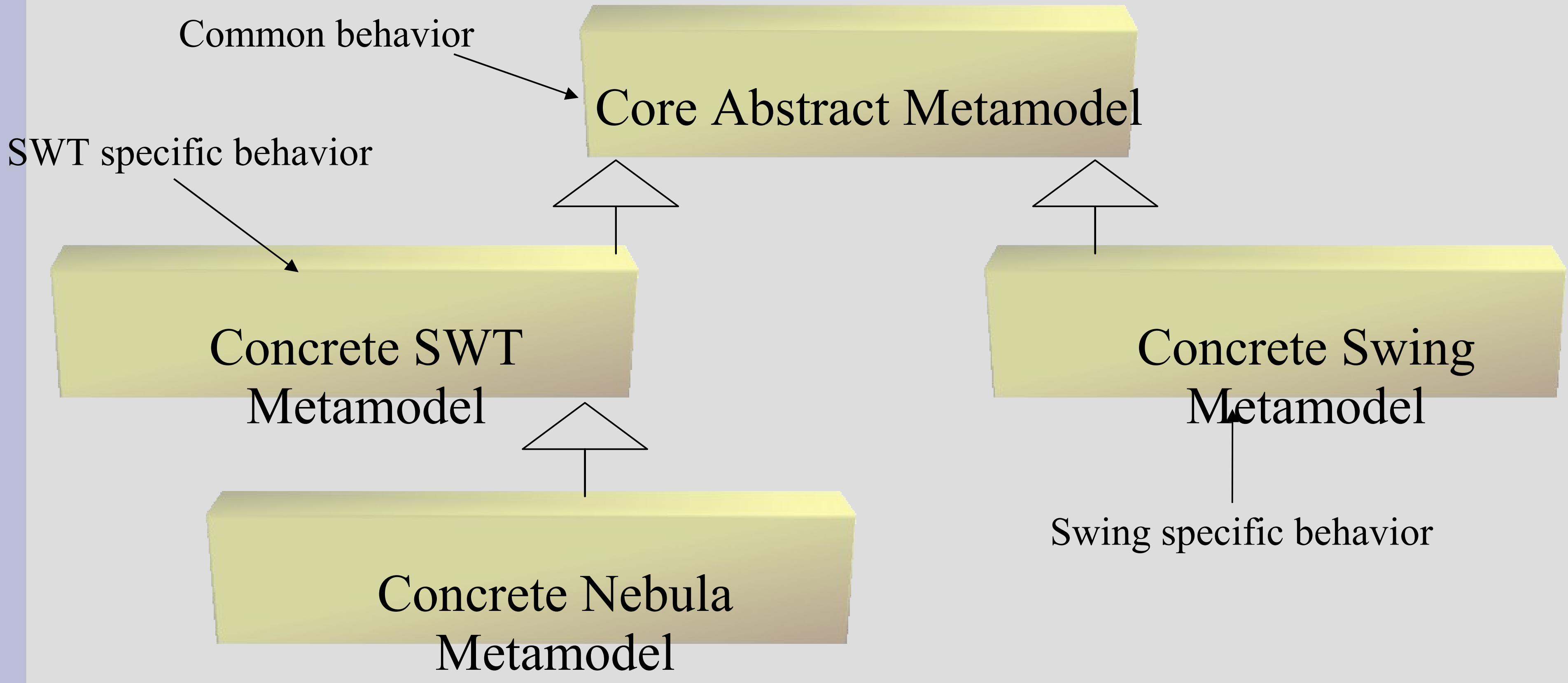
- Any change of the model is reflected in the UI:
  - Widget property
  - layout or layout data
  - structural change (UI part addition or removal)
  - etc...
- Any action of the user triggers a model change
  - Text fields,
  - selection in a list,
  - button actioned
  - etc...

How does it work ?

- A Model is rendered by a viewer, a visual component classically embedded in platform UI
- As much viewers than needed at one time
- Usable in standalone apps, RCP apps and Eclipse IDE
- At the moment 2 viewers implemented:
  - SWT
  - Swing
  - More to come : GEF, GWT, ...

Metamodels and policy of design

Most of visual components share a common behavior independantly of the target plateform (SWT, Swing, etc...)



Uses of wazaabi 2.0

- Declaration of a part or the whole UI
- Wazaabi helps to create a running prototype at a moment of the generation chains
- Wazaabi helps to change the UI when running application in real life (skining, styling, user prefs, etc..)

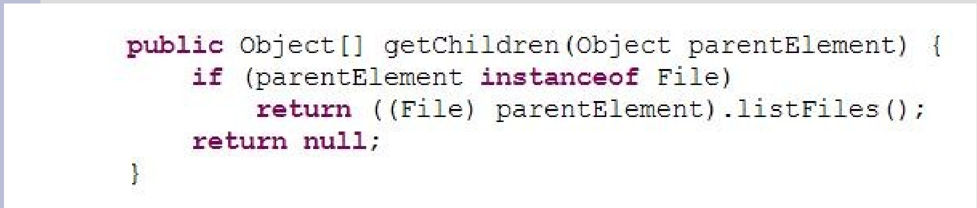
Focus on:
Wazaabi tables: key points

- Content is always domain objects, never items with labels
- No tree or list (or combo) !

  only a unique UI object : Table
- The designer chooses (at runtime) between hierarchical or flat view of the same content

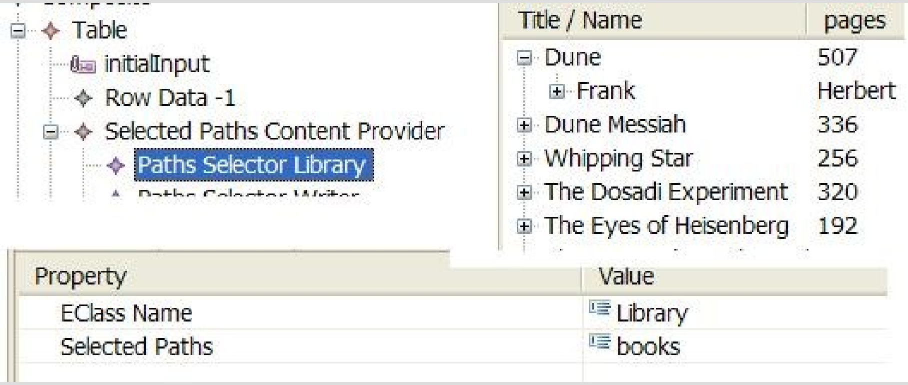
Populating Wazaabi tables (1)

JFace IContentProviders
& ILabelProviders



```
public Object[] getChildren(Object parentElement) {
    if (parentElement instanceof File)
        return ((File) parentElement).listFiles();
    return null;
}
```

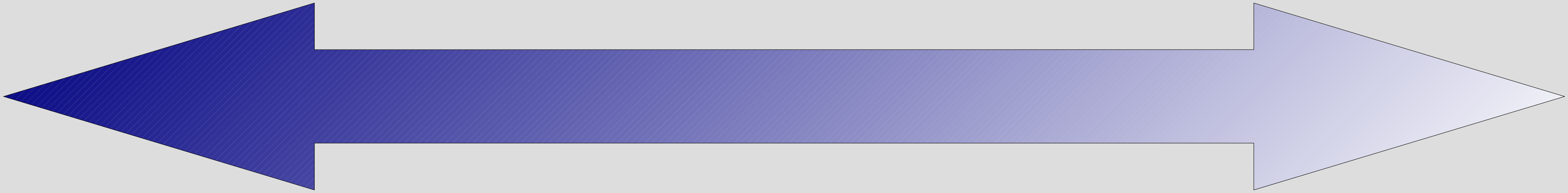
Declaratively based
on domain objects exploration
and live !!



```
Table
  initialInput
  Row Data -1
  Selected Paths Content Provider
    Paths Selector Library
    Paths Selector Writer
```

Title / Name	pages
Dune	507
Frank	Herbert
Dune Messiah	336
Whipping Star	256
The Dosadi Experiment	320
The Eyes of Heisenberg	192

Property	Value
EClass Name	Library
Selected Paths	books



Coding approach

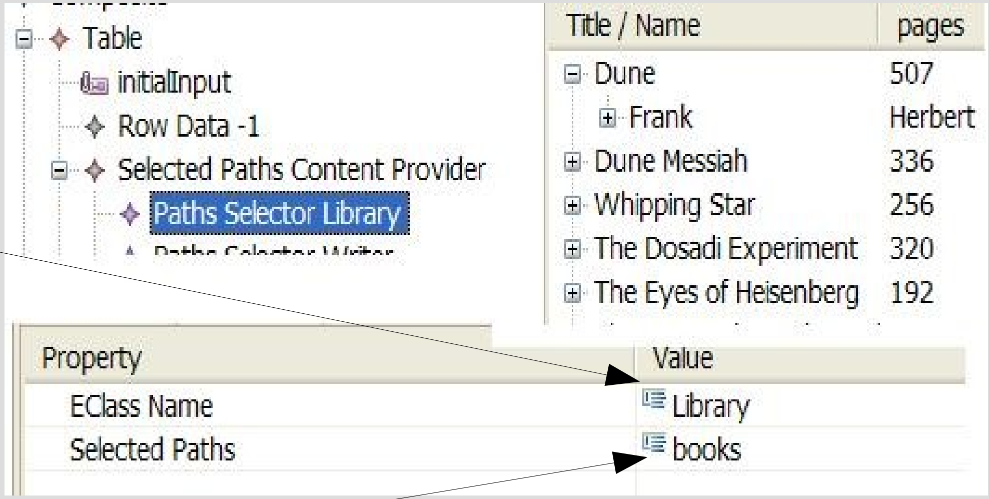
Declarative approach

Populating Wazaabi tables (2)

Intermediate way : A table could be declaratively bound to existing JFace Content & Label providers (platform:/plugin/..../myContentProvider)

Populating Wazaabi tables (3)

Declarative approach: getting domain object content



- Table
  - initialInput
  - Row Data -1
  - Selected Paths Content Provider
    - Paths Selector Library
    - Paths Selector Writer

Title / Name	pages
Dune	507
Frank	Herbert
Dune Messiah	336
Whipping Star	256
The Dosadi Experiment	320
The Eyes of Heisenberg	192

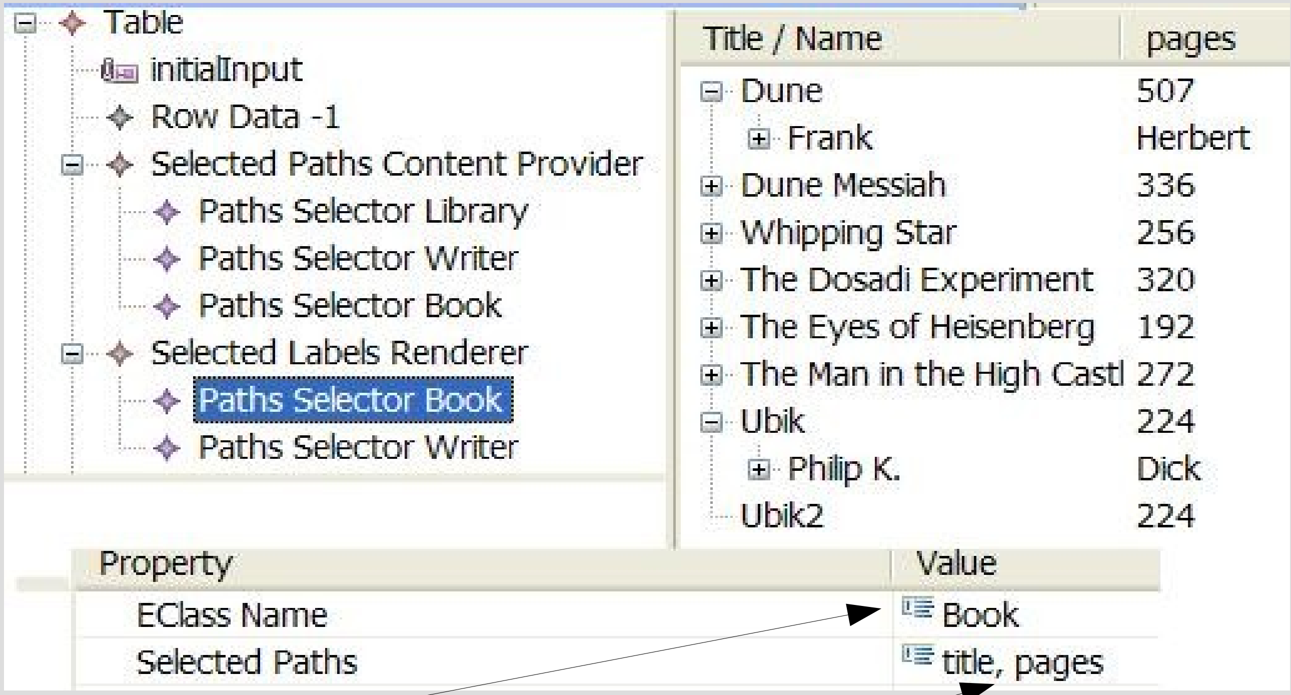
Property	Value
EClass Name	Library
Selected Paths	books

When the domain object is a

get children from

Populating Wazaabi tables (4)

Declarative approach: displaying object properties



- Table
  - initialInput
  - Row Data -1
  - Selected Paths Content Provider
    - Paths Selector Library
    - Paths Selector Writer
    - Paths Selector Book
  - Selected Labels Renderer
    - Paths Selector Book
    - Paths Selector Writer

Title / Name	pages
⊟ Dune	507
⊞ Frank	Herbert
⊞ Dune Messiah	336
⊞ Whipping Star	256
⊞ The Dosadi Experiment	320
⊞ The Eyes of Heisenberg	192
⊞ The Man in the High Castl	272
⊟ Ubik	224
⊞ Philip K.	Dick
Ubik2	224

Property	Value
EClass Name	Book
Selected Paths	title, pages

When the domain object is a

Display (in column order)

Demo & Questions

http://www.wazaabi.org

Olivier Moïses
olivier@wazaabi.org