Recovery and Migration of Application Logic from Legacy Systems

Wiktor Nowakowski , Michał Śmiałek, Albert Ambroziewicz, Norbert Jarzębowski, Tomasz Straszak

Warsaw University of Technology

Krajowa Konferencja Inżynierii Oprogramowania

Kraków, Poland, September 10-13, 2012

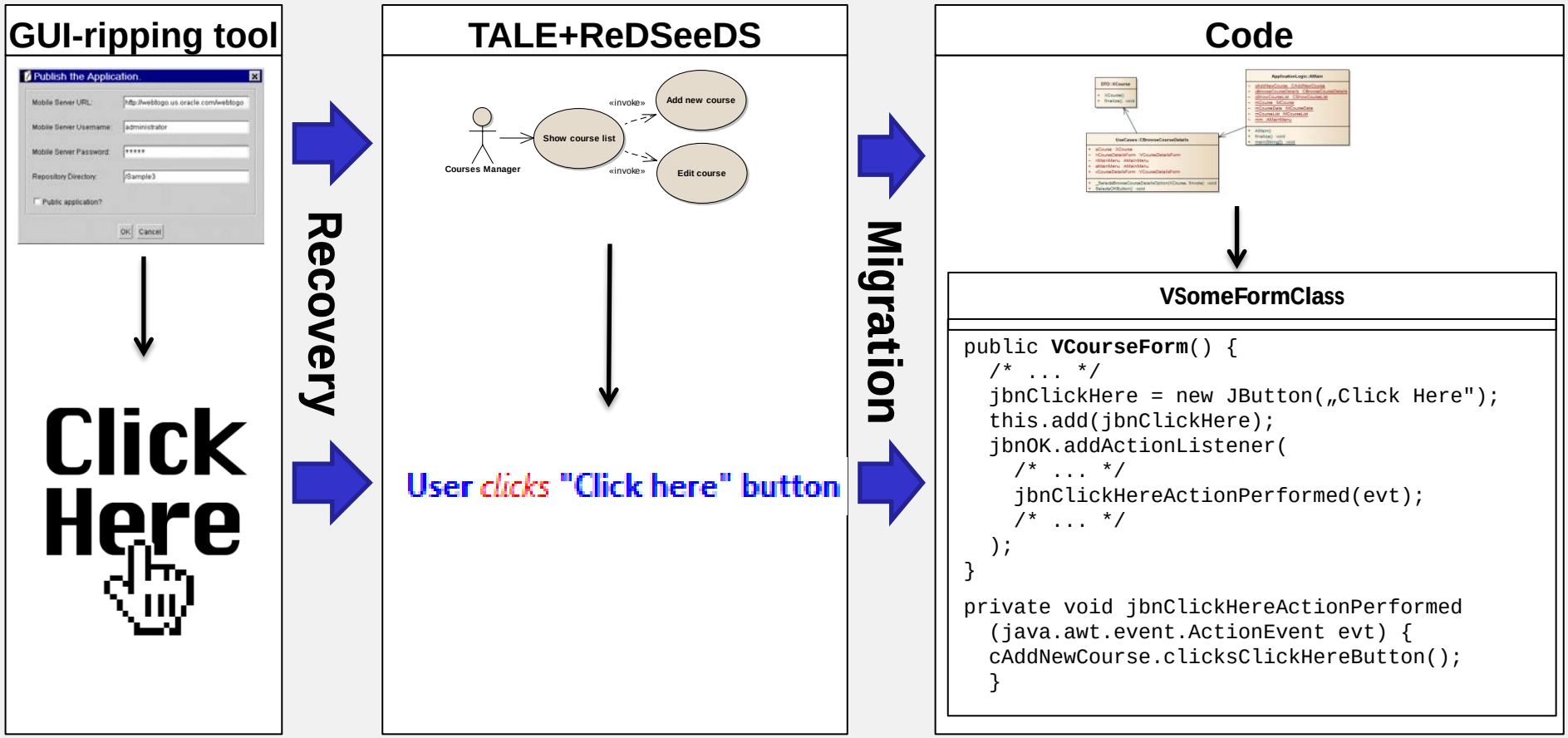
Overview

- Idea & Solution overview
- Essence
- Recovery from legacy system
- Migration to modern technology system
- Summary

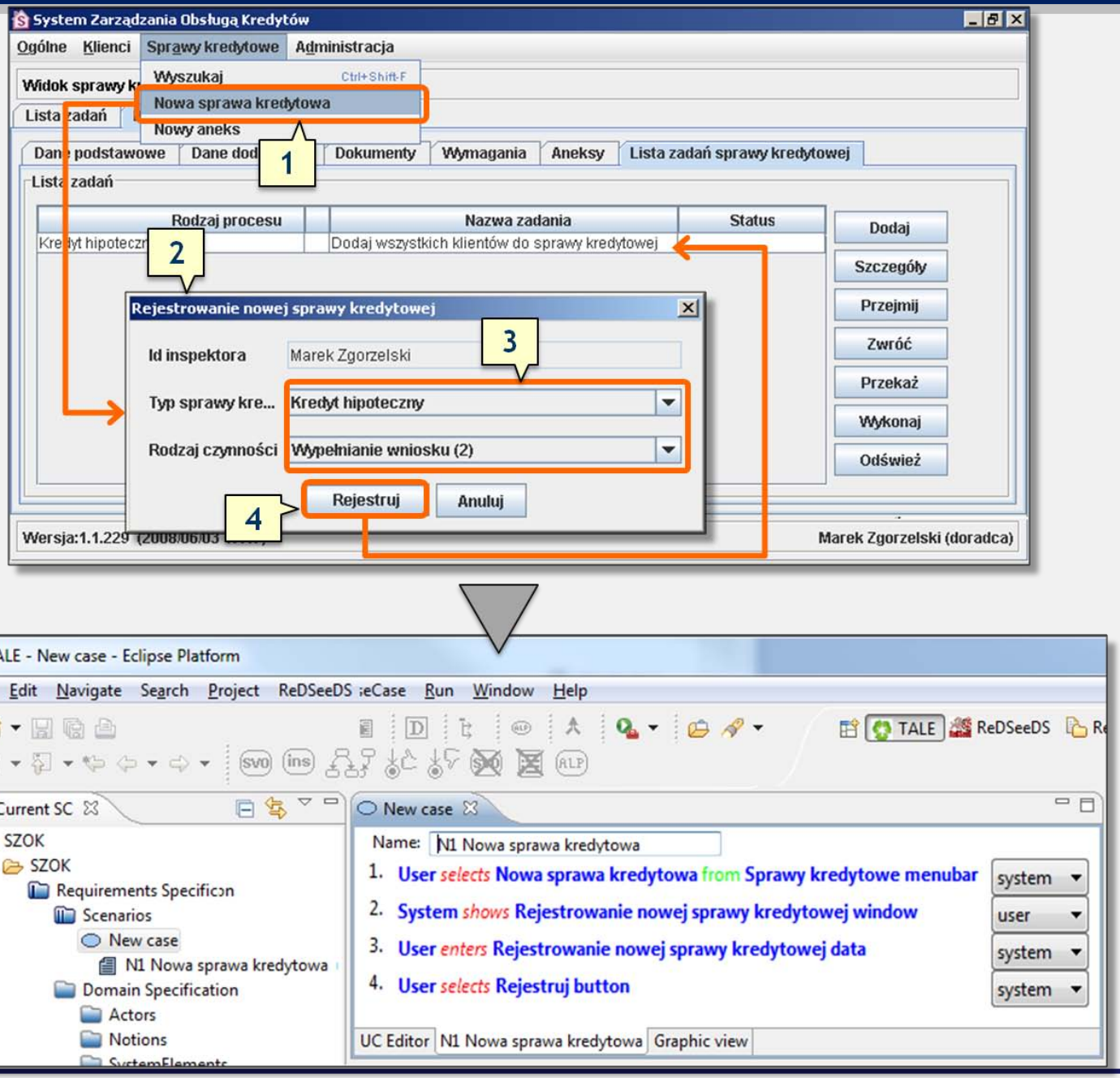
Idea

- Software recycling
- Developed within the REMICS project
- Reuse & Migration of Legacy Software Systems
- 7th Framework Program, Information & Communication Technologies

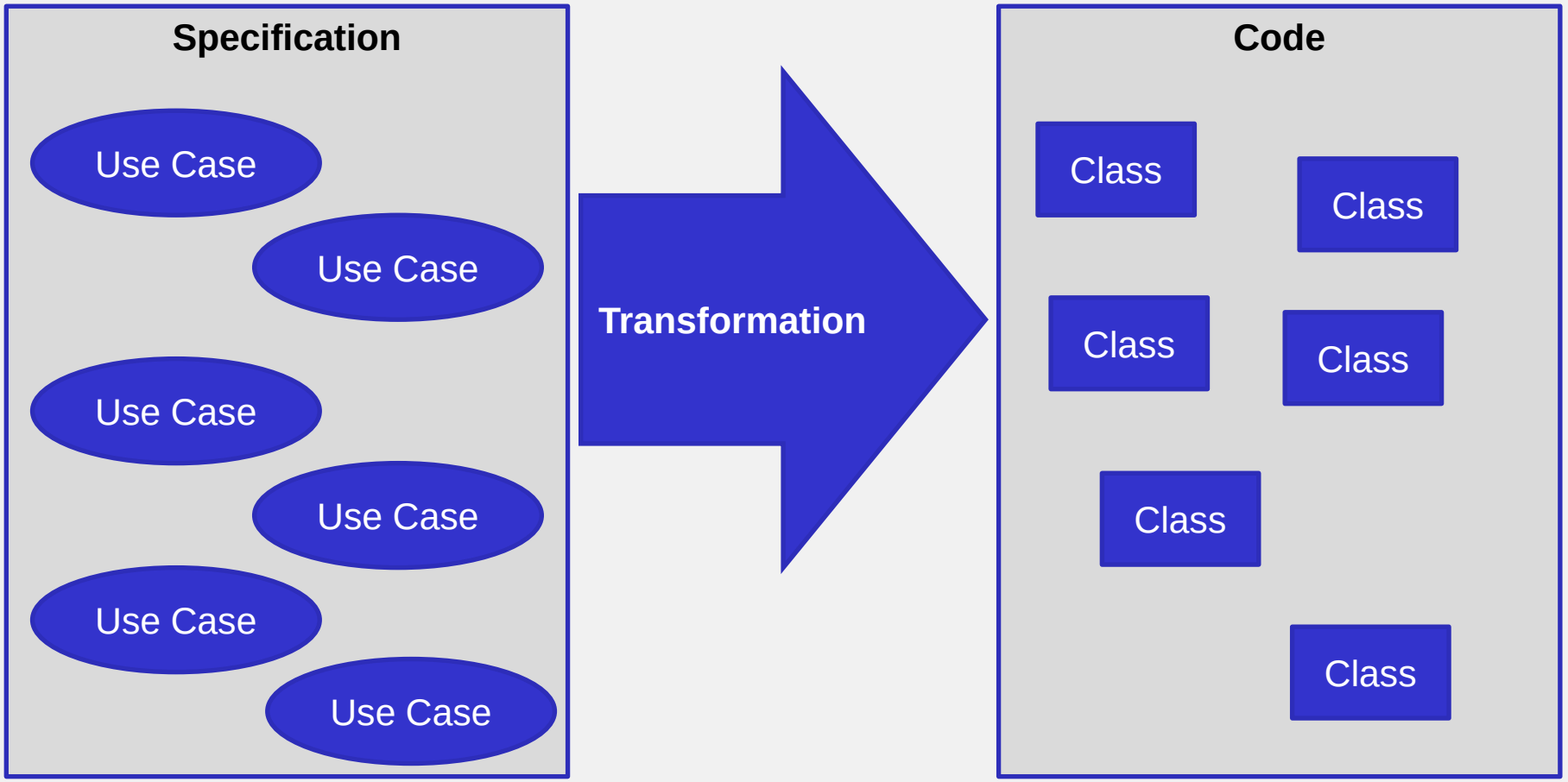
Solution overview



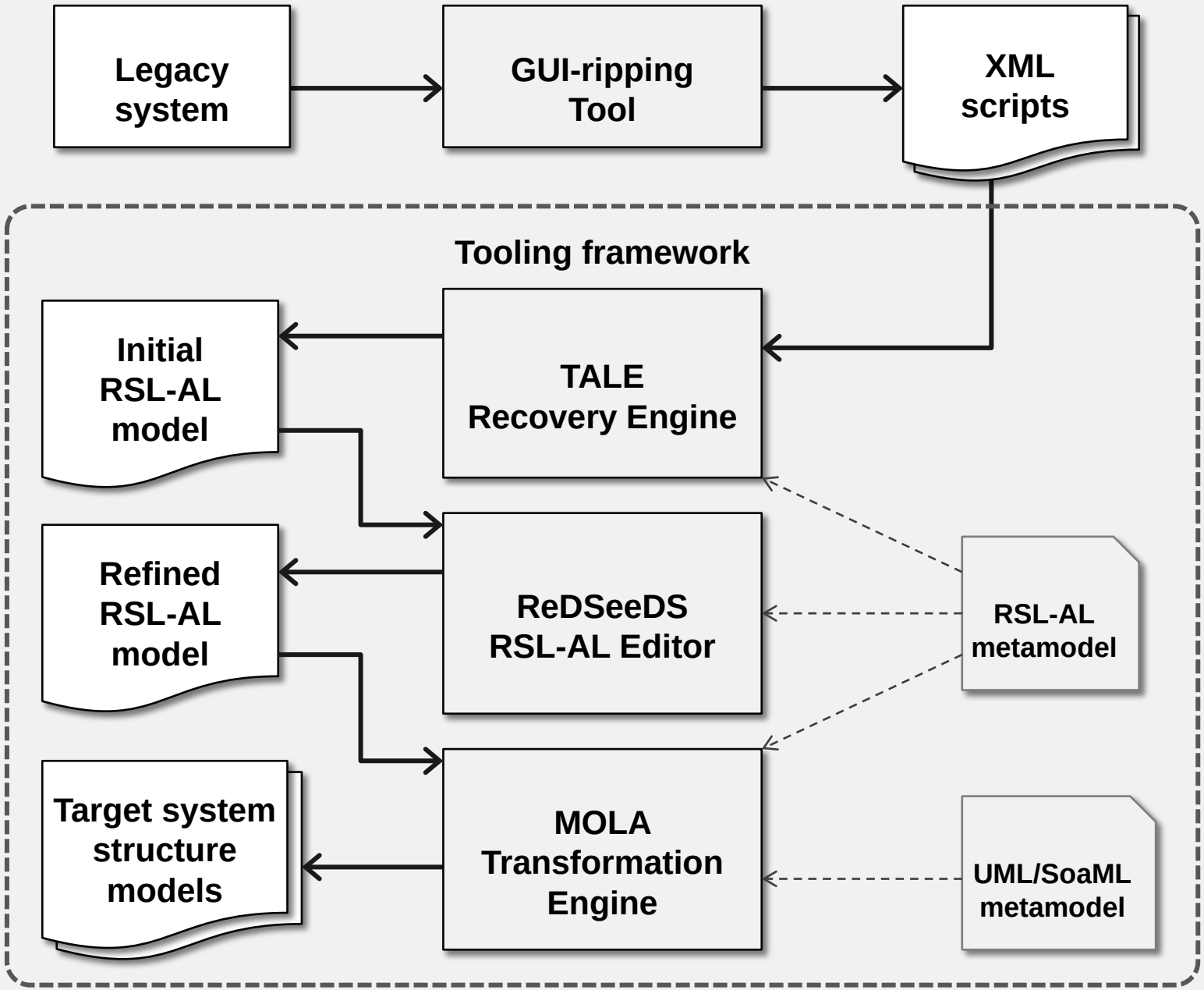
Recovery


System Zarządzania Obsługą Kredytów
Ogólne
Klienci
Sprawy kredytowe
Administracja
Wyszukaj
Ctrl+Shift-F
Nowa sprawa kredytowa
Nowy aneks
1
Dokumenty
Wymagania
Aneksy
Lista zadań sprawy kredytowej
Lista zadań
Rodzaj procesu
Nazwa zadania
Status
Dodaj wszystkich klientów do sprawy kredytowej
2
Dodaj
Szczegóły
Przejmij
Zwróć
Przekaż
Wykonaj
Odśwież
Rejestrowanie nowej sprawy kredytowej
Id inspektora
Marek Zgorzelski
3
Typ sprawy kre...
Kredyt hipoteczny
Rodzaj czynności
Wypełnianie wniosku (2)
4
Rejestruj
Anuluj
Marek Zgorzelski (doradca)
TALE - New case - Eclipse Platform
File
Edit
Navigate
Search
Project
ReDSeeDS
Run
Window
Help
Current SC
SZOK
Requirements Specificn
Scenarios
New case
N1 Nowa sprawa kredytowa
Domain Specification
Actors
Notions
New case
Name:
N1 Nowa sprawa kredytowa
1. User selects Nowa sprawa kredytowa from Sprawy kredytowe menubar
system
2. System shows Rejestrowanie nowej sprawy kredytowej window
user
3. User enters Rejestrowanie nowej sprawy kredytowej data
system
4. User selects Rejestruj button
system
UC Editor
N1 Nowa sprawa kredytowa
Graphic view

Migration

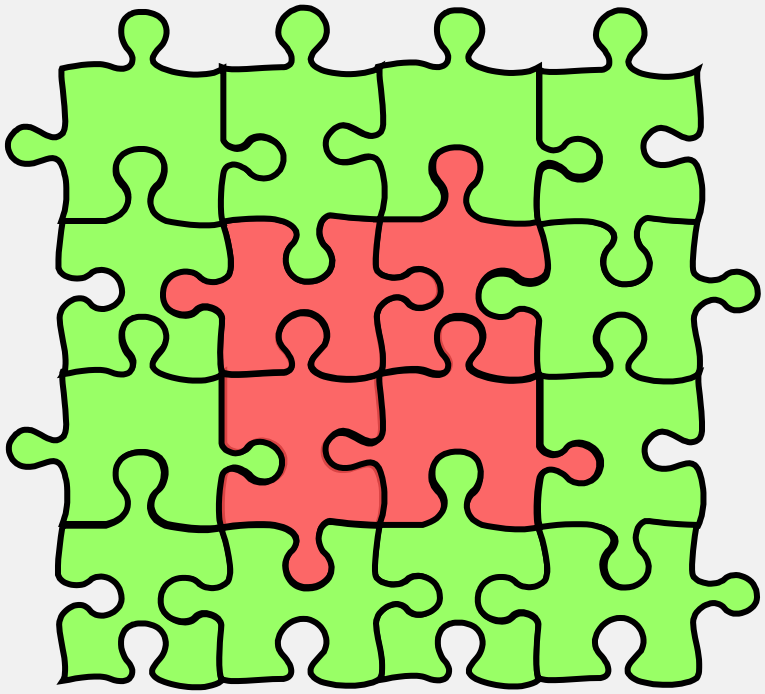

Specification
Use Case
Use Case
Use Case
Use Case
Use Case
Use Case
Transformation
Code
Class
Class
Class
Class
Class
Class

Metodology: Process



Essence

Essential and accidental complexity

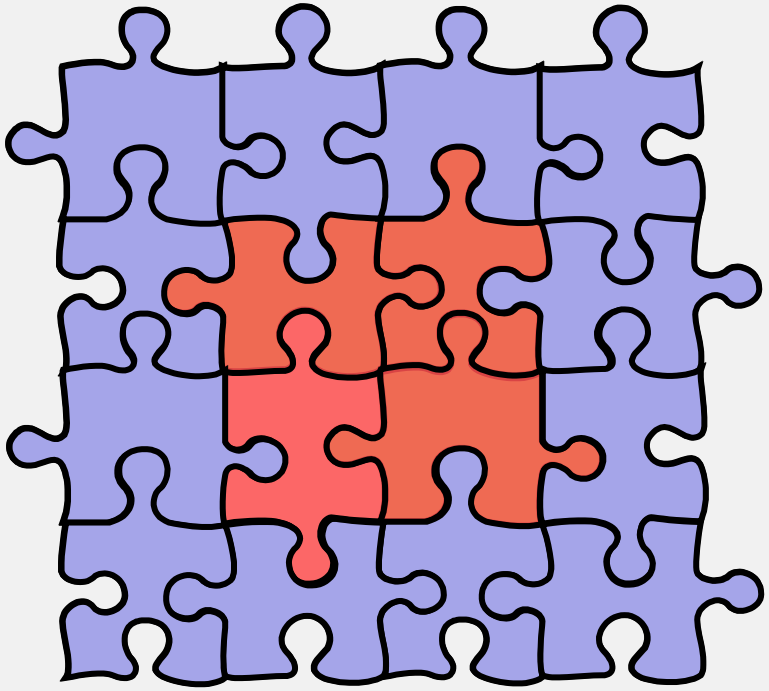


„The ***essence*** of a software entity is a construct of interlocking concepts: data sets, relationships among data items, algorithms, and invocations of functions.
[..] such a conceptual construct is the same under many different representations.” *

„[..] ***accidental*** tasks arise in representing the construct in language.” *

* F.P. Brooks, ”No silver bullet: Essence and accidents of software engineering”

The „essence” in legacy systems



Some characteristics of legacy systems:

- **poorly structured (eg. complex monolithic structure)**
- **technology obsolescence (poor interoperability)**
- **loss of knowledge (both technical and business)**

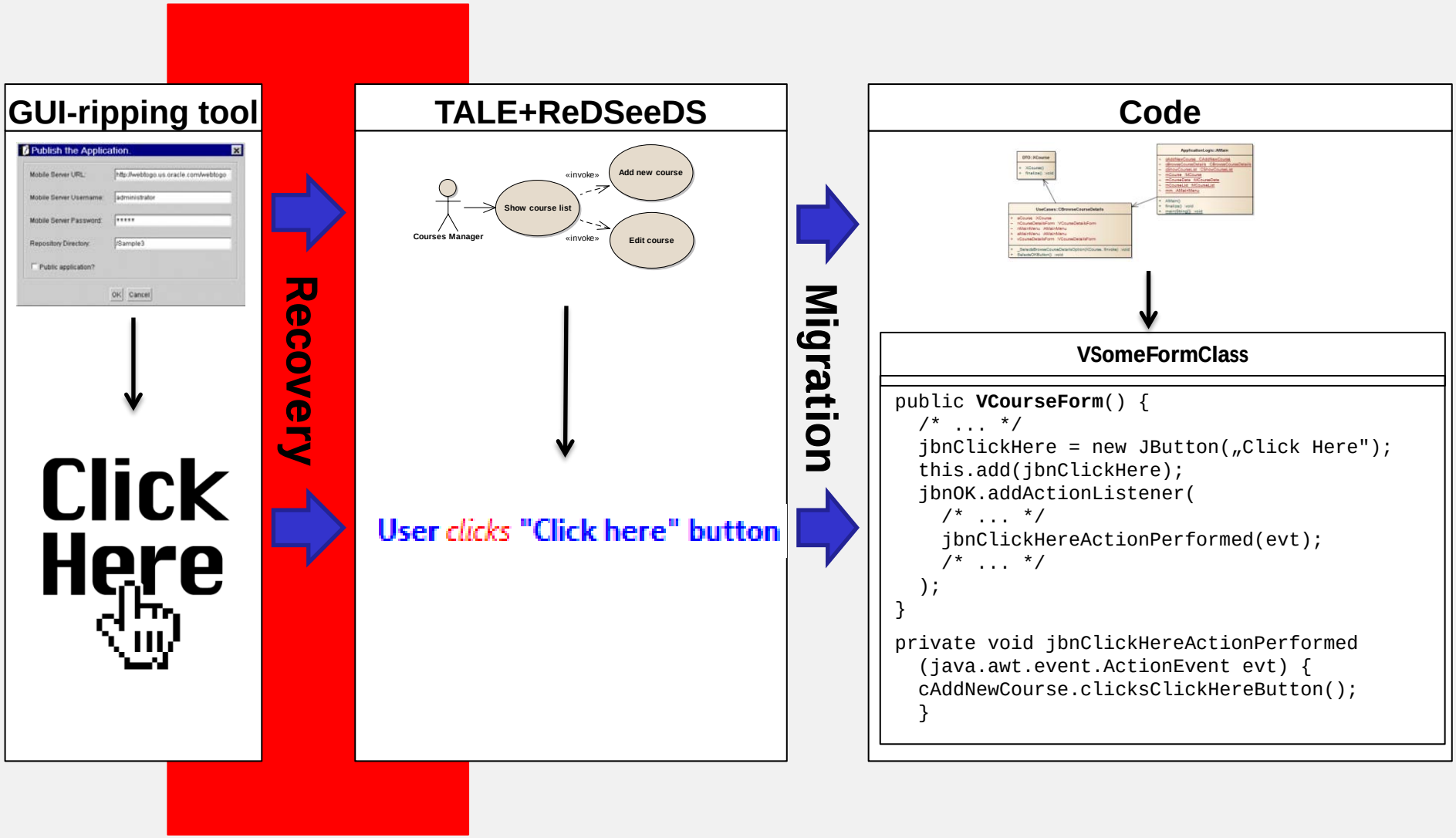
How to capture the essence of a legacy systems?

How to cater this essence for new functionality?

How to migrate it into a new technology?

Recovery from legacy system

Solution overview



GUI-ripping tool

- IBM Rational Functional Tester (RFT)
- Tool for automated testing of software applications
- Supports a broad range of applications such as Web-based, .Net, Java, terminal emulator based applications, Siebel, SAP, PowerBuilder, AJAX, Adobe Flex, Dojo Toolkit, GEF, Adobe PDF documents
- Recording mechanism creates a test script from the actions
- Object map containts information about GUI elements

From test scripts to requirements

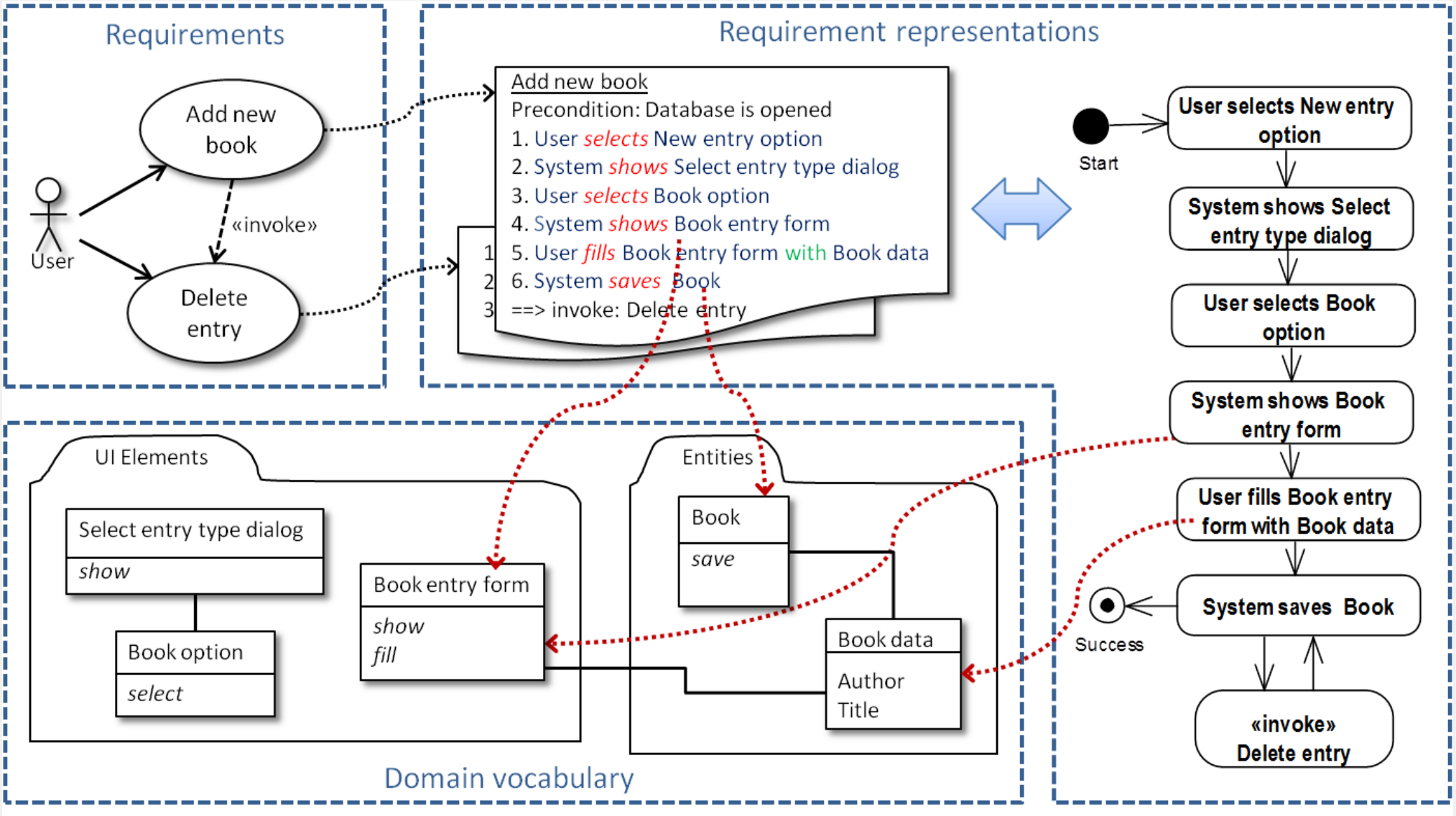
```
  ...
<testElements xsi:type="com.ibm.rational.test.ft.visualscript:ProxyMethod" name="book" type="GuiTestObject"
role="Button" elementType="TestObject" domain="Java" controlName="Book" topLevelWindow="//@topLevelWindows.2">
      <action name="click"/>
</testElements>
<testElements xsi:type="com.ibm.rational.test.ft.visualscript:ProxyMethod" name="jabRefUntitled"
type="TopLevelTestObject" role="Frame" elementType="TestObject" domain="Java" controlName="JabRef - untitled"
topLevelWindow="//@topLevelWindows.1">
      <action name="inputChars">
        <argument>
          <testelement xsi:type="com.ibm.rational.test.ft.visualscript:Value" value="&quot;Tytul&quot;"
elementType="Value" valueType="String"/>
        </argument>
      </action>
  ...
```

1. User selects New entry from BibTex
2. System shows Select entry type
3. User selects Book
4. System shows JabRef - untitled
5. User enters JabRef - untitled

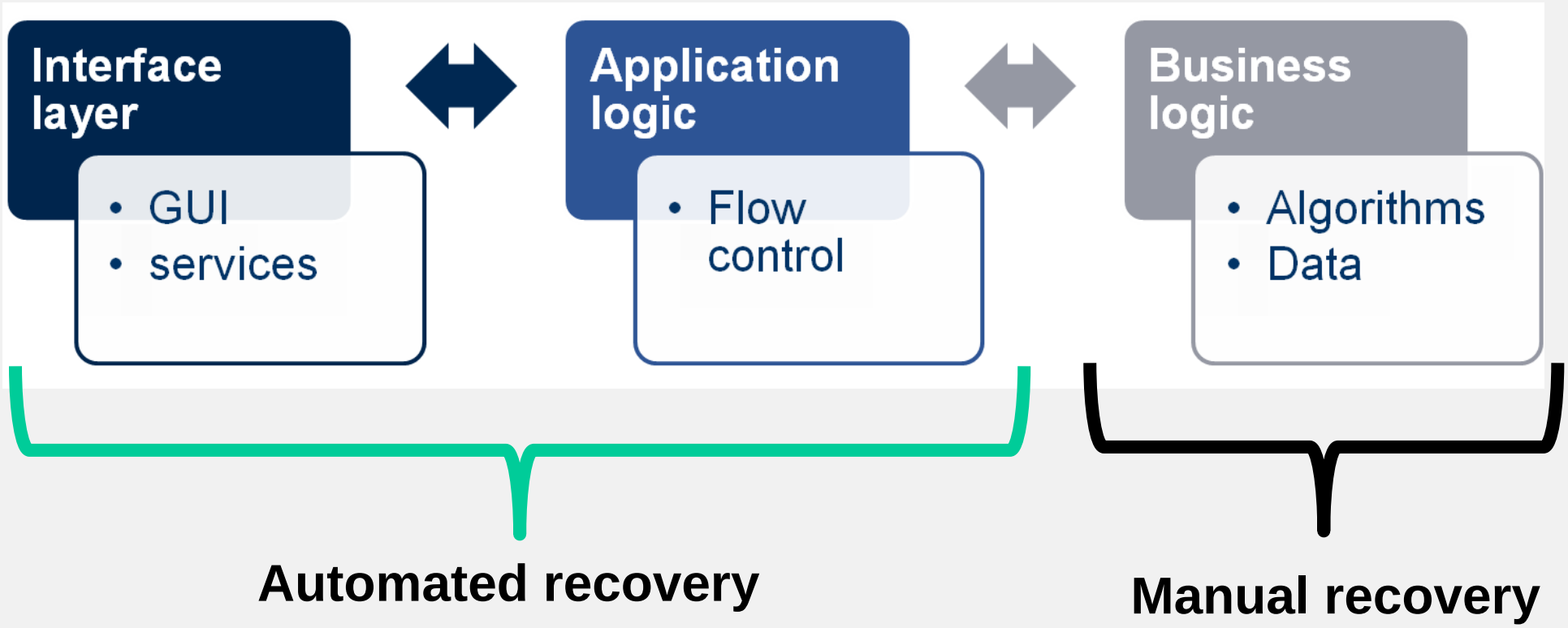
TALE & ReDSeeDS

- Eclipse based (perspectives)
- Integrated use cases, notions and diagrams editors
- Implements RSL metamodel
- Notions with meanings connected to WordNet database
- Enables requirements to code transformations
- Enables model export to Enterprise Architect

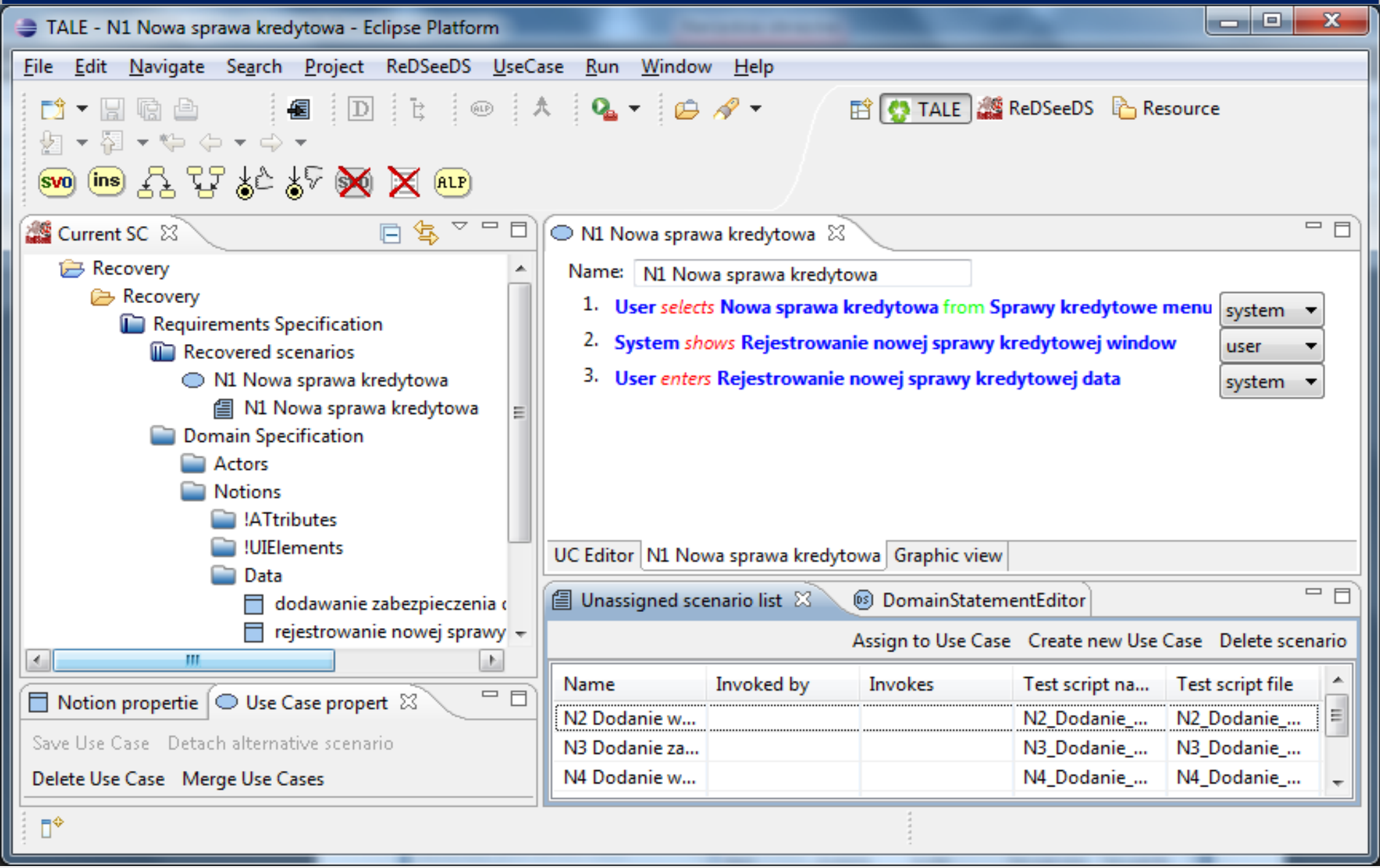
Recovered Notions



Recovery

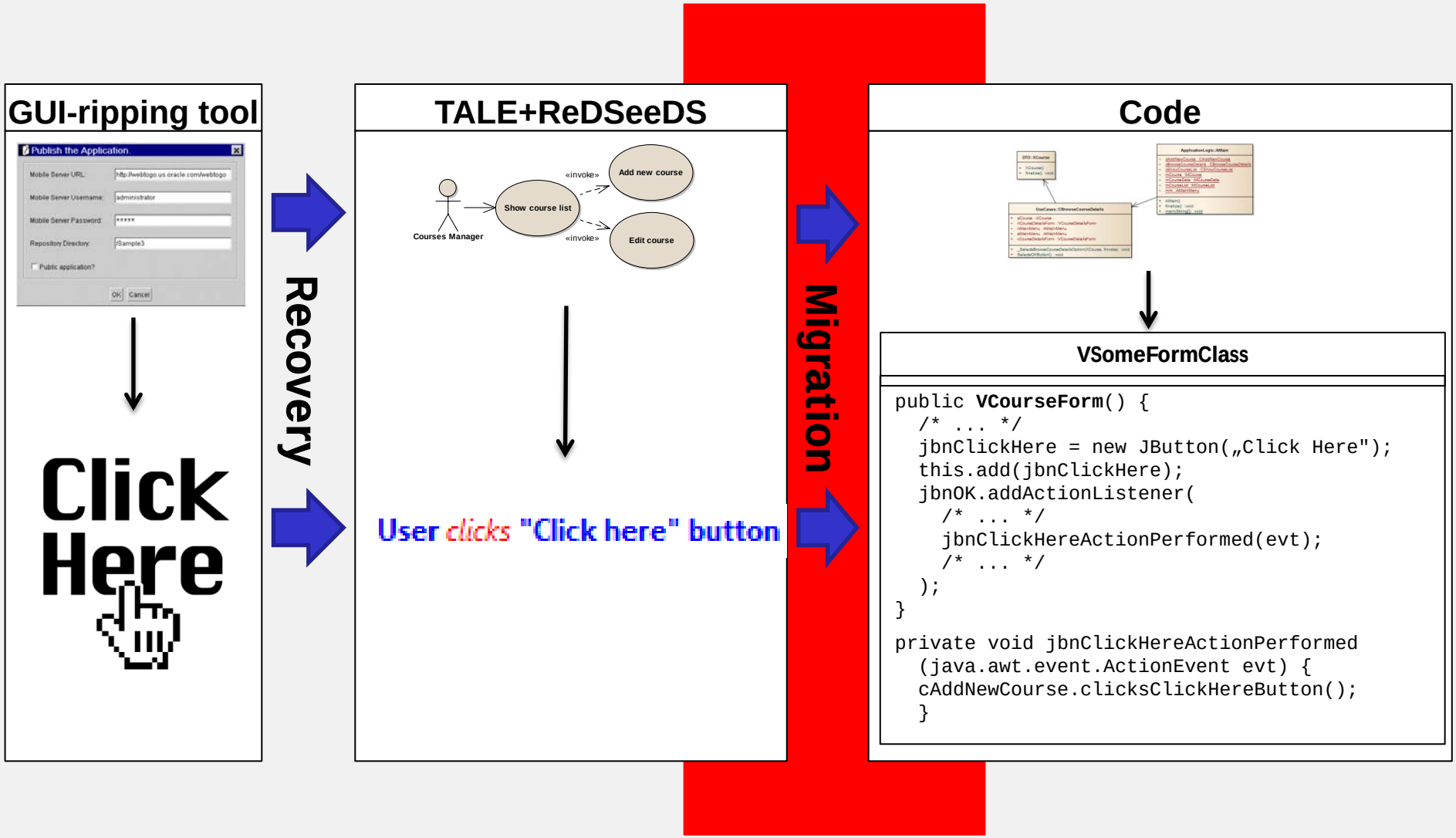

Interface layer
• GUI
• services
Application logic
• Flow control
Business logic
• Algorithms
• Data
Automated recovery
Manual recovery

TALE tool

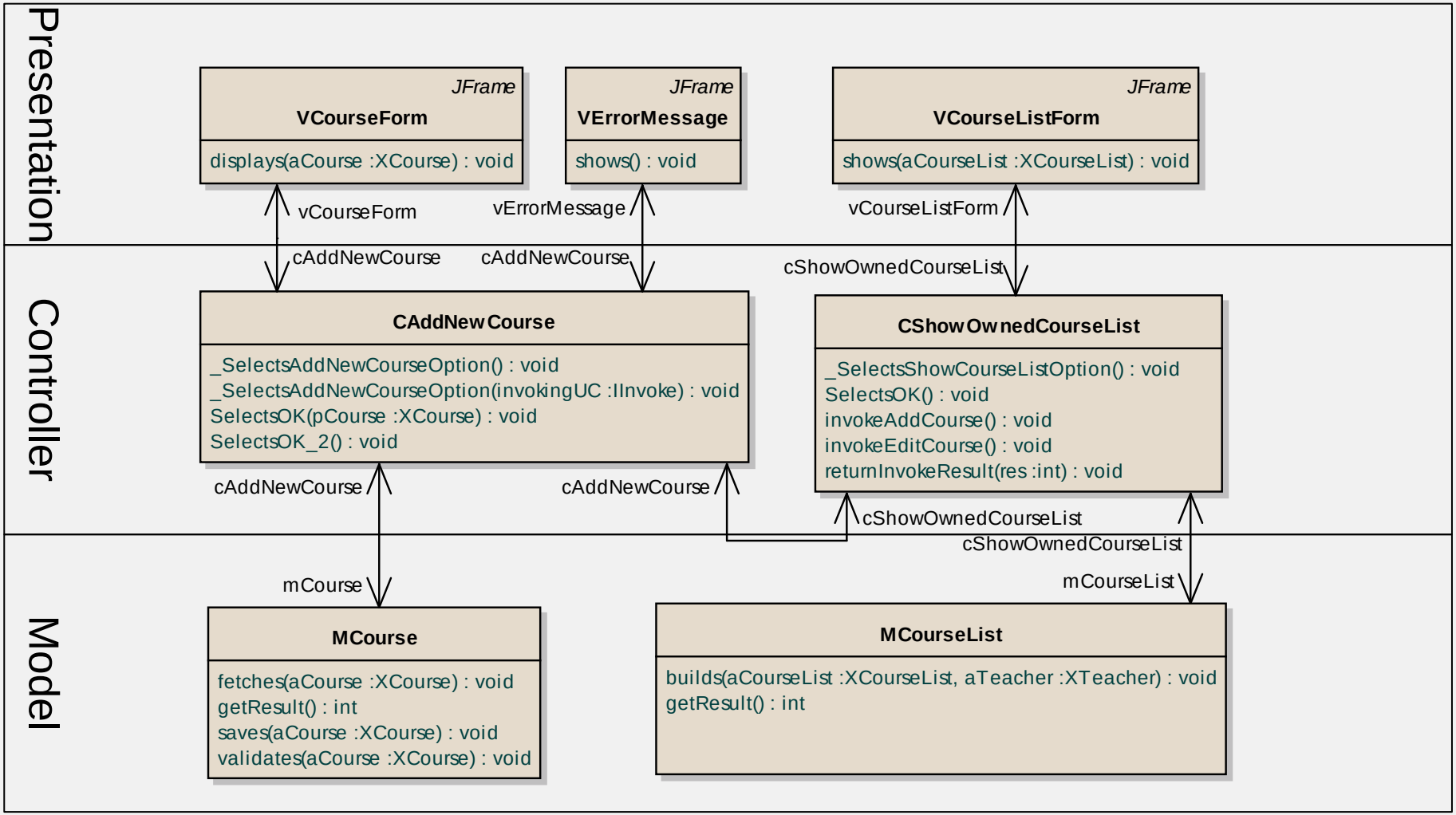

TALE - N1 Nowa sprawa kredytowa - Eclipse Platform
File Edit Navigate Search Project ReDSeeDS UseCase Run Window Help
TALE
ReDSeeDS
Resource
Current SC
Recovery
Recovery
Requirements Specification
Recovered scenarios
N1 Nowa sprawa kredytowa
N1 Nowa sprawa kredytowa
Domain Specification
Actors
Notions
!ATtributes
!UIElements
Data
dodawanie zabezpieczenia
rejestrowanie nowej sprawy
N1 Nowa sprawa kredytowa
Name: N1 Nowa sprawa kredytowa
1. User selects Nowa sprawa kredytowa from Sprawy kredytowe menu
system
2. System shows Rejestrowanie nowej sprawy kredytowej window
user
3. User enters Rejestrowanie nowej sprawy kredytowej data
system
UC Editor
N1 Nowa sprawa kredytowa
Graphic view
Unassigned scenario list
DomainStatementEditor
Assign to Use Case
Create new Use Case
Delete scenario
Name
Invoked by
Invokes
Test script na...
Test script file
N2 Dodanie w...
N2_Dodanie_...
N2_Dodanie_...
N3 Dodanie za...
N3_Dodanie_...
N3_Dodanie_...
N4 Dodanie w...
N4_Dodanie_...
N4_Dodanie_...
Notion propertie
Use Case propert
Save Use Case
Detach alternative scenario
Delete Use Case
Merge Use Cases

Migration to modern technology system

Solution overview

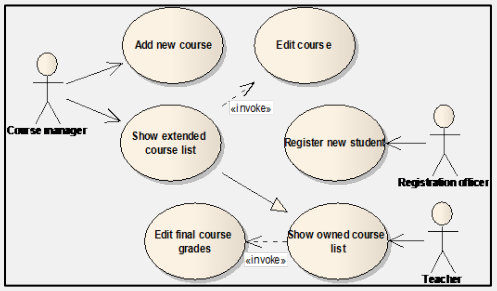


3-layer architecture



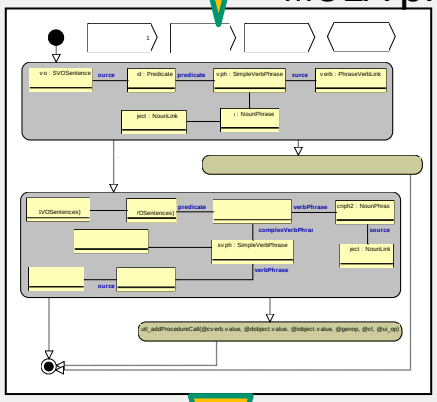
How it works?

Use Case model



Transformation rules

MOLA program

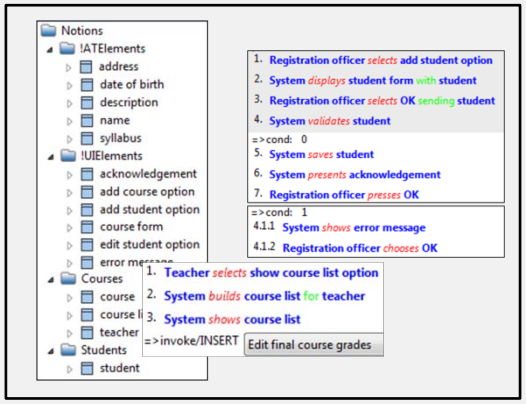


Java code

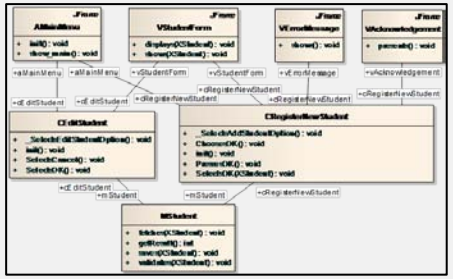
```
package App.ApplicationLogic.StudentManagement;
public class CRegisterNewStudent {
  public XStudent aStudent;
  public VStudentForm vStudentForm;
  public MStudent mStudent;
  public VAcknowledgement vAcknowledgement;
  public VErrorMessage vErrorMessage;

  public void _SelectsAddStudentOption(){
    int res;
    vStudentForm = new VStudentForm();
    vStudentForm.cRegisterNewStudent = this;
    vStudentForm.displays(aStudent);
  }

  public void SelectsOK(XStudent pStudent){
    int res;
    aStudent = pStudent;
    mStudent.validates(aStudent); res = mStudent.getResult();
    if (res == 0) {
      mStudent.saves(aStudent); res = mStudent.getResult();
      vAcknowledgement = new VAcknowledgement();
      vAcknowledgement.cRegisterNewStudent = this;
      vAcknowledgement.presents();
    } else if (res == 1) {
      vErrorMessage = new VErrorMessage();
      vErrorMessage.cRegisterNewStudent = this;
      vErrorMessage.shows();
    }
  }
}
```



RSL scenario model

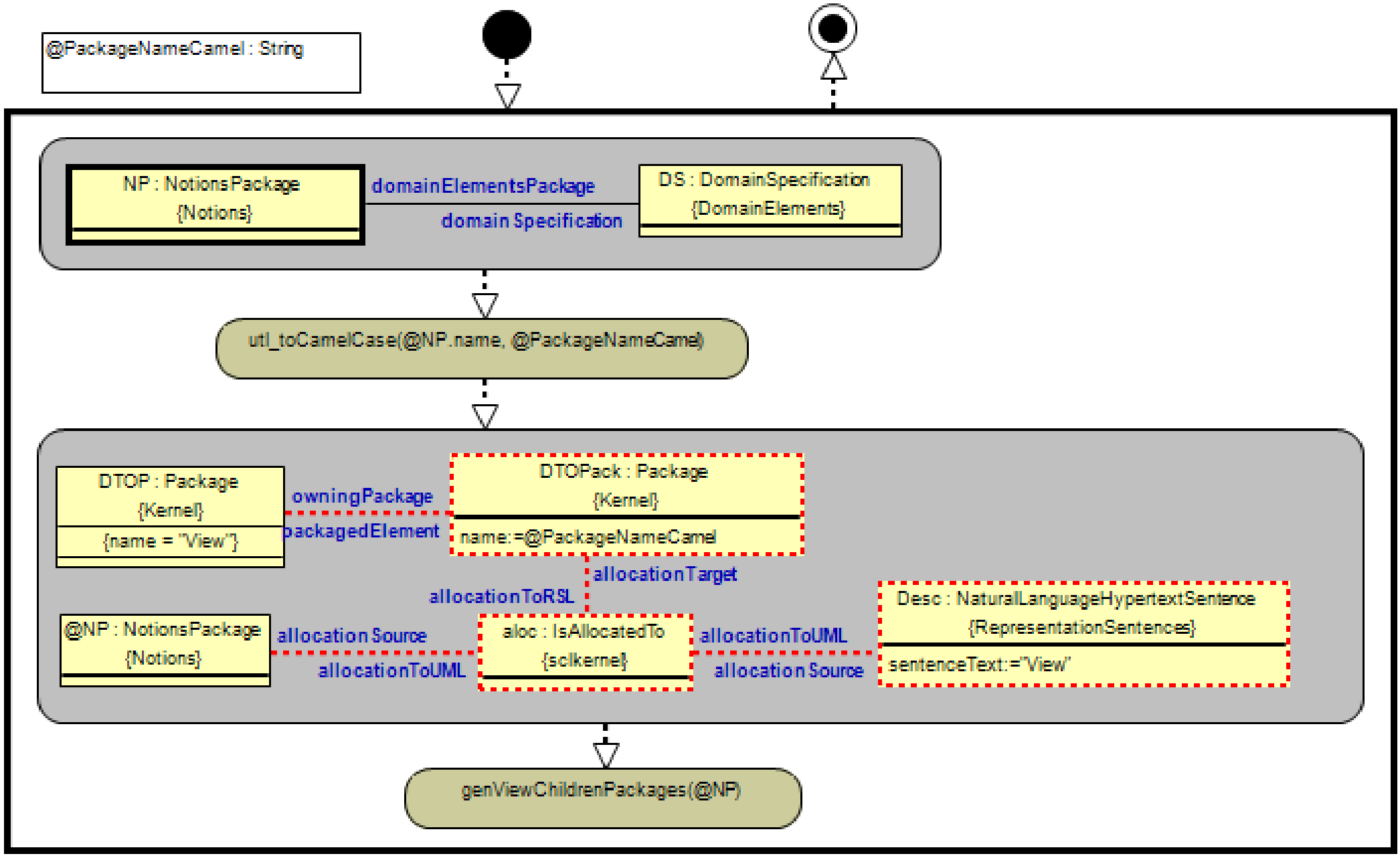


UML design model
+ Java method bodies

MOLA

- MOdel transformation Language
- Graphical language
- Based on metamodel-level pattern matching
- Transformation (program) describes how to transform one model into antoher

MOLA - example



Summary

- Combination of existing approaches:
  - Software Reverse Engineering
  - Requirements Engineering
  - Model Driven Development
- Shortens time needed for transition from legacy to modern technology system
- Validated using a real-life system

Thank You!