System testing from UML diagrams

Eliane Martins

Institute of Computer – State University of Campinas (Unicamp)

eliane@ic.unicamp.br

Harpia

- Goal:
  - Development of web applications for the Tax Department
- Goal of the testing group:
  - Model-based system testing
  - Performability testing
    - Performance
    - Avaliability

Fault injection

Harpia

- Goal:
  - Development of web applications for the Tax Department
- Goal of the testing group:
  - **Model-based system testing**
  - Performability testing
    - Performance
    - Avaliability

Fault injection

Motivation

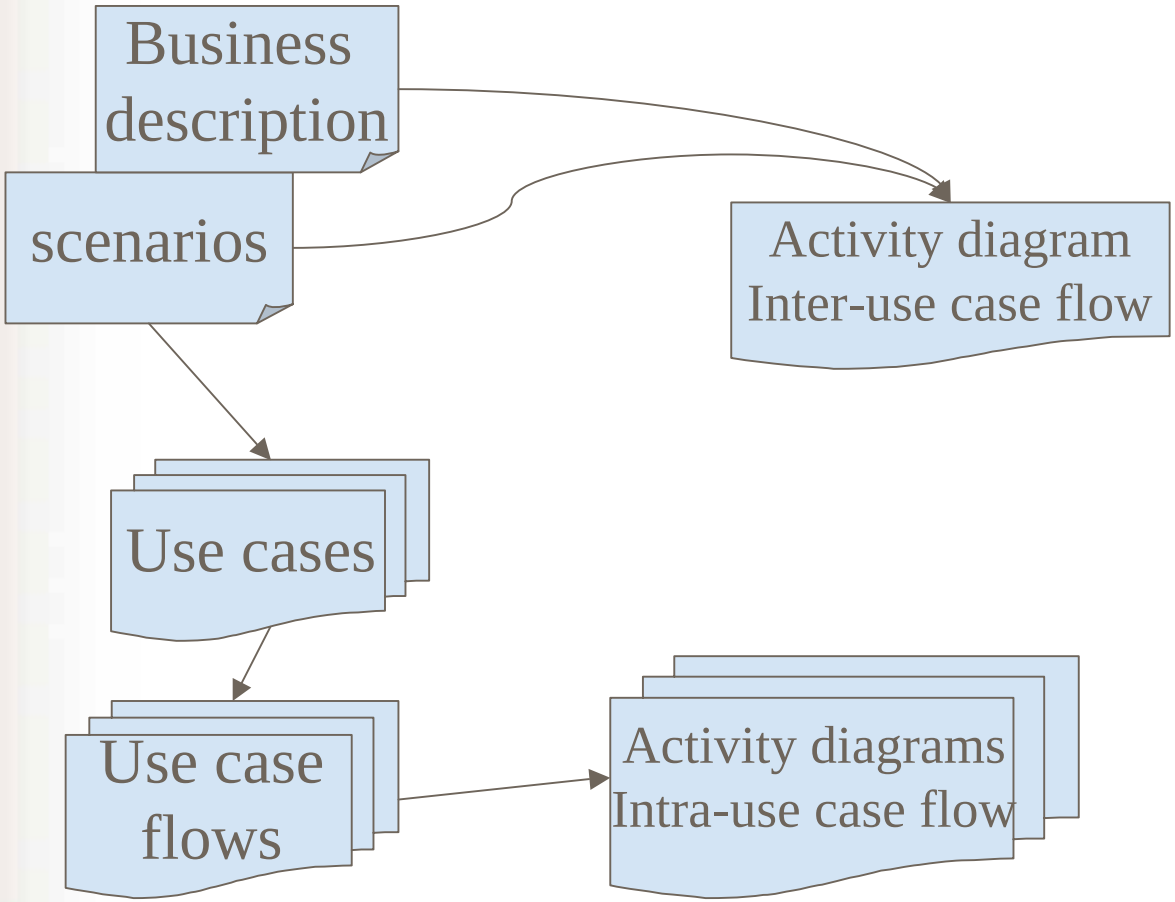
- In Brazil: system developers commonly use UML notations for specification and design
- Scenarios are popular as part of requirements specification
  - Scenarios describe how users and system interact to provide some service
  - Many scenarios are needed to describe a system

☞ How to generate test cases based on these scenarios?

Commonly used approaches

- Since many scenarios are needed to describe a system → how to combine these scenarios?
- Approaches :
  - Sequence Diagram or Message Sequence Chart
  - Finite State Machines
  - Combination: Activity Diagrams and Sequence Diagrams
- Our approach: hierarchy of Activity Diagrams

Modelling the system

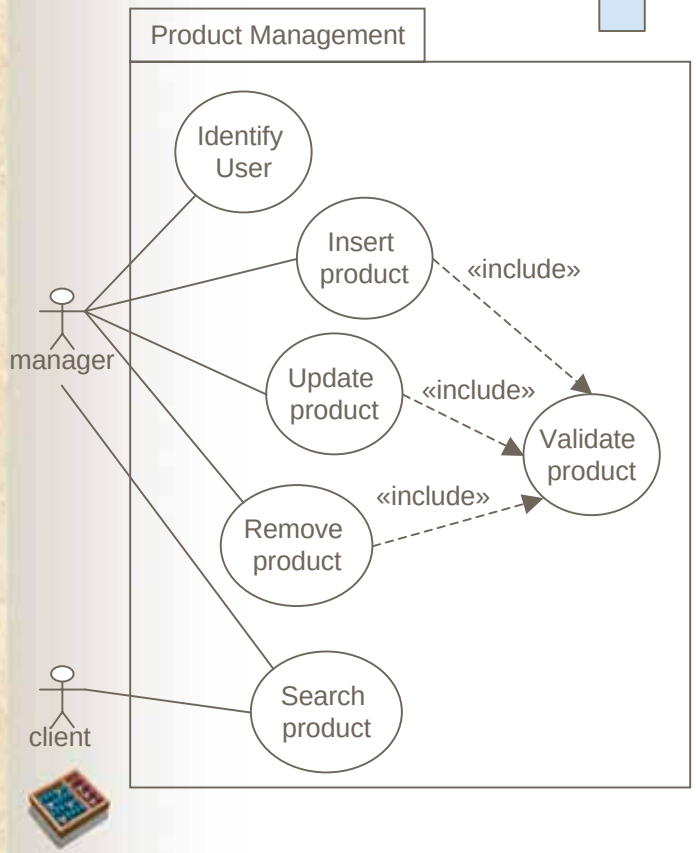


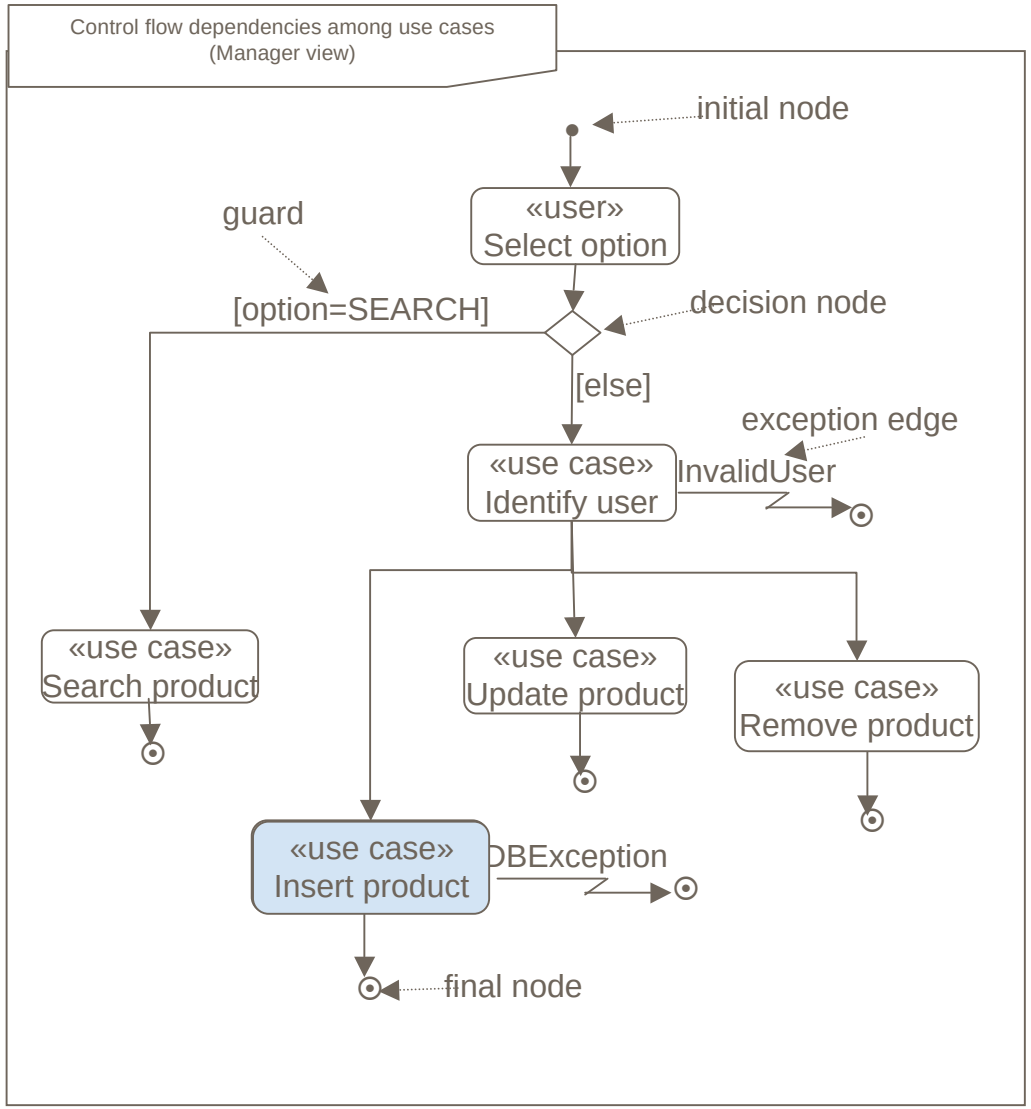
Use case description

Use case name	Insert Product
Description	The goal of this use case is to provide a solution for the creation of new products in the data base.
Actors	Manager
Pre-conditions	The user must be logged in and selected the “Insertion” option.
Invariants	None
Main Flow	**P1. Check user identification** If user is not valid then throw E1. **P2. Get product information** The system shows a form to be filled with the information concerning the product. **P3. Validate product** Includes use case Validate Product. If the product is not valid execute alternate flow A1. **P4. Confirm product insertion** The system exhibits a message asking the user to confirm the product insertion into the database. If the user confirms, then go to P7 else the system cancels the insertion. **P7. Insert new productin the database** If problems with the insertion in the database then generate exception E2 else the use case terminates successfully.
Alternate Flow	**A1. Mensagens de Alerta do Produto** The system exhibits error messages, showing the fields in the form that are wrong. Go to P2 to allow the user to correct the errors.
Exception Flow	**E1. User not authentified** a.An exception corresponding to the error is thrown. **E2. Exception generated by the DB** a.The exception is captured by na exception handler, that must guarantee the data is inserted.
Post-conditions	The product is inserted into the database or the database remains unchanged.

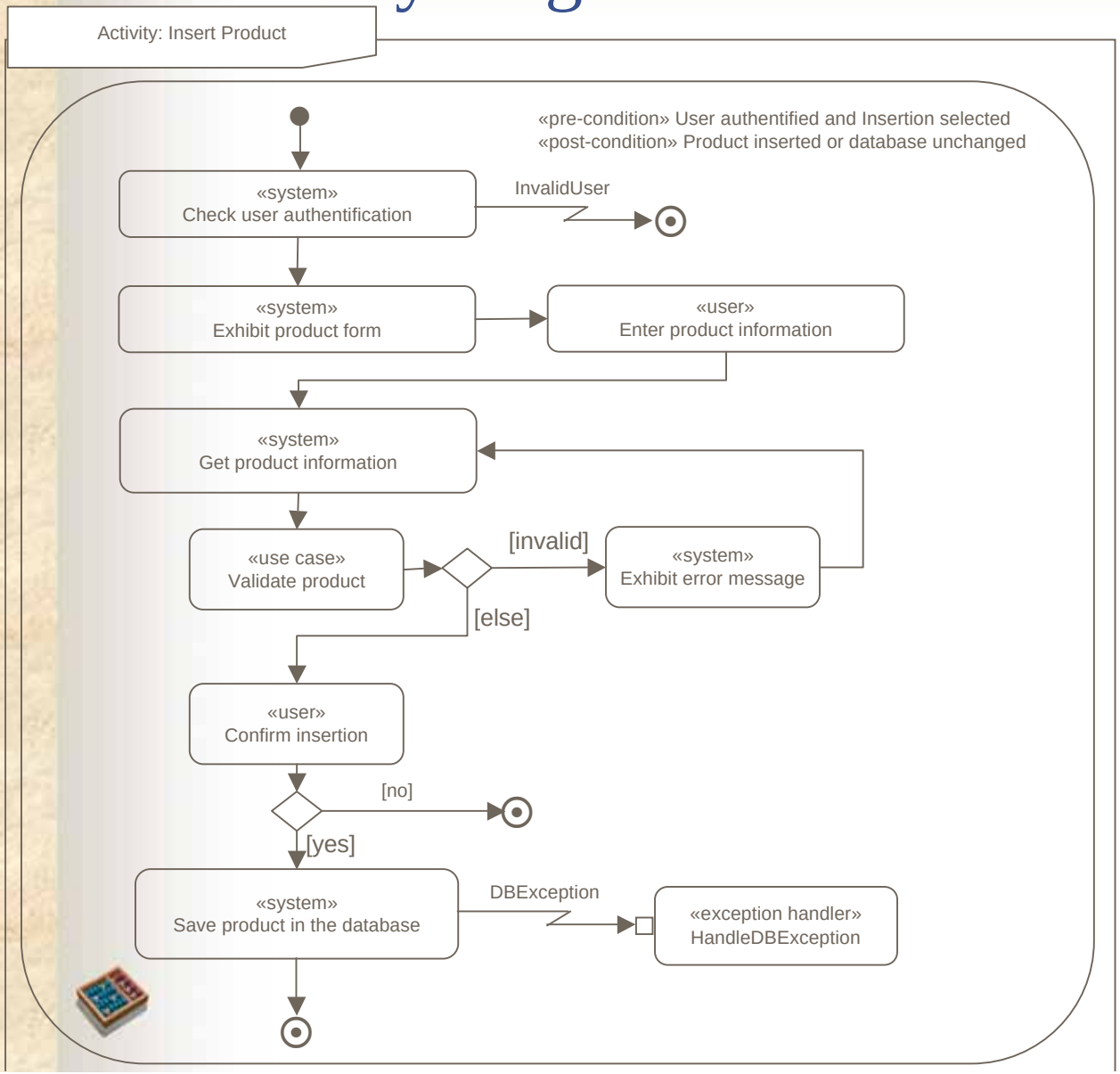
Activity diagram – Inter use cases

Scenarios
+
Business
Description



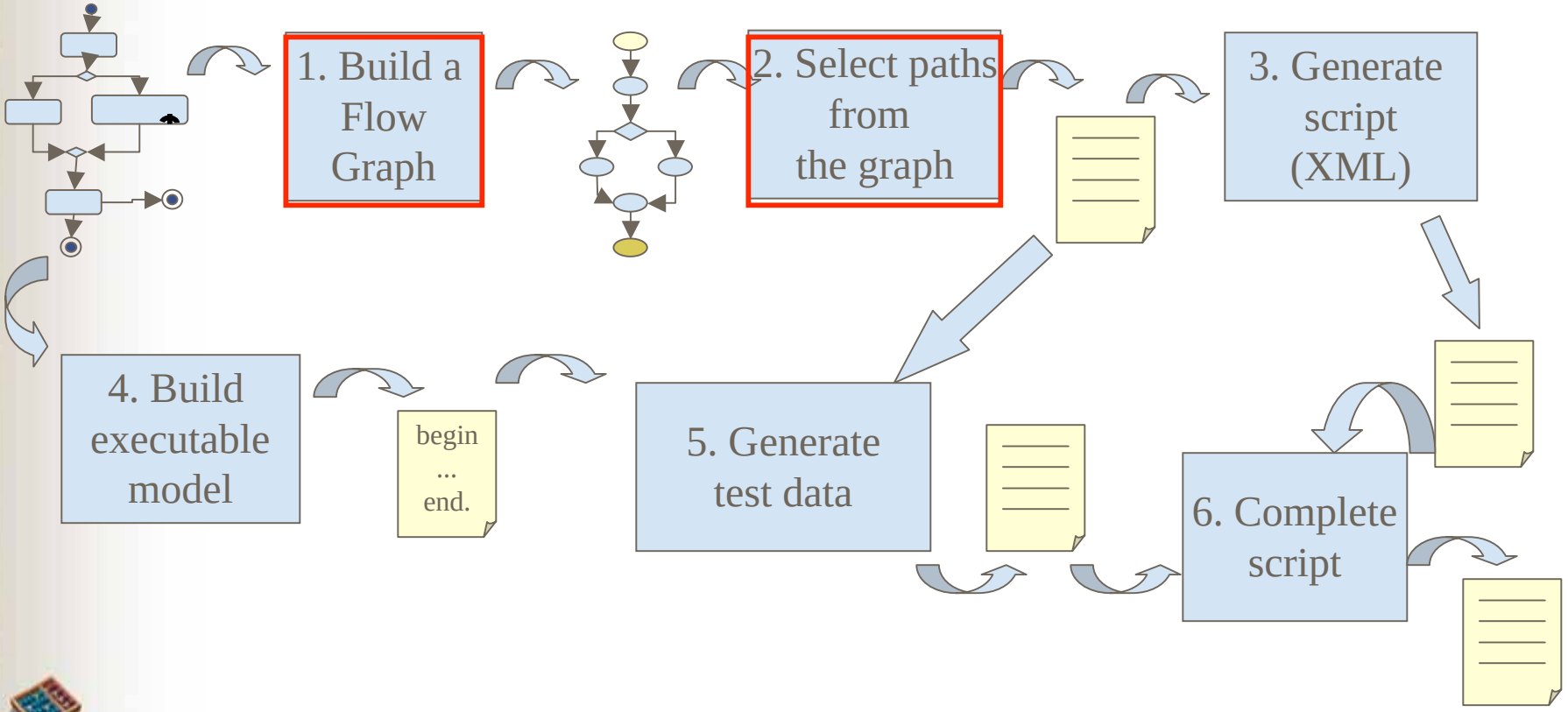


Activity diagrams – intra use cases

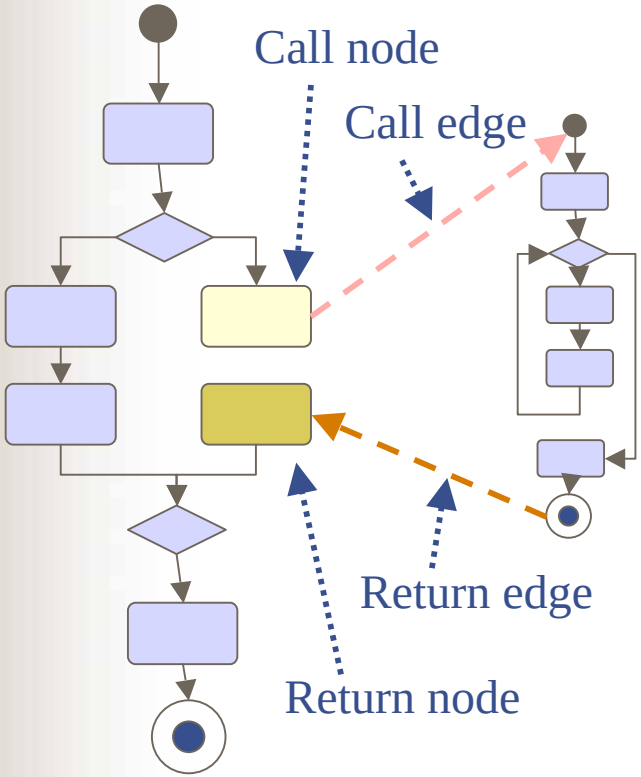


Test Case Generation

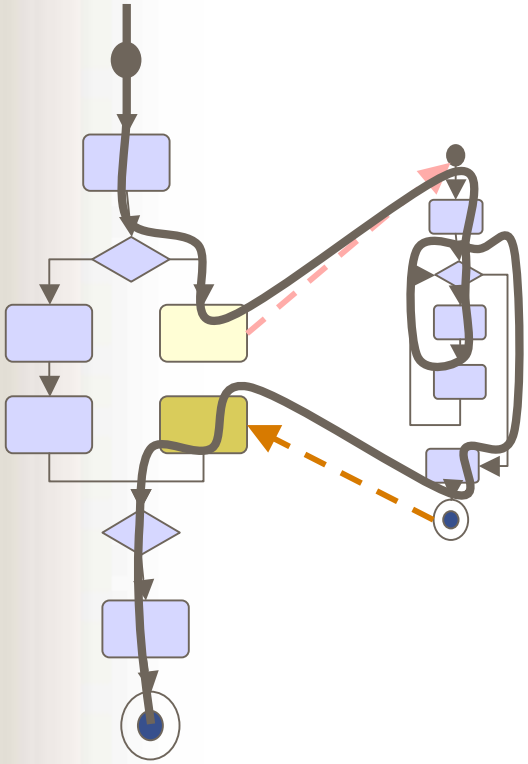
UML Activity Diagram



The test model



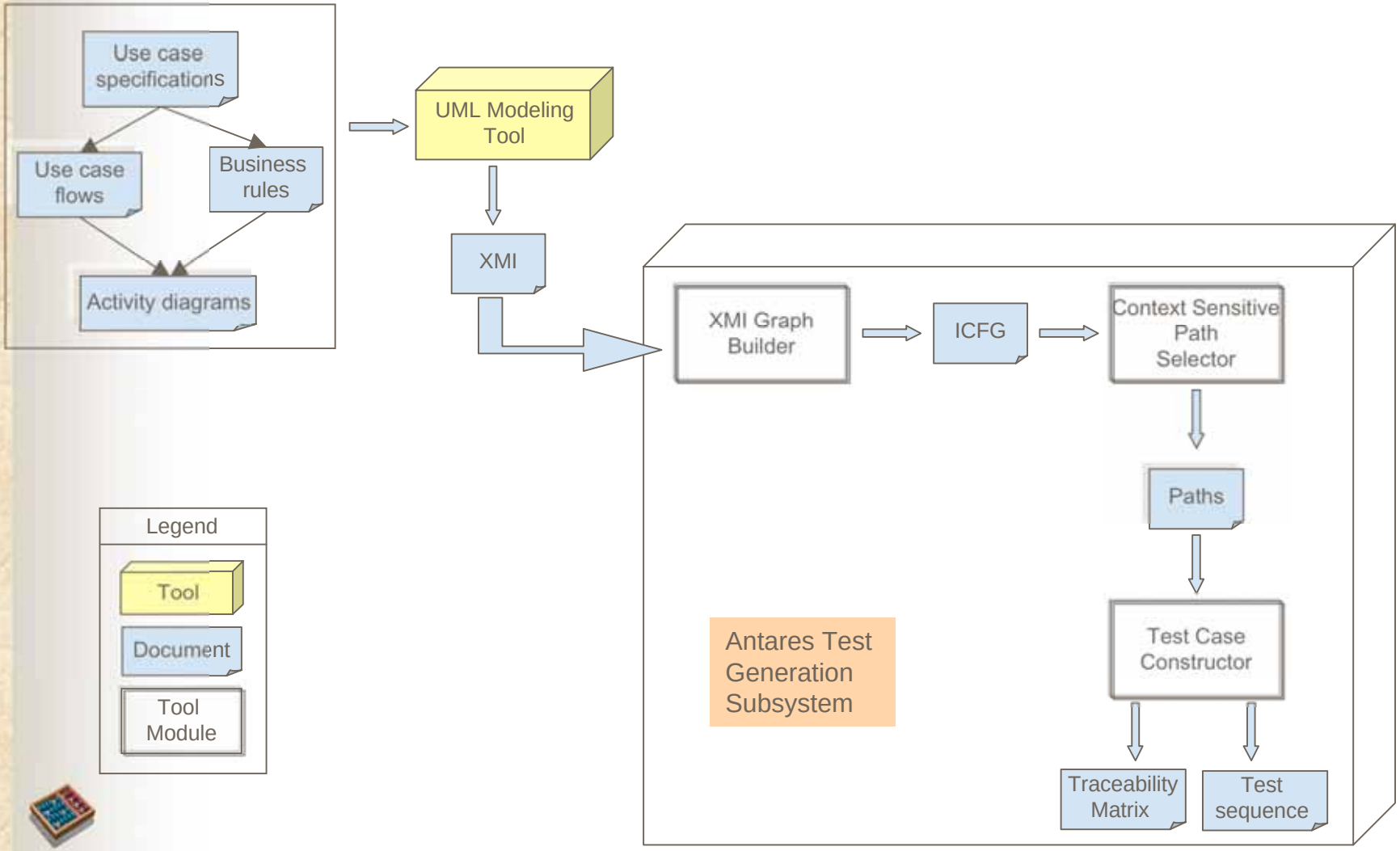
Path-oriented test selection



- Problems:
  - How to select paths?
    - Control flow based criteria (e.g. all edges, all nodes)
  - How to select realizable paths?
    - Various call-return in a path
    - **Realizable** path: each call edge is matched with its return edge

      ⇩
    - Context sensitive search for a path
  - How to deal with loops to avoid infinite number of paths?
    - Limit number of repetitions
    - Loop testing

The tools


Use case specifications
Use case flows
Business rules
Activity diagrams
UML Modeling Tool
XMI
XMI Graph Builder
ICFG
Context Sensitive Path Selector
Paths
Test Case Constructor
Traceability Matrix
Test sequence
Antares Test Generation Subsystem
Legend
Tool
Document
Tool Module

Some results

Test case design		Test case execution	
# UC	**27**	# executed test cases	**76**
# ICFG edges	**441**	# fault revealing test cases	**37**
# ICFG nodes	**530**	# Failures	**131**
# test cases (TC)	**142**	Average TC execution time	**40 min**

Conclusions

- On-going work
  - Regression testing selection based on Activity Diagram
  - Testing process still in use → more measurements are being performed
  - Systematic creation of surrogates (or proxies) for exception handling testing (to be obtained from test cases)
- Future work:
  - Implementation of other test criteria
  - Model validation (e.g. simulation)
  - Data flow and test data generation
  - Considering concurrency
    - various actors using the system

Thanks!