

# Workshop on Automation of Software Testing AST 2009

## Charette Session`s Challenge: Model (Test) Generation from Requirements

Team Members

Guilherme Horta Travassos, COPPE/UFRJ

Leif Singer, LUH, Germany

Heiko Stallbaum, Uni DuE, Germanys

David Connolly, SToRC, DkIT, Ireland

Nicky Williams, CEA List, France

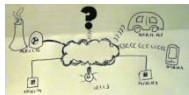
# Software Development Perspectives

## REQUIREMENTS

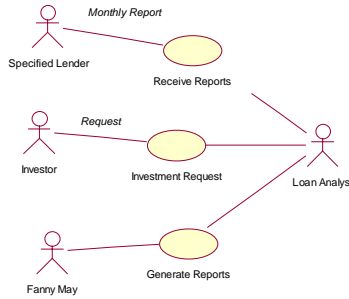
Loan-Arranger Requirements Specification – Jan. 8, 1999

### Background

Banks generate income in many ways, often by borrowing money from their depositors at a low interest rate, and then lending that same money at a higher interest rate in the form of bank loans. However, property loans, such as mortgages, typically have terms of 15, 25 or even 30 years. For example, suppose that you purchase a \$150,000 house with a \$50,000 down payment and borrow a \$100,000 mortgage from National Bank for thirty years at 5% interest. That means that National Bank gives you \$100,000 to pay the balance on your house, and you pay National Bank back at a rate of 5% per year over a period of thirty years. You must pay back both principal and interest. That is, the initial principal, \$100,000, is paid back in 360 installments (once a month for 30 years), with interest on the unpaid balance. In this case the monthly payment is \$536.82. Although the income from interest on these loans is lucrative, the loans tie up money for a long time, preventing the banks from using their money for other transactions. Consequently, the banks often sell their loans to consolidating organizations such as Fannie Mae and Freddie Mac, taking less long-term profit in exchange for freeing the capital for use in other ways.



AD-HOC

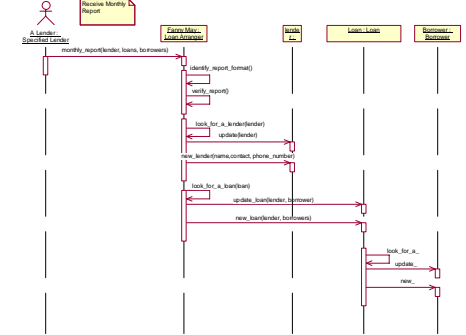
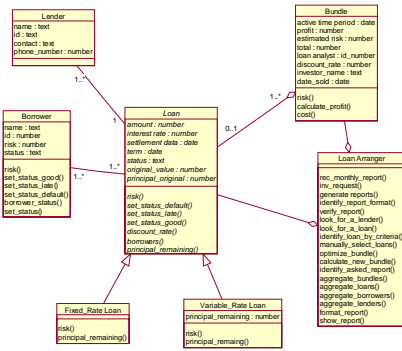
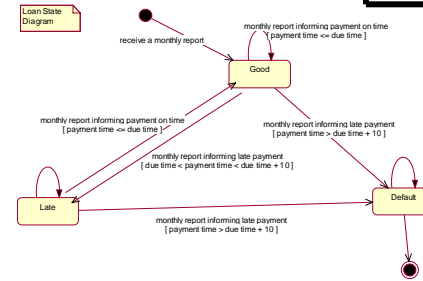


Scalene Triangle:

$$\{<x,y,z>: (x \neq y) \wedge (x \neq z) \wedge (y \neq z)\}$$

FORMAL

| TEST CASES  |   |   |   |
|-------------|---|---|---|
| CLASS       | X | Y | Z |
| Scalene     | 3 | 4 | 5 |
| Isosceles   | 5 | 5 | 8 |
| Isosceles   | 3 | 4 | 3 |
| Isosceles   | 4 | 7 | 7 |
| Equilateral | 2 | 2 | 2 |
| No-triangle | 1 | 2 | 3 |
| No-triangle | 5 | 1 | 4 |
| No-triangle | 3 | 5 | 2 |



(Possible) GENERATED MODELS

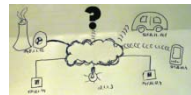
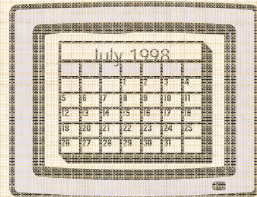
# Software Development Perspectives

## REQUIREMENTS

Loan-Arranger Requirements Specification -- Jan. 8, 1999

### Background

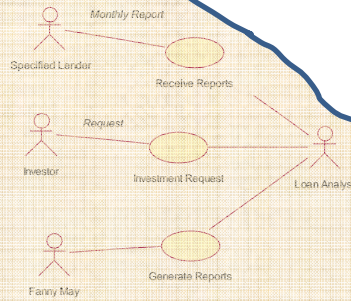
Banks generate income in many ways, often by borrowing money from their depositors at a low interest rate, and then lending that same money at a higher interest rate in the form of bank loans. However, property loans, such as mortgages, typically have terms of 15, 25 or even 30 years. For example, suppose that you purchase a \$150,000 house with a \$50,000 down payment and borrow a \$100,000 mortgage from National Bank for thirty years at 5% interest. That means that National Bank gives you \$100,000 to pay the balance on your house, and you pay National Bank back at a rate of 5% per year over a period of thirty years. You must pay back both principal and interest. That is, the initial principal, \$100,000, is paid back in 360 installments (once a month for 30 years), with interest on the unpaid balance. In this case the monthly payment is \$536.82. Although the income from interest on these loans is lucrative, the loans tie up money for a long time, preventing the banks from using their money for other transactions. Consequently, the banks often sell their loans to consolidating organizations such as Fannie Mae and Freddie Mac, taking less long-term profit in exchange for freeing the capital for use in other ways.



Tacit requirements



AD-HOC

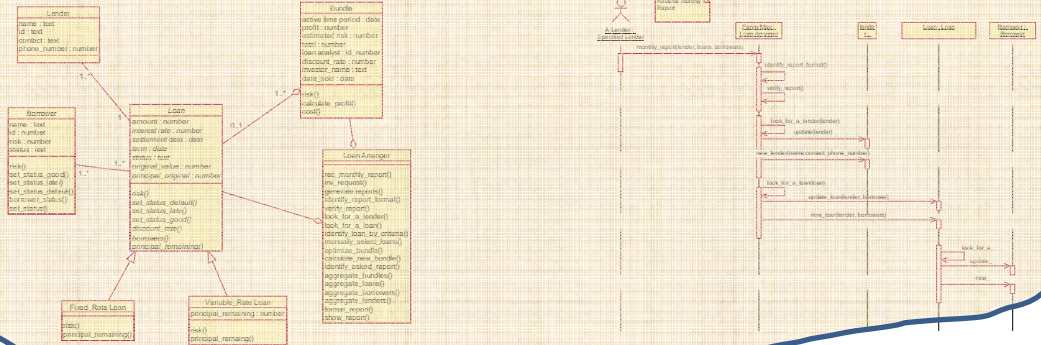
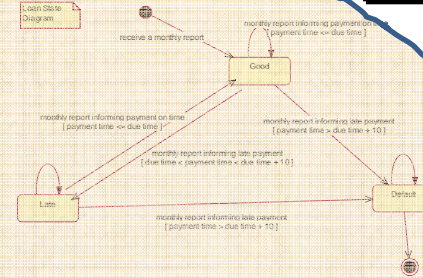
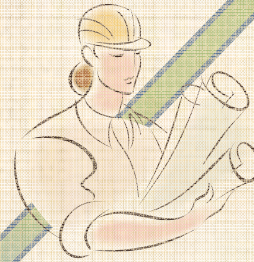


Scalene Triangle:

$$\{<x,y,z>: (x \neq y) \wedge (x \neq z) \wedge (y \neq z)\}$$

FORMAL

| TEST CASES  |   |   |   |
|-------------|---|---|---|
| CLASS       | X | Y | Z |
| Scalene     | 3 | 4 | 5 |
| Isosceles   | 5 | 5 | 8 |
| Isosceles   | 3 | 4 | 3 |
| Isosceles   | 4 | 7 | 7 |
| Equilateral | 2 | 2 | 2 |
| No-triangle | 1 | 2 | 3 |
| No-triangle | 5 | 1 | 4 |
| No-triangle | 3 | 5 | 2 |



(Possible) GENERATED MODELS

# Model (Test) Generation from Requirements

- Team`s assumptions
  - Quality of Requirements
    - Testability
    - Readability
    - Validity
    - Completeness
    - Precision
    - Traceability
    - ...
  - Models generated for testing shall be validated.

# Model (Test) Generation from Requirements

- Experience based Suggestions (perhaps individual's viewpoints...)
  - Requirements shall be reviewed/inspected
    - To provide heuristics to guide the review
  - Requirements for requirements
    - Some specific characteristics should be identified and explored to support model generation for testing
    - Semi-automated tasks must be identified (depending on the type of the system)
  - Each type of testing (i.e. performance, acceptance,...) requires different set of requirements
    - Requirements classification schemas could help to generate models/information for such testing types
    - Based on the classes of requirements developers could identify the type of testing and models needed considering the requirements set for a particular system

# Model (Test) Generation from Requirements

- Experience based Suggestions (perhaps individual's viewpoints...)
  - Requirements should be tagged (accordingly a pre-defined classification by the organization)
    - Doing so, that could be possible to develop a tool to scan the requirements specification classifying requirements and advising the models and testing activities
    - Besides, that could be possible to find out a way to combine an automated classification and organization of requirements aiming at the generation of possible models

# What do we need?

- Classes of Requirements
  - Tool Support for Classification of Requirements
  - Mapping of Test Methodologies to Requirements Classes
- Test Method needs certain Models
  - What Qualities do we need in Requirements to derive exactly these models?

# Next Steps

- What kind of test methods do we want to support?
- What kind of models are required for each of these methods?
- What are the elements of these models?
- What are the qualities needed from Requirements to derive each of these elements?

## Next Steps (cont.)

- How can this approach be integrated into the *system* development process?
- What kind of tool support could be made available for the derivation of models / refinement of requirements?
- Evaluation of Approaches / Proofs of Concept
  - To get industrial sponsorship and research funding