

Behaviour-Driven Development of Foundational UML Components

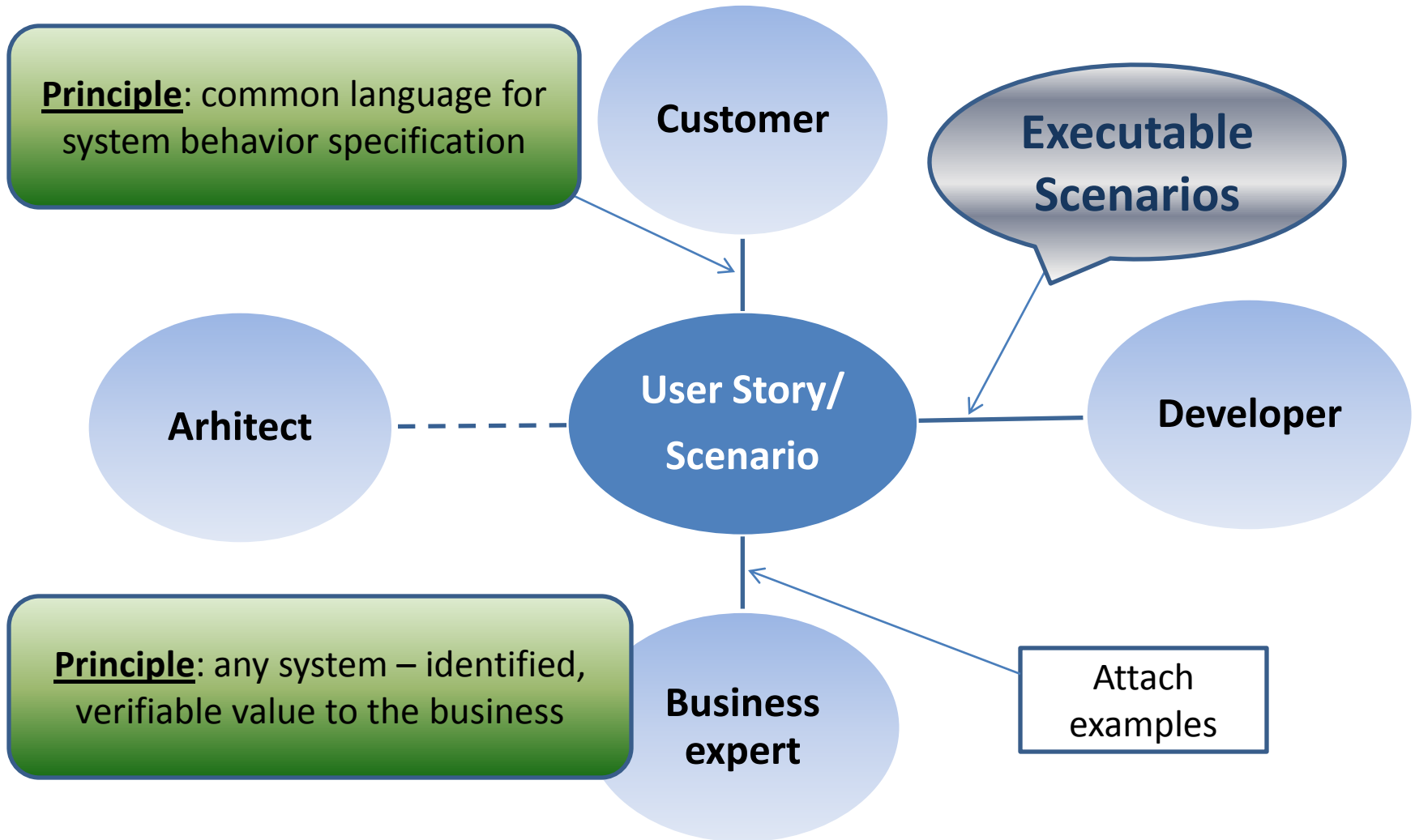
I. Lazar, S. Motogna. B. Parv

Department of Computer Science

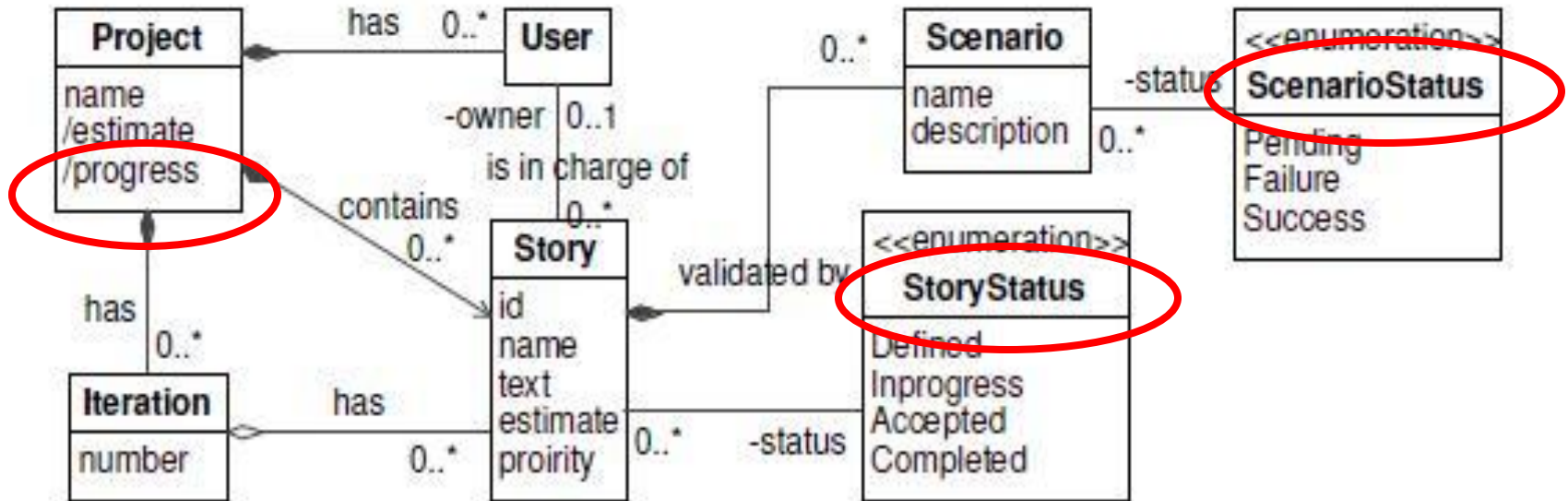
Babes-Bolyai University

Cluj-Napoca, Romania


Behavior Driven Development (BDD)



Concepts in BDD



BDD activities as user stories

| | |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Requirements analysis | US1. Users add new stories to a project  US2. Users add new scenarios to a story |
| Project Planning | US3. Users add new iterations to a project US4. Users allocate stories to iterations US5. Developers accept stories |
| Development iteration | US6. Developers implement stories US7. Update iteration status after scenario execution US8. Update project progress after scenario execution |
| Verifiable progress | US9. Users can obtain iteration status reports US10. Users can obtain project status reports |

Agile project planning

**<<Story>>
Add new
stories to
a project
{id = "US1" }**

<<Scenario>>

Verify story initialization

<<Scenario>>

**Estimate a project containing two
estimated stories**

<<Scenario>>

**Estimate a project containing an
estimated story**

Scenario description

(a) Verify story initialization

- **given** "a project"
- **when** "a new story is created"
- **then** "story status should be defined" and "project should include story"

(b) Estimate project =
2 estimated stories

- **given** "a project" and "two new stories"
- **when** "both stories have estimates"
- **then** "project estimate = Σ story estimates"

(c) Estimate project =
estimated story

- **given** "a project" and "a new story"
- **when** "the story is estimated"
- **then** "the project estimate = story estimate"

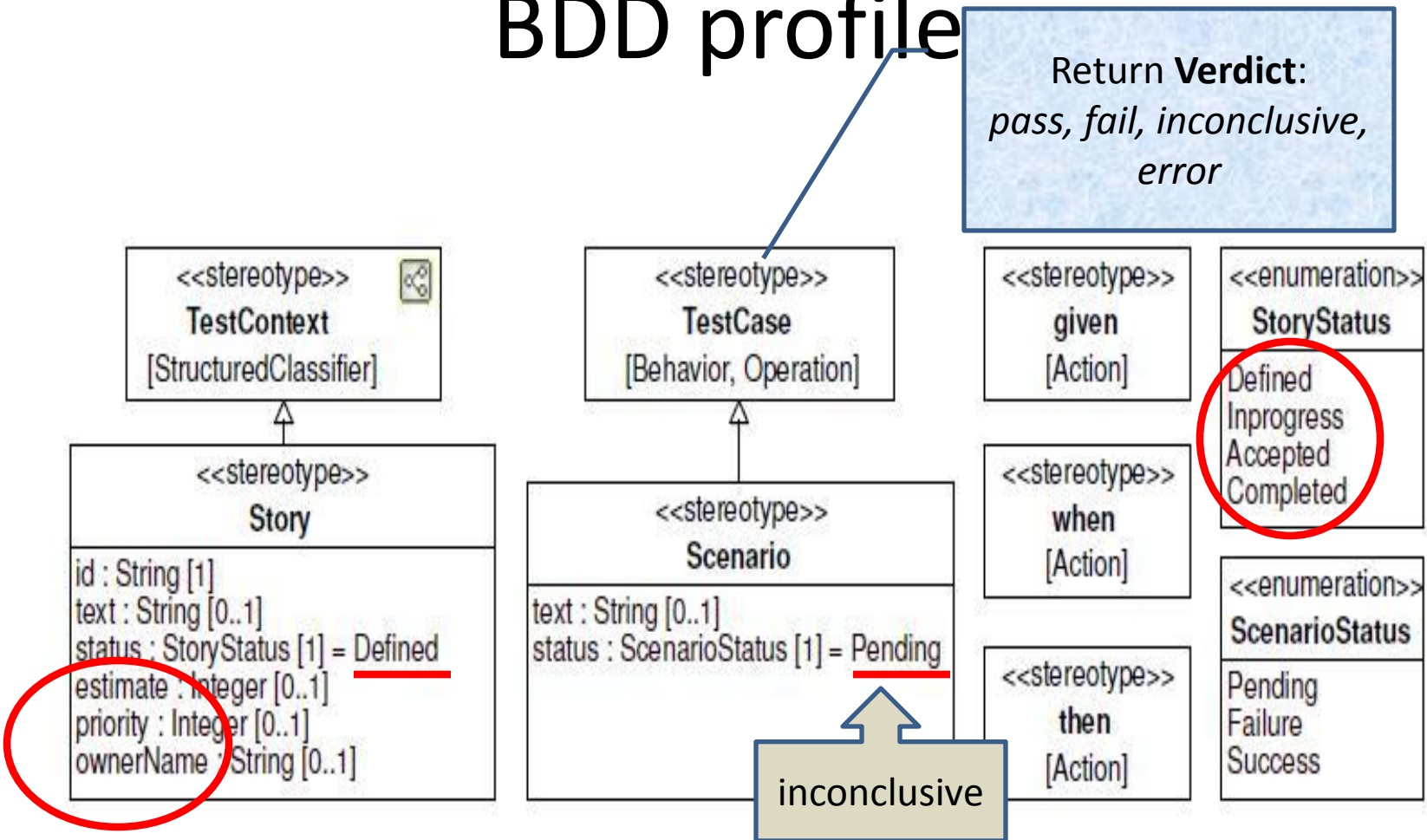
Research problem

- BDD approach for developing executable fUML components
- fUML defines implementation-oriented abstractions - refer to Platform Independent Models (PIM).
- Refer only structural and behavioral constructs defined by the fUML specification
- tailored to the algorithmic and data-intensive types of programs
- concrete textual syntax is needed - enforces a certain way of constructing models
- generated code is meant to be complete

Solution – consists of:

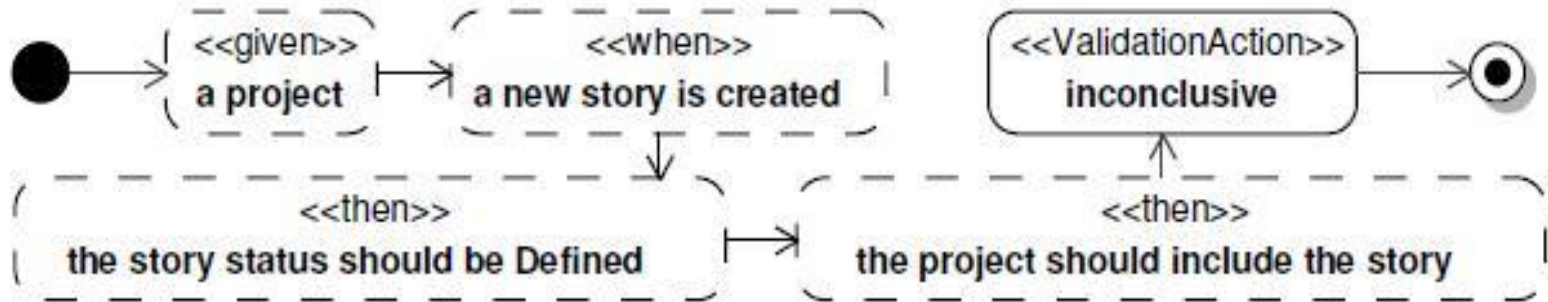
- UML profile
- Library \supseteq fUML activities – equality & inclusion
- Activities – executable scenarios
 - Concrete syntax for BDD scenarios
- bUML – BDD tool for fUML models

BDD profile



Scenario description textual & graphical representation

- **given** "a project"
- **when** "a new story is created"
- **then** "story status should be defined" and "project should include story"

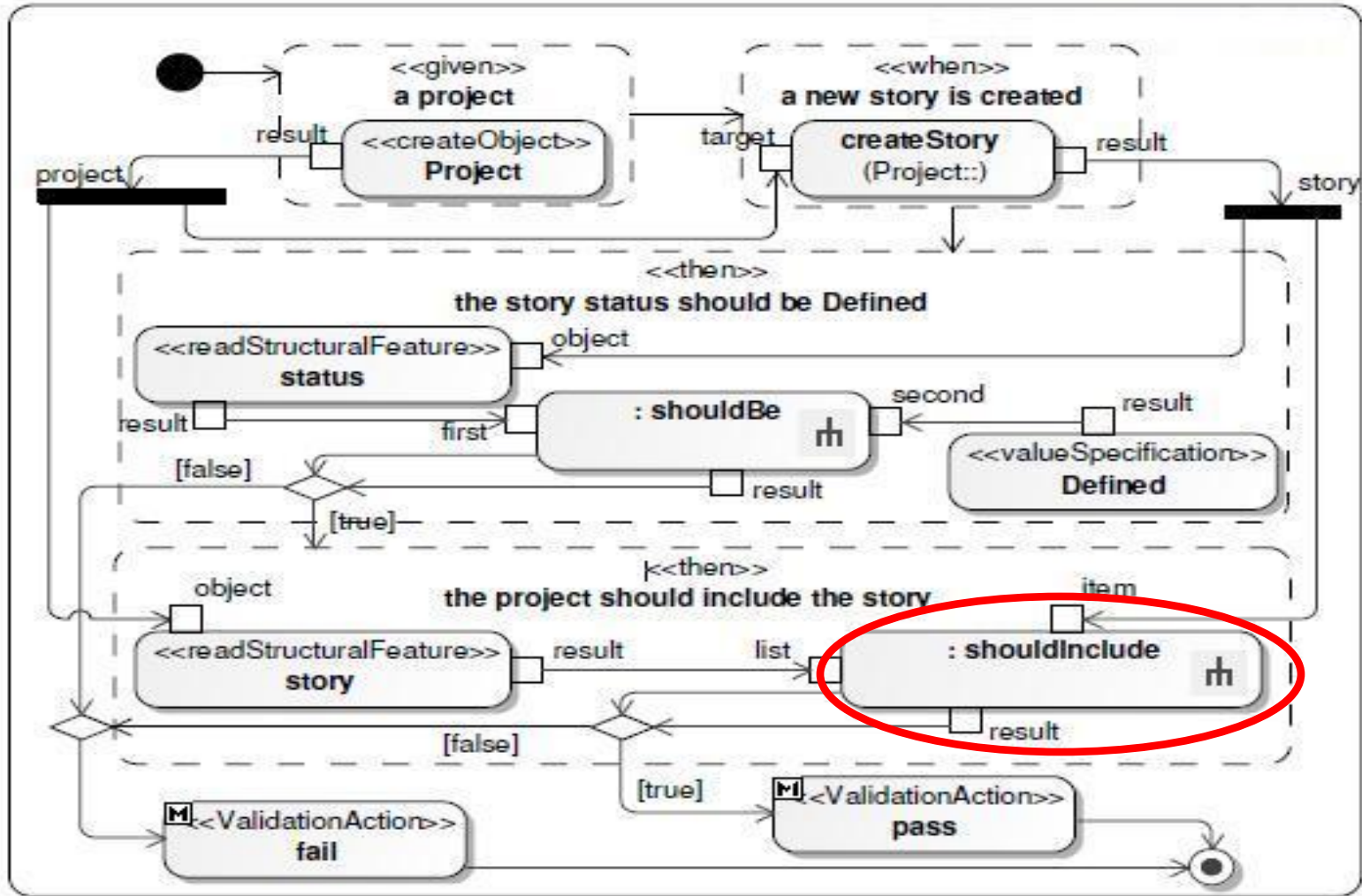


Scenario implementation

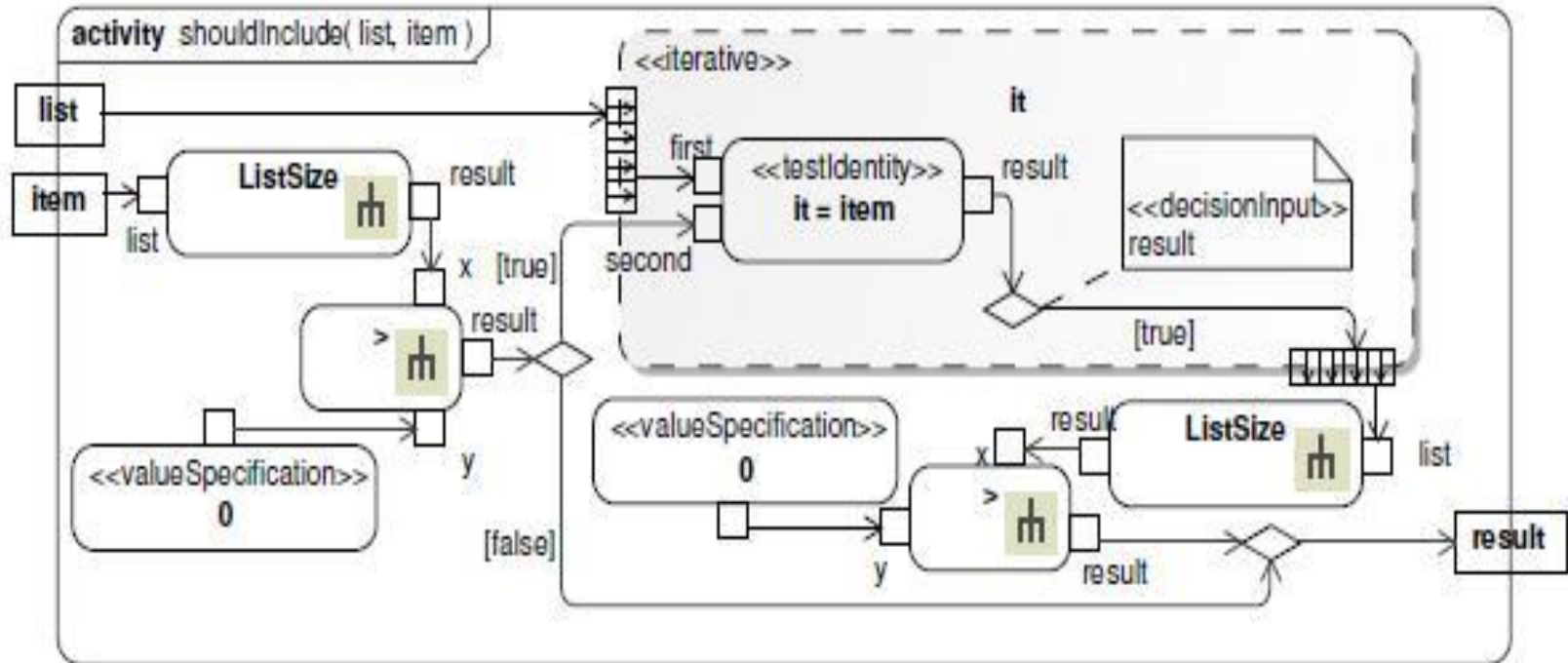
Difficulties:

- ❑ Implement scenario – return verdict
 - fUML primitives – low level
- ❑ Implement scenario – complex
 - how fUML define behavior of methods
- ❑ *then* part of scenario
 - Testing equality and inclusions

Scenario implementation



shouldInclude(list[], item[0..1]): Boolean*



bUML Tool

| | |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Requirements analysis | US1. Users add new stories to a project US2. Users add new scenarios to a story |
| Project Planning | US3. Users add new iterations to a project US4. Users allocate stories to iterations US5. Developers accept stories |
| Development iteration | US6. Developers implement stories US7. Update iteration status after scenario execution US8. Update project progress after scenario execution |
| Verifiable progress | US9. Users can obtain iteration status reports US10. Users can obtain project status reports |

Verifiable progress – remarks:

- After execution, models automatically update:
 - Scenario status
 - User story status.
- Different from platform dependent approaches
- Executed on models

Related work

- no other existing works combine BDD and MDD approaches
- close to easyb <http://www.easyb.org/> (2008)
- Business Motivation Model (BMM) and SysML requirements - consistency with the UML testing profile.
- Executable Acceptance Test-Driven Development (EATDD) [Park, Maurer] (2008)

Thank you