



Automation of Software Test (AST'09)
May 18, Vancouver, Canada

Towards an automated testing framework to manage variability using the UML Testing Profile

Beatriz Pérez Lamancha

*Software Testing Centre
(CES)*

*School of Engineering
University of the Republic
Montevideo, Uruguay
bperez@fing.edu.uy*

Macario Polo, Mario Piattini

*Alarcos Research Group
Information Systems and
Technologies Department
University of Castilla-La Mancha
Ciudad Real, Spain
{macario.polo,mario.piattini}@uclm.es*



Outline

- Introduction
 - Software Product Line
 - Orthogonal Variability Model (OVM)
 - UML Testing Profile
- Proposal: automated testing in SPL
 - Variability in test case behaviour
 - Variability in test architecture
- Conclusions and future work



Software Product Lines

- *A software product line (SPL) is a set of software intensive systems sharing a common, managed set of features which satisfy the specific needs of a particular market segment or mission and which are developed from a common set of core assets in a prescribed way*

(Clements and Northrop, 2002)



Commonality and Variability

- Commonalities: are assumptions that are true for each member of the SPL
- Variabilities: are assumptions on how family members differ
 - A variation point locates a variability and its variants.
 - Each variant is one way to realize a particular variability in a specific way.
- Variability management plays a central role in software product lines



Example - Lottery SPL

- Manages the bets and payments for different lottery-type games.
- There are many versions of lotteries:
 - Lotto: player chooses six numbers from 1 to 49

lotto

£1.00 per line

Number of weeks: Please select

Choose draws: Please select

Choose numbers from 1 - 49

Lucky Dip

Clear

Add more lines

Play



Example - Lottery SPL

- Manages the bets and payments for different lottery-type games.
- There are many versions of lotteries:
 - Lotto: player chooses six numbers from 1 to 49
 - Keno: the player win when get five numbers in a row
 - Instant lottery: player can remove the silver layer and see if won or not





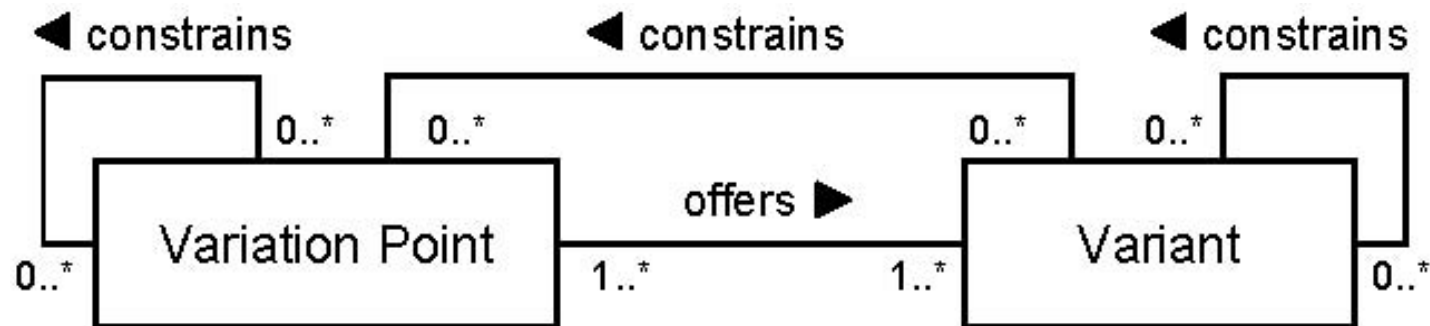
Example - Lottery SPL

- Manages the bets and payments for different lottery-type games.
- There are many versions of lotteries:
 - Lotto: player chooses six numbers from 1 to 49
 - Keno: the player win when get five numbers in a row
 - Instant lottery: player can remove the silver layer and see if won or not
- Can be played in: Web site or POS
- Method of payment can be: Cash or credit card



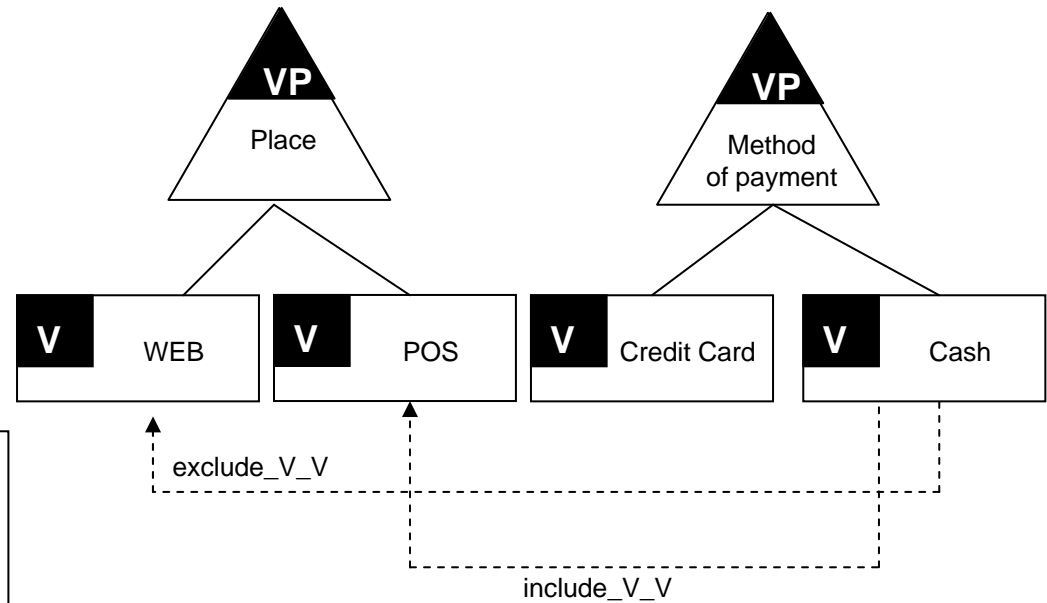
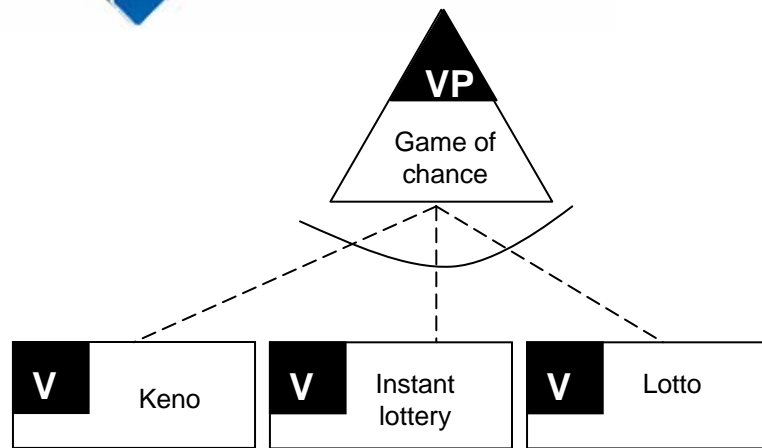
Orthogonal Variability Model (OVM) (Pohl et al., 2006)

- Defines the variability information in a separate model
- Associates the Variation Point and its variants
- Constrains the association between them
 - These constraints can be include or exclude



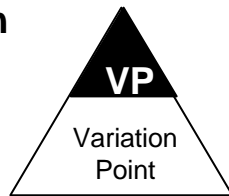


Example: Lottery SPL



OVM Graphical notation

Variation Point



Variant



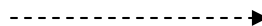
Mandatory association



Optional association



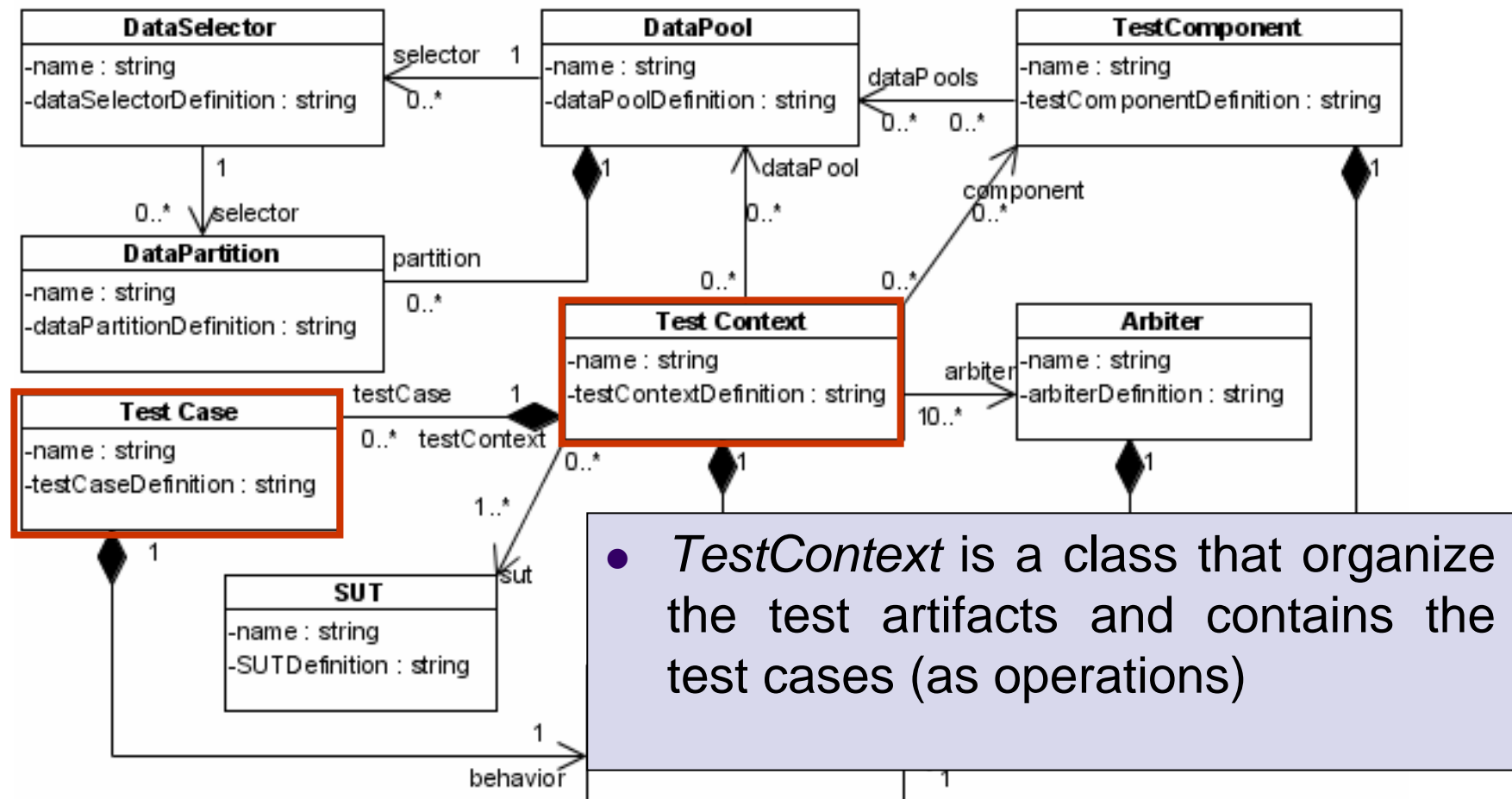
Constraint association
(include, exclude)





UML Testing Profile (UML-TP)

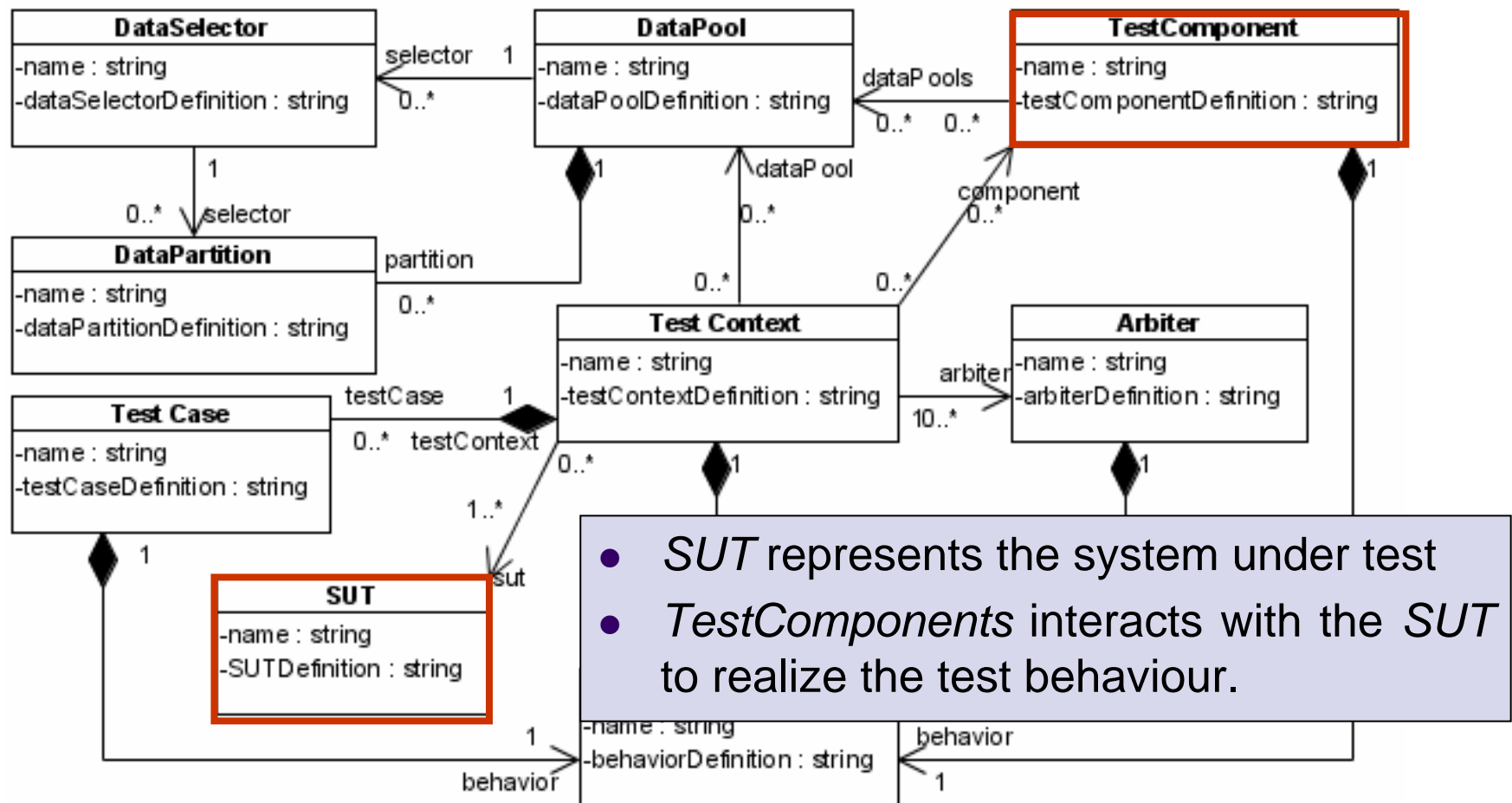
- Extends UML 2.0 with test specific concepts for testing





UML Testing Profile (UML-TP)

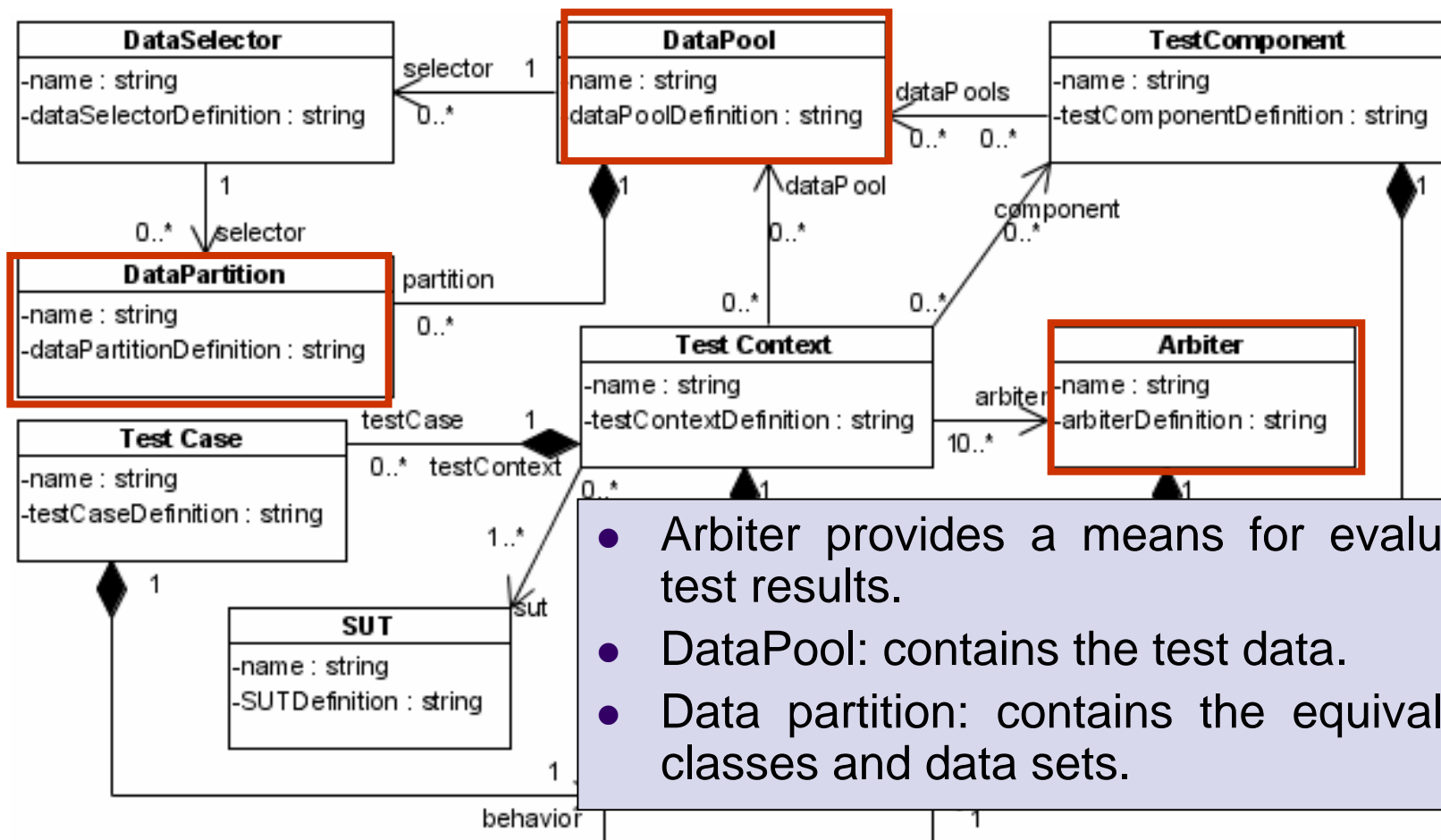
- Extends UML 2.0 with test specific concepts for testing





UML Testing Profile (UML-TP)

- Extends UML 2.0 with test specific concepts for testing

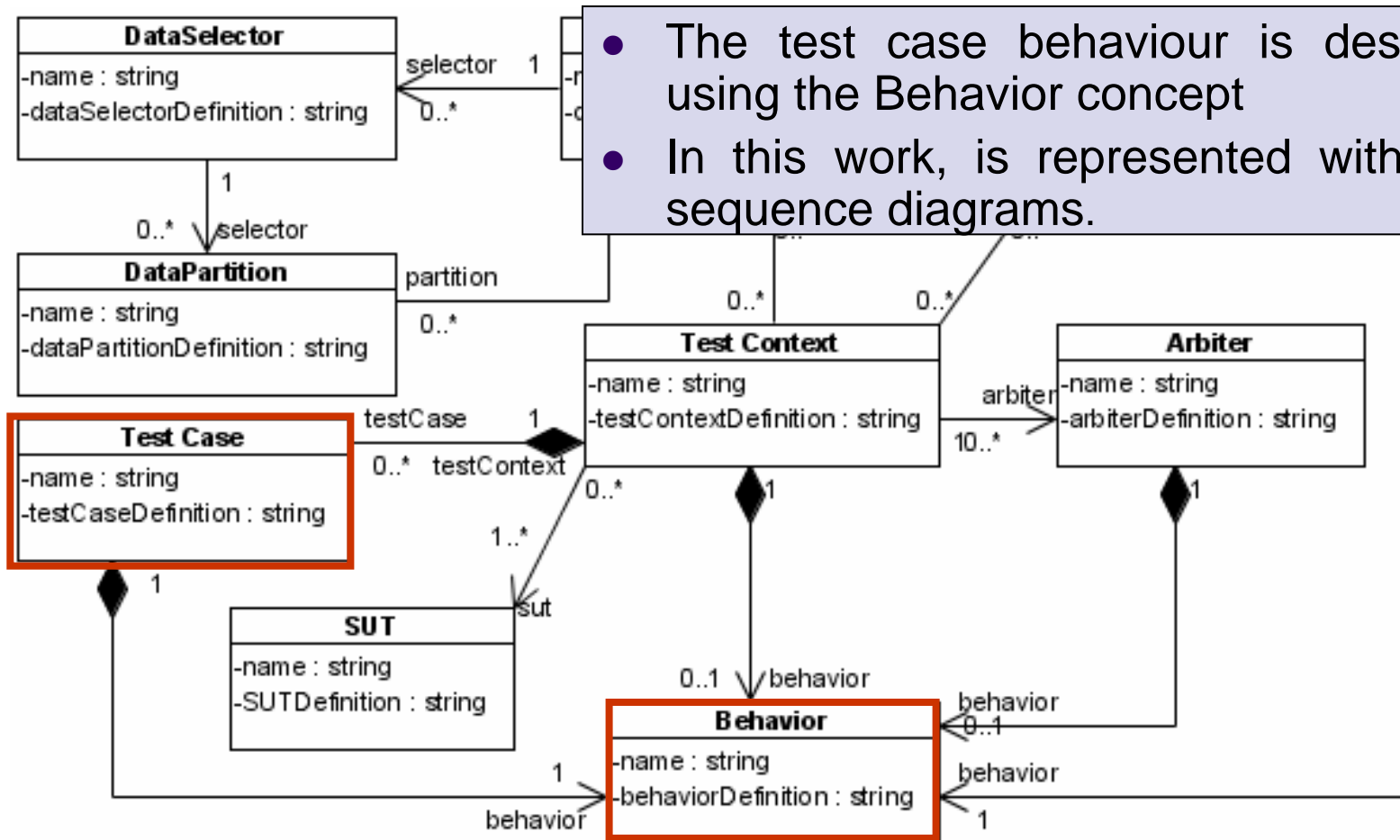


- Arbiter provides a means for evaluating test results.
- DataPool: contains the test data.
- Data partition: contains the equivalence classes and data sets.



UML Testing Profile (UML-TP)

- Extends UML 2.0 with test specific concepts for testing



- The test case behaviour is described using the Behavior concept
- In this work, is represented with UML sequence diagrams.



Outline

- Introduction
 - Software Product Line
 - Orthogonal Variability Model (OVM)
 - UML Testing Profile
- **Proposal: automated testing in SPL**
 - **Variability in test case behaviour**
 - **Variability in test architecture**
- Conclusions and future work



Our proposal

- A methodology is defined to derivation of test cases in SPL context
 - describes test cases using UML models
 - extends both UML 2.0 and the UML Testing Profile to support variability and to allow model transformation.
- This work has two main points:
 1. The inclusion of extensions in UML and UML-TP for managing variability.
 2. The definition of the test cases behaviour to manage variation points in the SPL.



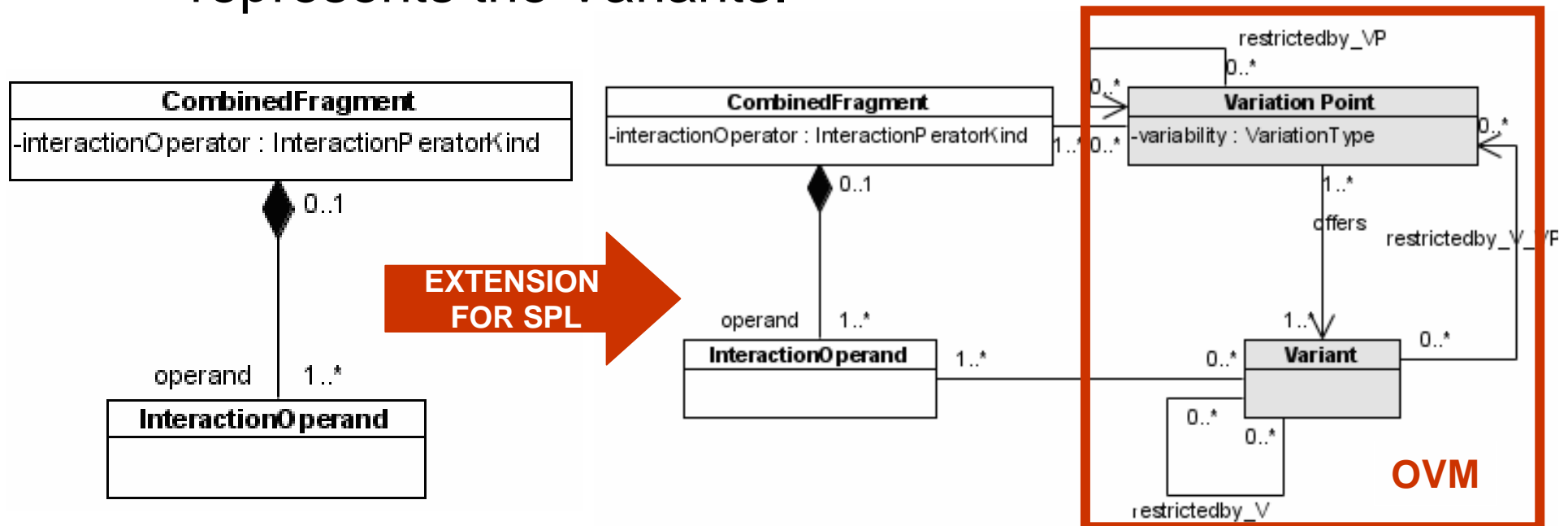
Standardized proposal

- In SPL, the best practices are intensively applied and the use of standards and tools becomes essential.
- Our proposal makes a significant contribution, since the proposal is completely framed within well-known OMG standards.
 - UML 2.0
 - UML Testing Profile
 - Query/View/Transformation (QVT)



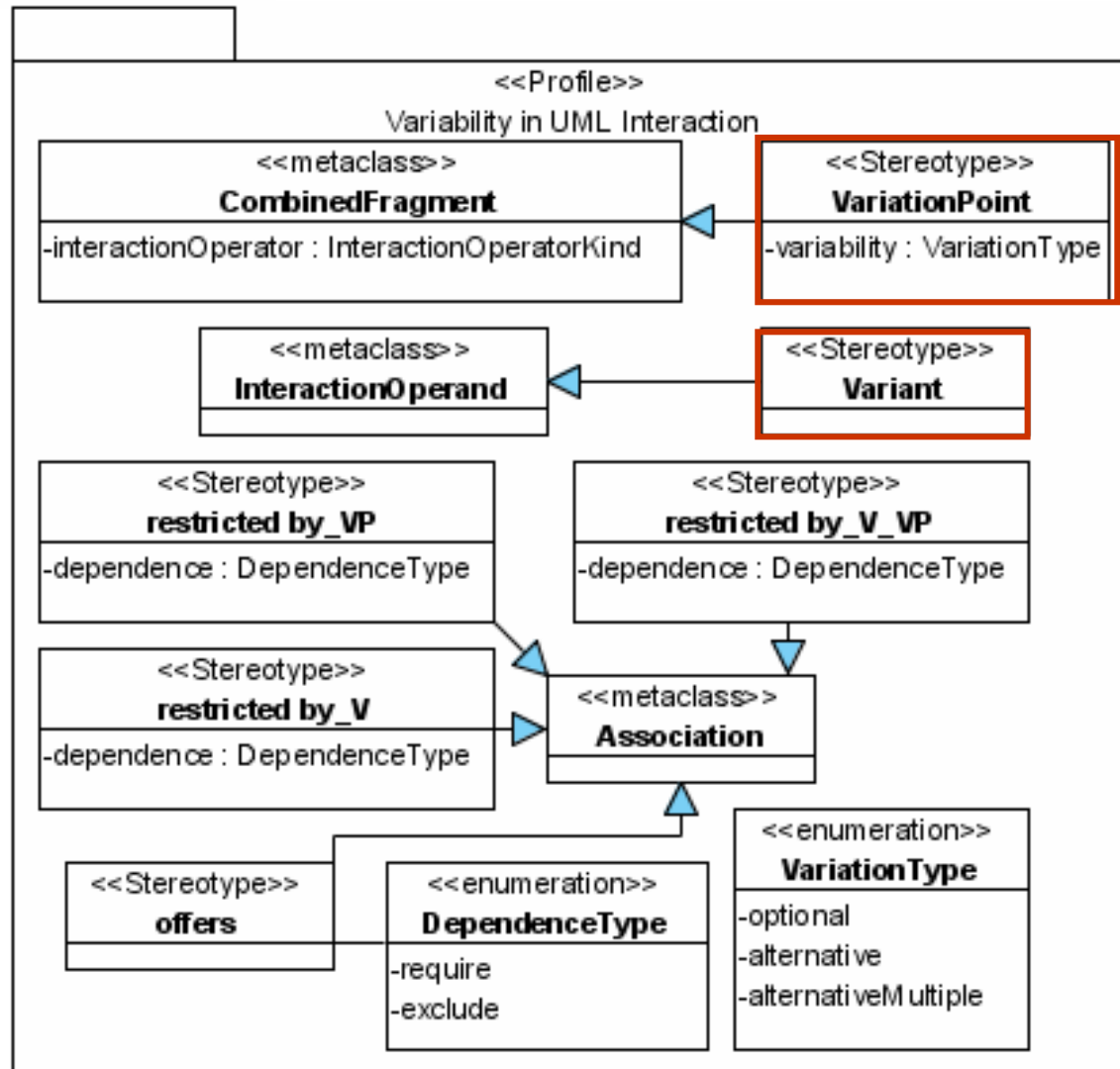
Extension to UML sequence diagram

- To represent variability in sequence diagram, the CombinedFragment is extended.
- In the extension, CombinedFragment represents the Variation Point and the InteractionOperand represents the Variants.





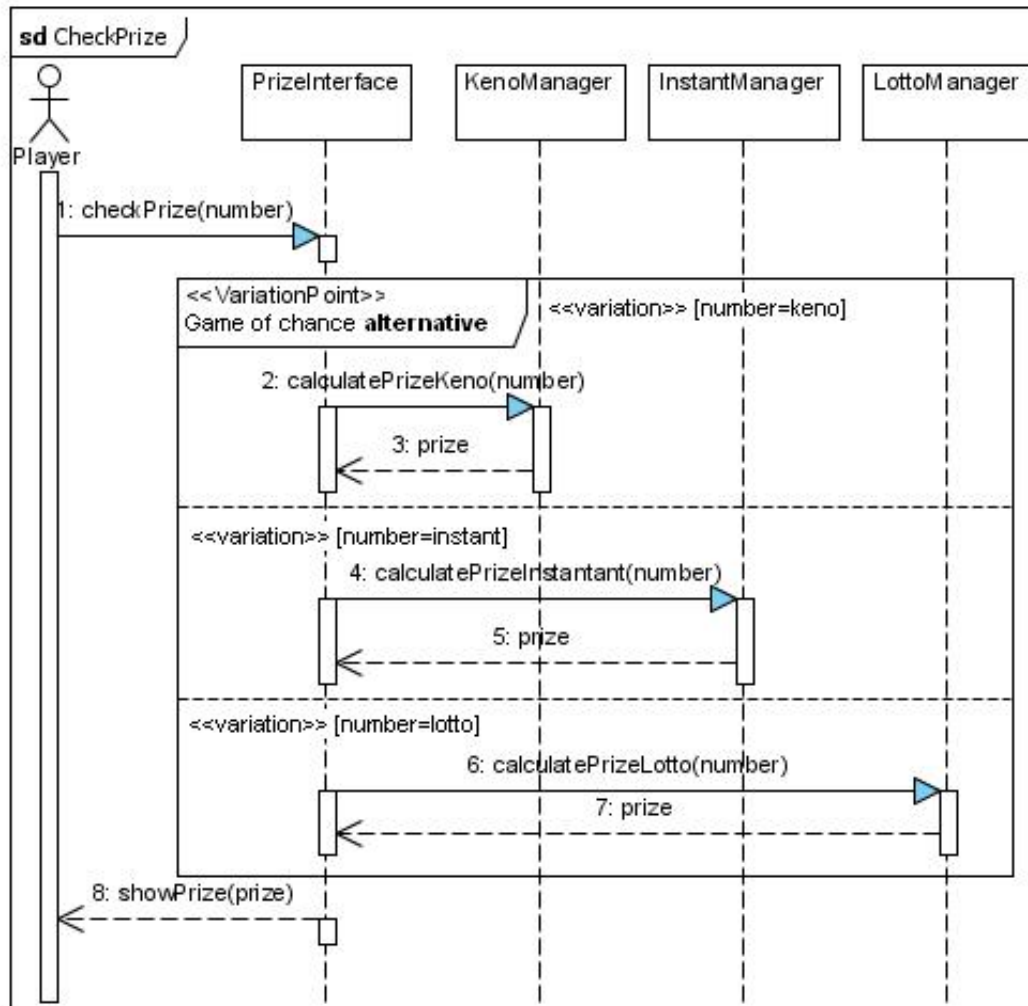
Extension to UML sequence diagram as a UML Profile





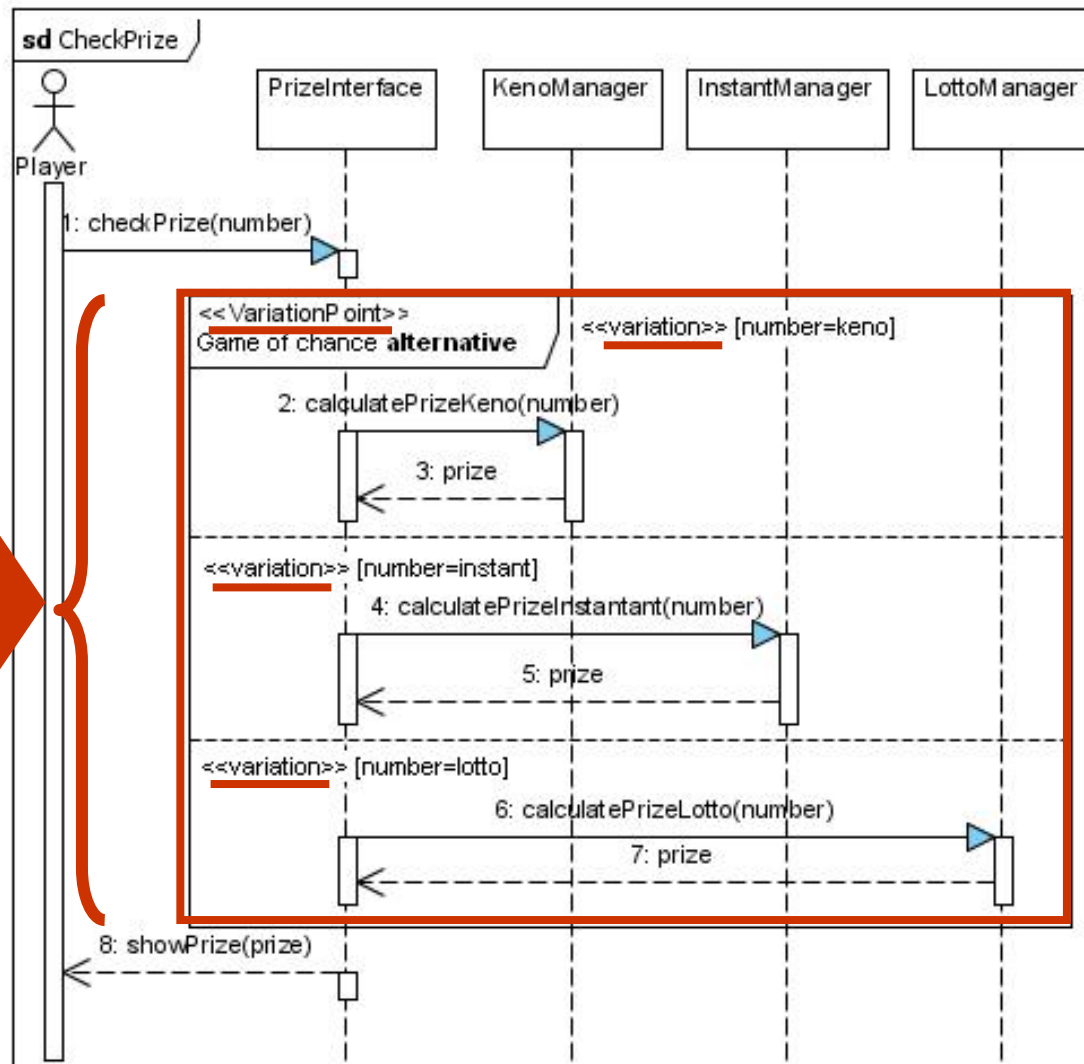
Example CheckPrize functionality

- Player wants to know if his previously purchased ticket is a prizewinner.





Example – CheckPrize functionality



VARIATION POINT

VARIANT

VARIANT

VARIANT

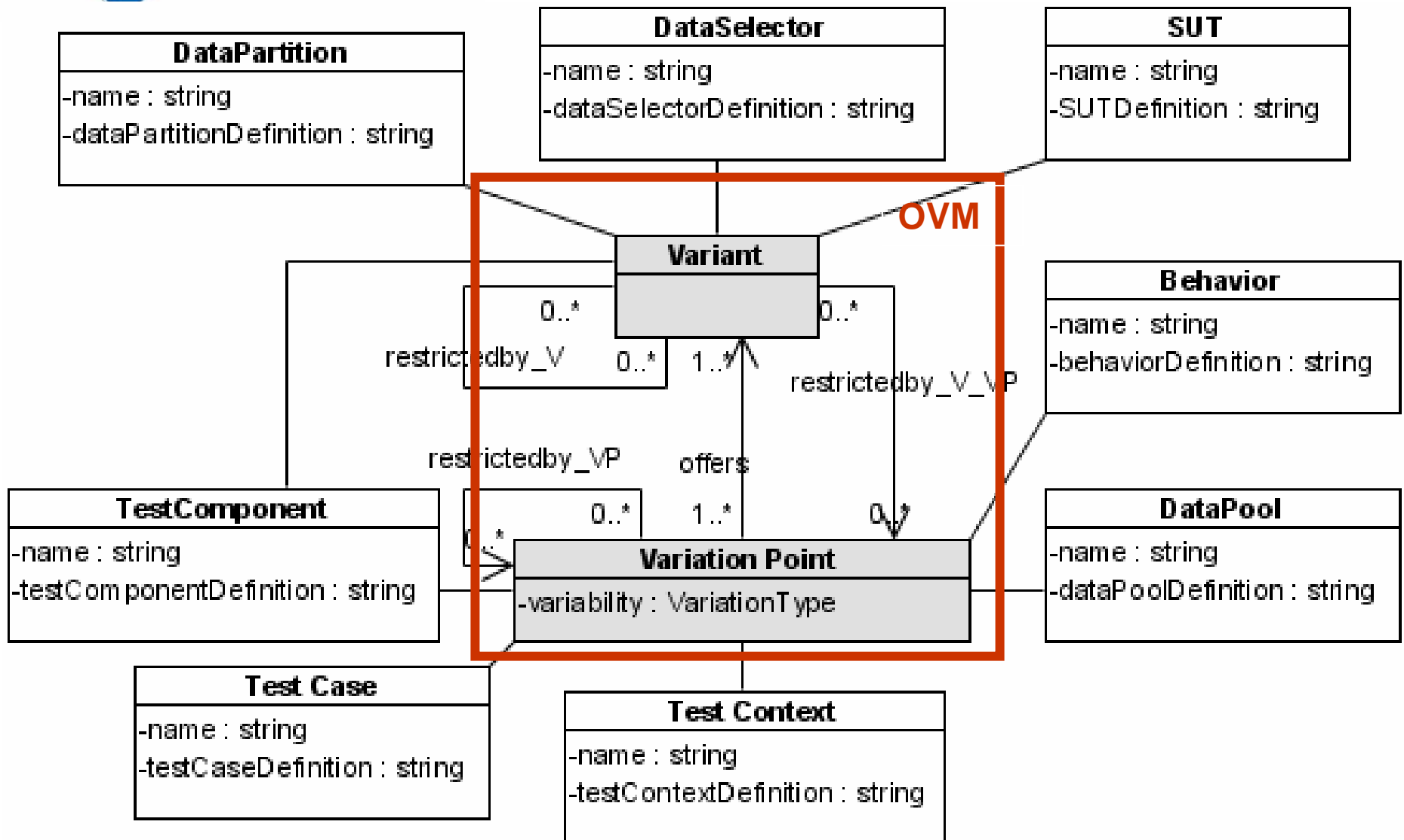


Combined Fragment extension

- In the UML 2.0, the CombinedFragment and its operands are part of the final product
 - In execution time the actor selects the option to execute
- In SPL, the variants parts can be not included in the final product.
- In our extension, a CombinedFragment stereotyped as «Variation Point», the functionality in each option can be part of the product or not
 - If it is not part, then the functionality will not be present in the final product.

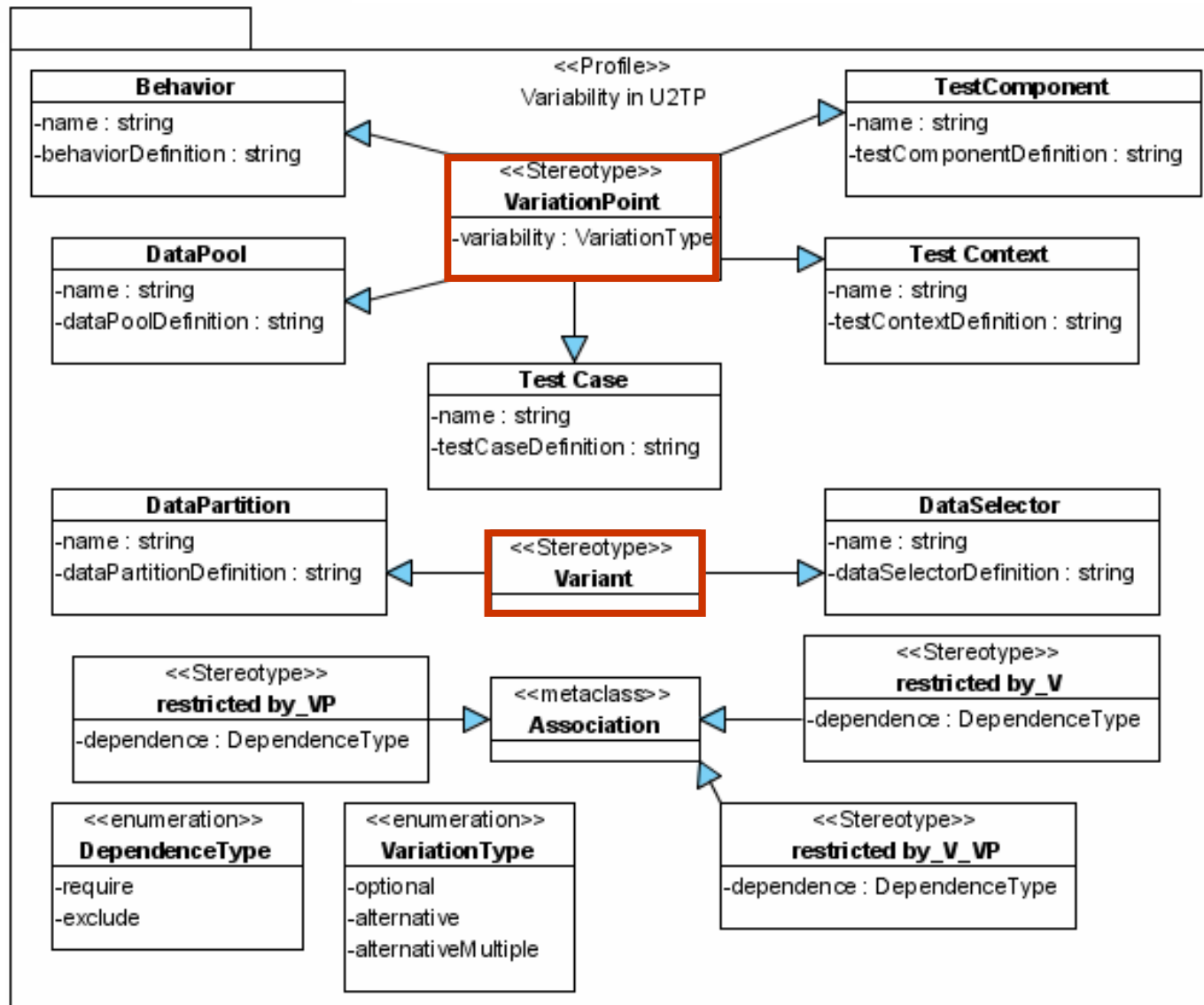


UML-TP Extension





UML-TP Extension



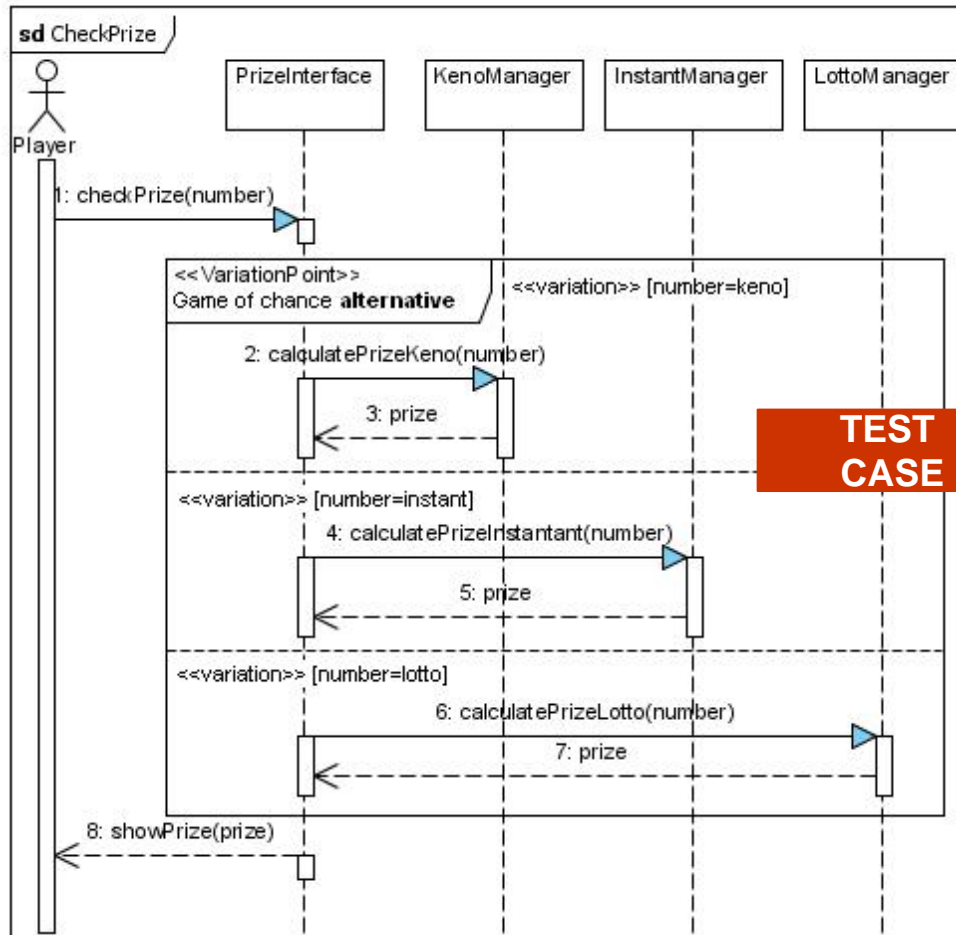


Test case behaviour in SPL

- Functional testing (black box)
- The steps to generate the test case behaviour are:
 1. Test the common features in the entire SPL
 2. Test the variant features in each product
 3. Compare the results between 1 and 2



Example - Test Case



Steps to obtain the test case:

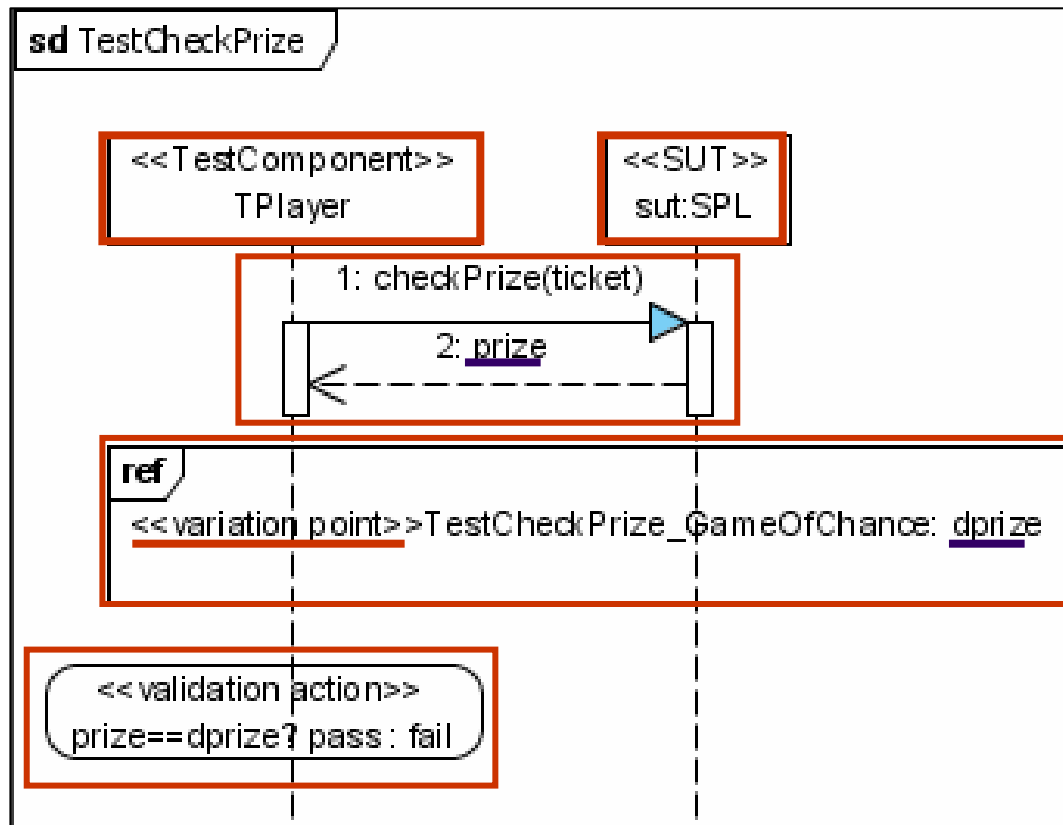
- Obtain the ticket prize from the SUT representing the entire SPL.
- Test the variability in another test case
- The prizes resulting from the first and second part are compared and the test case is considered to be successful if both prizes are equal.



Example - Test Case

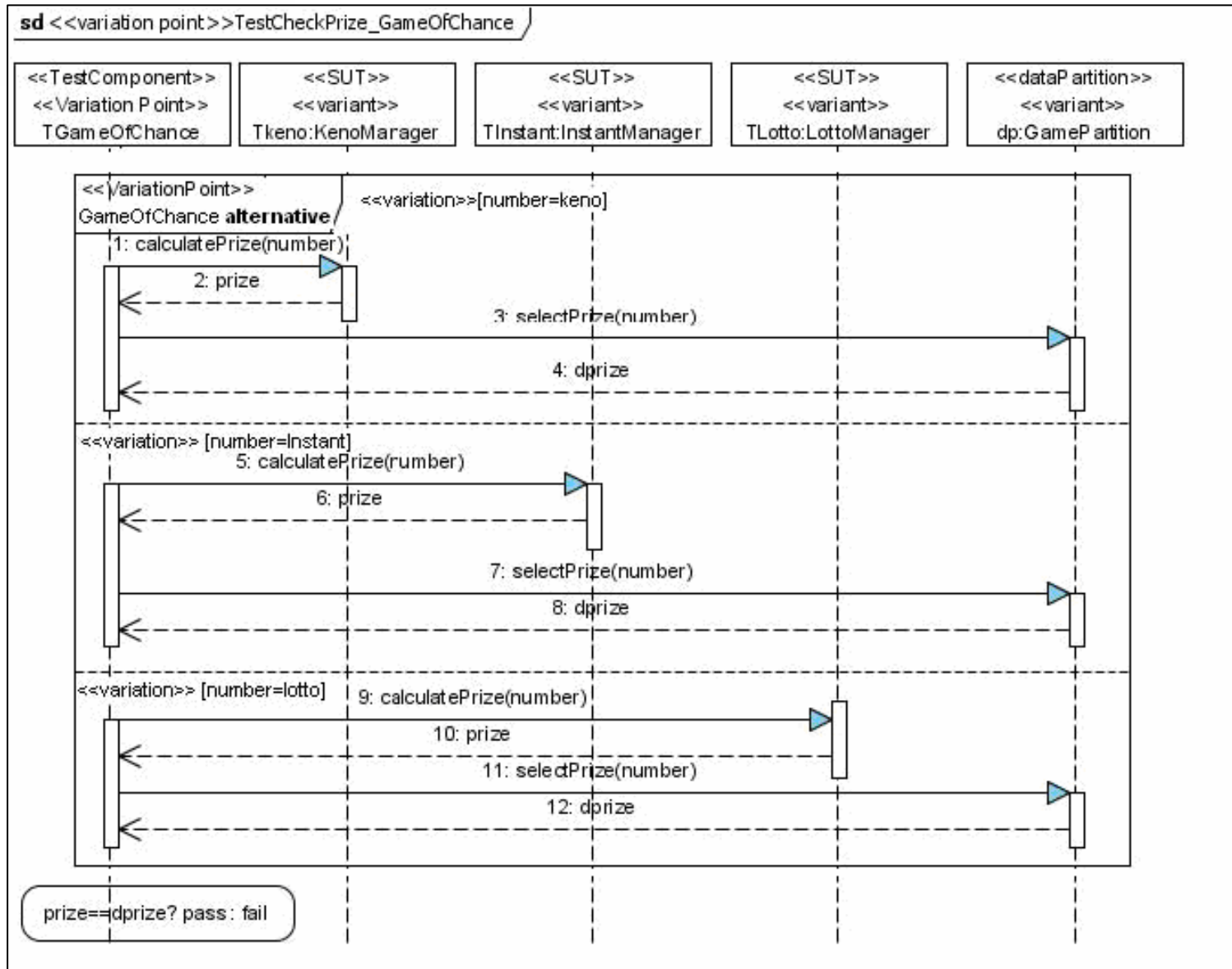
Steps to obtain the test case:

- Obtain the ticket prize from the SUT representing the entire SPL.
- Test the variability in another test case
- The prizes resulting from the first and second part are compared and the test case is considered to be successful if both prizes are equal.





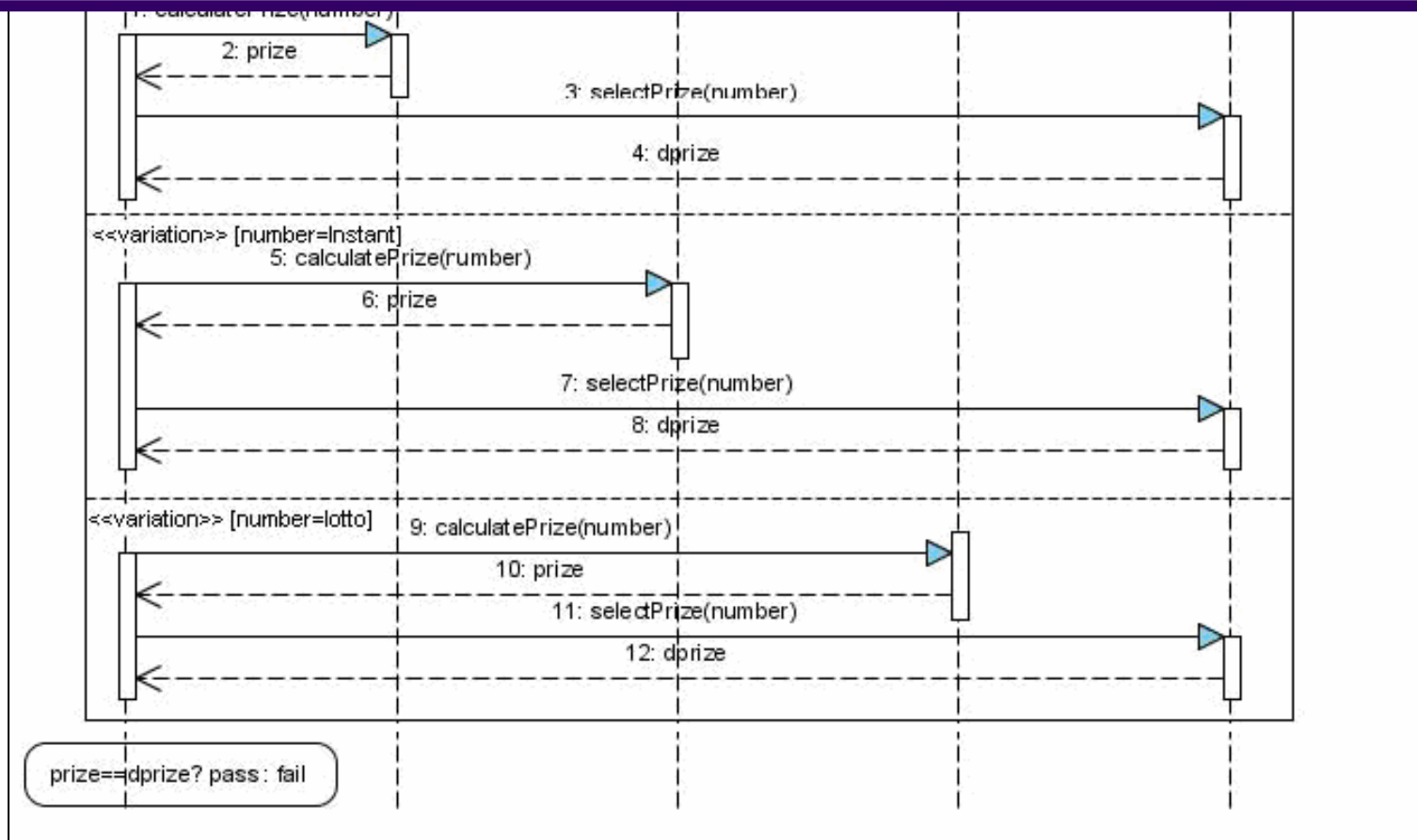
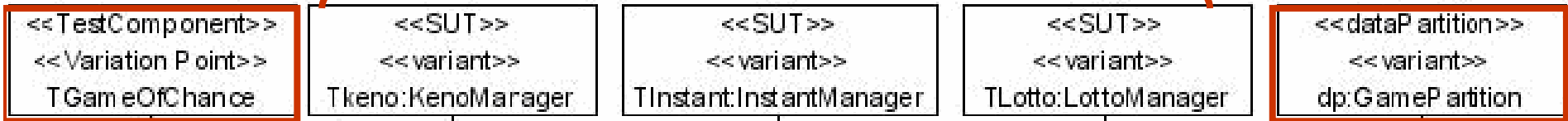
Referenced test case





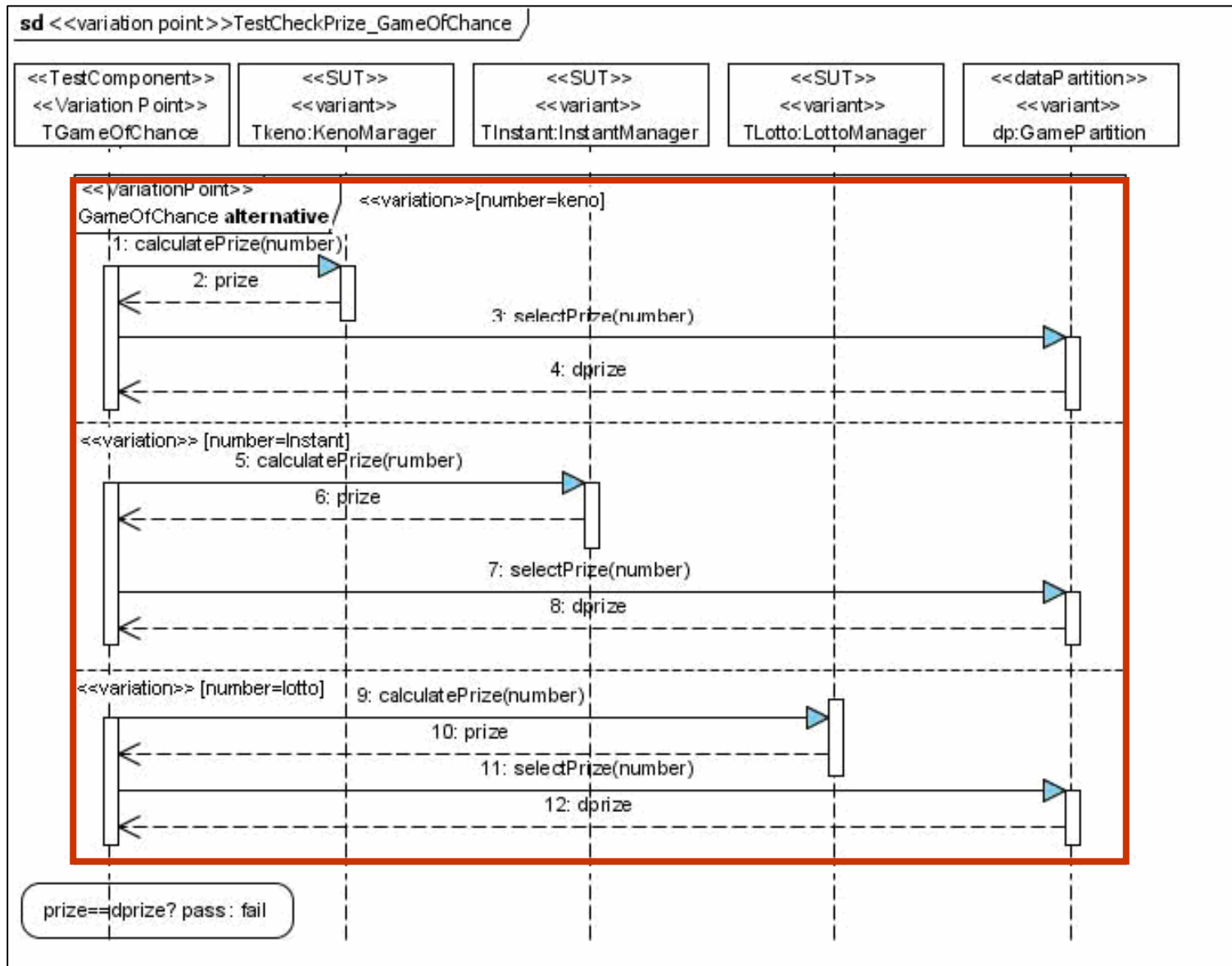
Referenced test case

sd <<variation point>>TestCheckPrize_GameOfChance



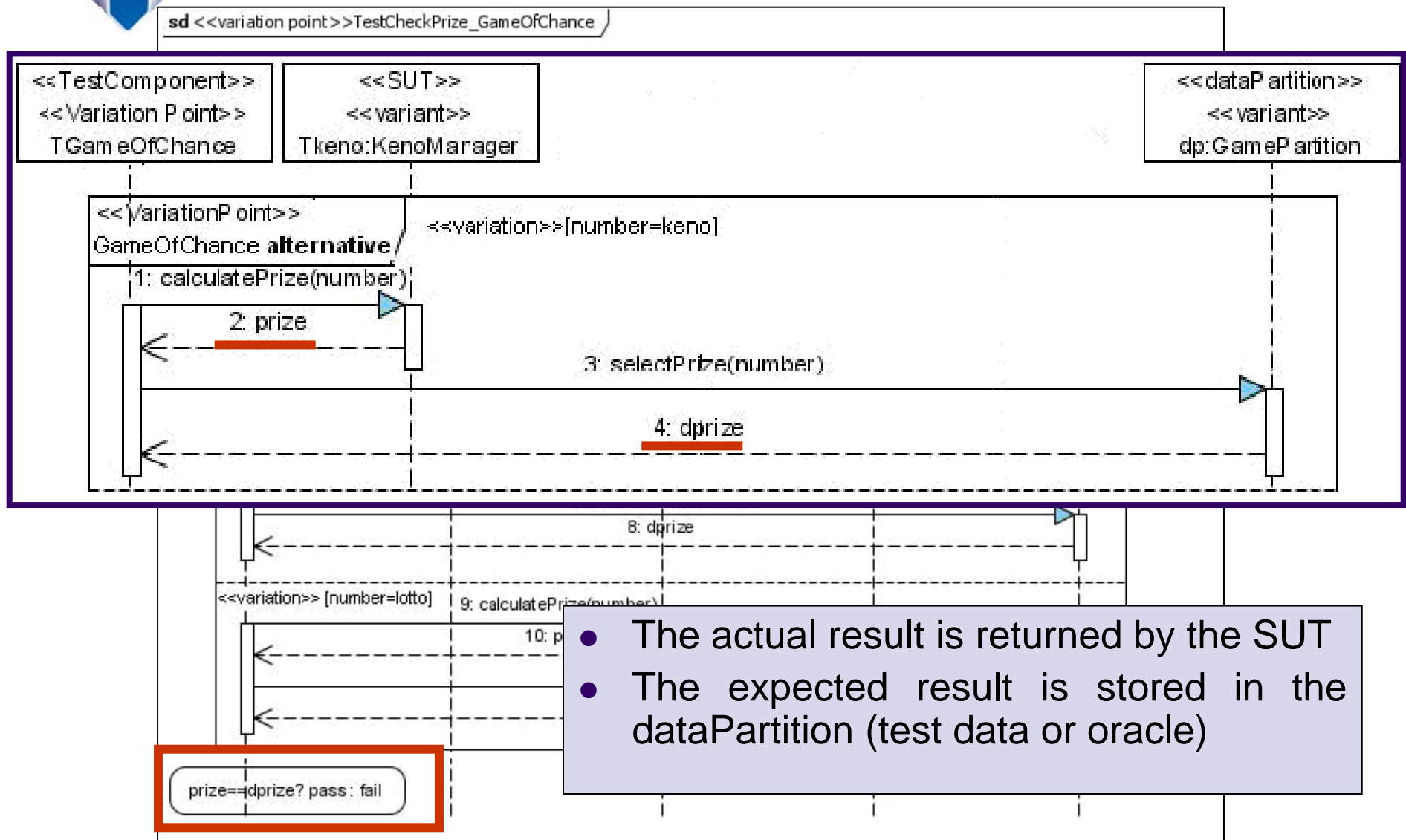


Referenced test case





Referenced test case





Conclusion

- This paper defines a model to handle the variability in SPL testing, reusing the metamodel defined by the UML Testing Profile.
- The test case behavior is modeled with sequence diagrams. An extension to the UML interactions to manage variability has been defined.



Future work

- Generate automated test models for SPL using the model transformation language Query/View/Transformation (QVT)
- Source models:
 - UML sequence diagram using the Profile for variability defined
- Target model:
 - Test model conforms UML Testing Profile using the Profile for variability defined
- These transformations are being successfully developed



Questions?

Thank you for your attention

Beatriz Pérez Lamancha

*Software Testing Centre
(CES)*

*School of Engineering
University of the Republic
Montevideo, Uruguay
bperez@fing.edu.uy*

Macario Polo, Mario Piattini

*Alarcos Research Group
Information Systems and
Technologies Department*

*University of Castilla-La Mancha
Ciudad Real, Spain*

{macario.polo,mario.piattini}@uclm.es