

PLEES'03

Derivation of Domain Test Scenarios from Activity Diagrams¹


Andreas Reuys, Sacha Reis, Erik Kamsties, Klaus Pohl

University of Duisburg-Essen
Software Systems Engineering
Schützenbahn 70, 45117 Essen

Erfurt, 23. September 2003

¹This work was partially funded by the CAFÉ project „From Concept to Application in System Family Engineering“; Eureka 2I 2023 Programme, ITEA Project ip00004 (BMBF, Förderkennzeichen 01 IS 002 C) and the state Nord-Rhein-Westfalia.


Universität Duisburg-Essen
Software Systems Engineering23.9.2003© Sacha Reis 1/14





Contents

1. Introduction
2. Representation of Variability in Activity Diagrams
3. Coverage Criteria for Test Scenario Derivation
4. Procedure for Test Case Derivation
5. Summary

Universität Duisburg-Essen
Software Systems Engineering23.9.2003© Sacha Reis 2/14

| | | |
|--|---------------------------------|-------------------|
|  | Goals & Requirements | 1.1 |
| <p>Goals</p> <ul style="list-style-type: none"> ▀ Derive scenarios for test case derivation ▀ Derive domain test scenarios to reduce test effort <ul style="list-style-type: none"> • use of common test scenarios for each application • adaptation of variability containing domain test scenarios for specific applications ▀ Systematic derivation of test scenarios to ensure sufficient test coverage <p>Requirements</p> <ul style="list-style-type: none"> ▀ Representation of domain scenarios containing variability ▀ Coverage criteria for systematic derivation | | |
| Universität Duisburg-Essen Software Systems Engineering | 23.9.2003 | © Sacha Reis 3/14 |

| | | |
|---|------------------------------------|-------------------|
|  | Assumptions & Questions | 1.2 |
| <p>Assumptions</p> <ul style="list-style-type: none"> ▀ Use case scenarios are well proven for test case derivation ▀ Activity diagrams are one possibility to represent a set of test scenarios ▀ All scenarios can be represented by one activity diagram ▀ For an early start of test activities, test scenarios must be derived during the development of use cases <p>Open Questions that must be answered:</p> <ul style="list-style-type: none"> ▀ How can variability be represented by activity diagrams? ▀ How can test scenarios be derived from activity diagrams containing variability? | | |
| Universität Duisburg-Essen Software Systems Engineering | 23.9.2003 | © Sacha Reis 4/14 |




Brief Overview of Variability

1.3

- **Variation Point:** Represents a delayed design decision and the possible choices.
- **Variant:** One available choice of a variation point
- A variant in a context of a variation point may be:
 - Optional
 - Mandatory
- A variation point needs additional binding information:
 - Alternative (e.g. 1 of 3)
 - Co-existing (e.g. 2 of 3)

must be represented by the activity diagram

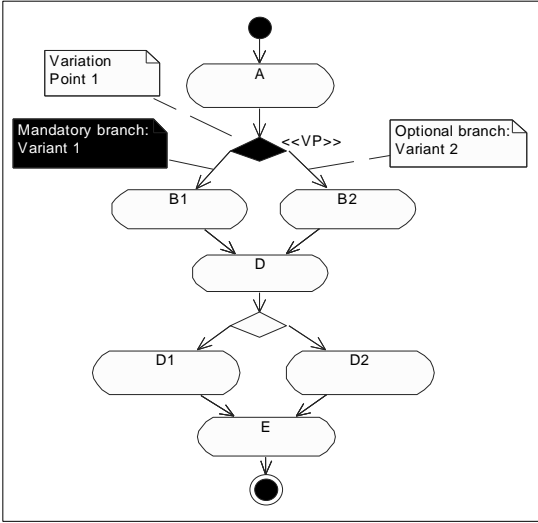
Universität Duisburg-Essen
Software Systems Engineering
23.9.2003
© Sacha Reis 5/14



Variability in Activity Diagrams

2.1

- **New stereotype** for decision points: VP
- **Marking VP** with black color
- **Annotations** for:
 - Variation point
 - Mandatory branch
 - Optional branch
- Call the extension **V-activity diagrams**



The diagram illustrates a flow starting at a start node, leading to activity node A. From A, a decision diamond labeled <<VP>> branches into two paths: a mandatory branch (black diamond) leading to B1, and an optional branch (white diamond) leading to B2. Both B1 and B2 lead to node D. From D, another decision diamond branches into D1 and D2, which both lead to node E, ending at a final node.

Universität Duisburg-Essen
Software Systems Engineering
23.9.2003
© Sacha Reis 6/14

Alternative and Co-Existing Variants

2.2

- **Annotations for:**
 - Alternative
 - Co-Existing

- Information is needed for later binding the variants in **application engineering**

Universität Duisburg-Essen
Software Systems Engineering
23.9.2003
© Sacha Reis 7/14


Test Scenario Derivation

3.1

- Well-known **coverage criteria** for source code:
 - Statement coverage
 - Branch coverage
 - Path coverage

- Reuse of existing **adoption to use cases**
- Statement coverage is a poor criteria
- Path coverage caused an enormous number of scenarios
- **Branch coverage** seems to be the most practical criteria
 - **Must be proofed concerning the use for V-activity diagrams**

Universität Duisburg-Essen
Software Systems Engineering
23.9.2003
© Sacha Reis 8/14

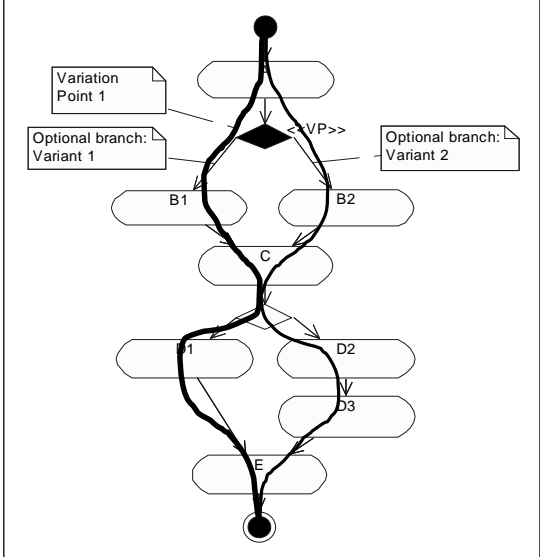


Problem with Conventional Branch Coverage


3.2

- With conventional branch coverage:
 1. {A, B1, C, D1, E}
 2. {A, B2, C, D2, D3, E}

- Binding variant 1: D2 and D3 are not considered by a scenario



Universität Duisburg-Essen
Software Systems Engineering
23.9.2003
© Sacha Reis 9/14



Extent for Branch Coverage for Solution

3.3

- Extension:
 - **Each branch of each possible application must be covered by at least one test scenario**

- Criteria would lead to a large number of test scenarios
 - **Test scenarios must also contain variability**

- **Two-step procedure** for derivation
 1. Derive scenarios without variability and with placeholders for variants
 2. Refine scenarios by filling placeholders with variants

Universität Duisburg-Essen
Software Systems Engineering
23.9.2003
© Sacha Reis 10/14

Step 1 of Test Scenario Derivation 4.1


1. Derive test scenarios so that each branch that does not reflect an optional variant of the V-activity diagram is covered at least once. Leave an empty variation point in the scenarios for the other kinds of branches.

Universität Duisburg-Essen
Software Systems Engineering 23.9.2003 © Sacha Reis 11/14

Step 2 of Test Scenario Derivation 4.2

2. Add to the scenarios the steps that cover the optional variants of a variation point. All branches of all variants must be covered. If necessary, additional scenarios must be created.

Universität Duisburg-Essen
Software Systems Engineering 23.9.2003 © Sacha Reis 12/14

| | | |
|--|--------------------------|-----------------------------------|
|  | Summary | 5.1 |
| <ul style="list-style-type: none">➤ Derivation of test scenarios describes the first step of the ScenTED method for test case derivation.➤ V-activity diagrams represents variability in control flow.➤ The extension of the conventional branch coverage considers variability aspects<ul style="list-style-type: none">▪ optional / mandatory▪ alternative / co-existing➤ A two-step procedure helps to derive test scenarios from V-activity diagrams in domain engineering.➤ Shortcoming: Dependencies between variation points are not considered so far. | | |
| <small>Universität Duisburg-Essen Software Systems Engineering</small> | <small>23.9.2003</small> | <small>© Sacha Reis 13/14</small> |

| | | |
|---|--|-----------------------------------|
|  | <h1>Thank you very much !</h1> <h2>Questions ?</h2> | |
| <small>Universität Duisburg-Essen Software Systems Engineering</small> | <small>23.9.2003</small> | <small>© Sacha Reis 14/14</small> |