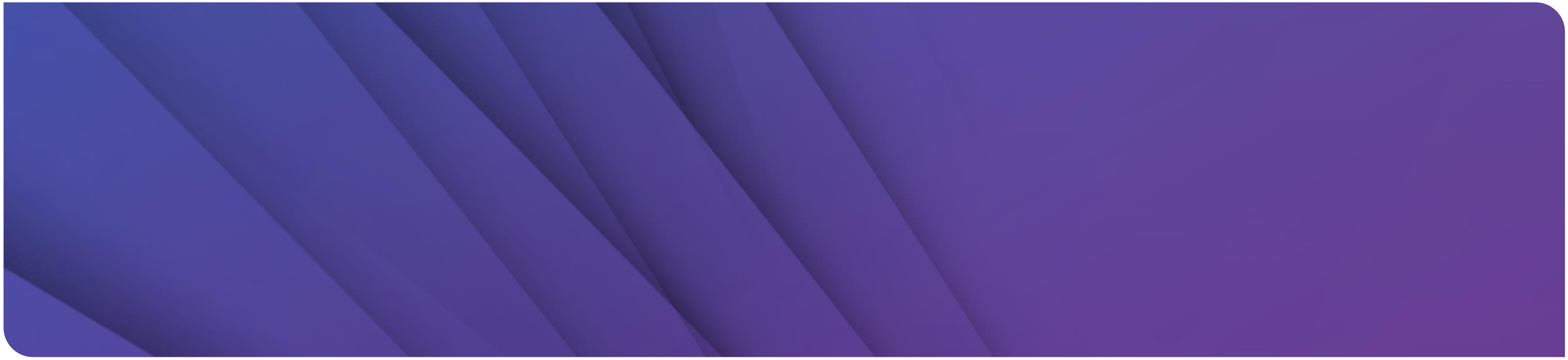
Karlsruher Institut für Technologie

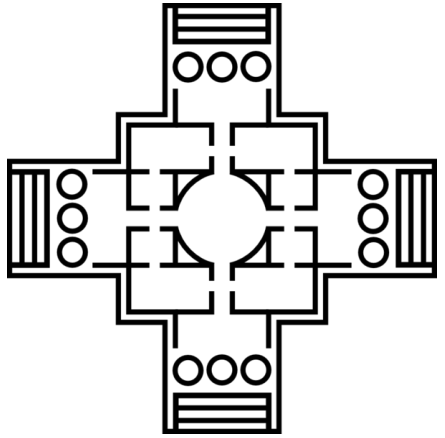
Palladio Fluent API Model Generator – Usage Model

Eva-Maria Neumann
Praktikum: Werkzeuge für Agile Modellierung
Betreuer: Yves Kirschner



Inhalt

- Motivation
- Grundlagen
  - Palladio Model
  - Fluent Interfaces
- Palladio Fluent API
  - Vorarbeiten
  - Besonderheiten Usage Model



Motivation

- Graphische Editoren
  - \- Unübersichtlich bei großen Modellen
  - \- Viel Klicken
  - \+ Anschaulich
  - \+ Benutzerfreundlich
  - \+ Für Einsteiger geeignet
- Textuell
  - \- Aufwändig
  - \- Viel Quelltext
  - \- Unverständlich
  - \- Unübersichtlich
  - \+ Hohe Lernkurve
  - \+ Schneller

Fluent Interfaces

- Natürliche Sprache
- Verkettung von Methoden

```
private void makeNormal(Customer customer) {
    Order o1 = new Order();
    customer.addOrder(o1);
    OLine line1 = new OLine(6,product(„Eier"));
    o1.addLine(line1);
    OLine line2 = new OLine(5,product(„Tomate"));
    o1.addLine(line2);
    OLine line3 = new OLine(3,product(„Gurke"));
    o1.addLine(line3);
    line2.setSkippable(true);
    o1.setRush(true);
}
```

- Besser Vorstellbar
- Weniger Fehler

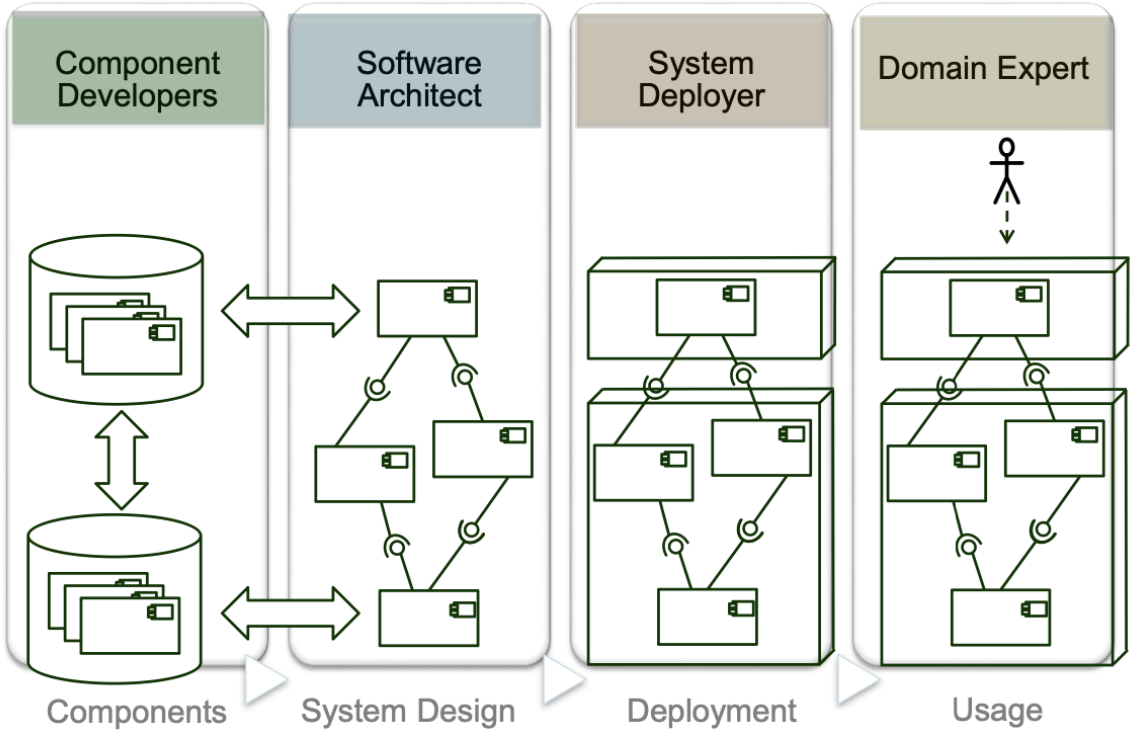
```
private void makeFluent(Customer customer) {
    customer.newOrder()
          .with(6, „Eier")
          .with(5, „Tomate").skippable()
          .with(3, „Gurke")
          .priorityRush();
}
```

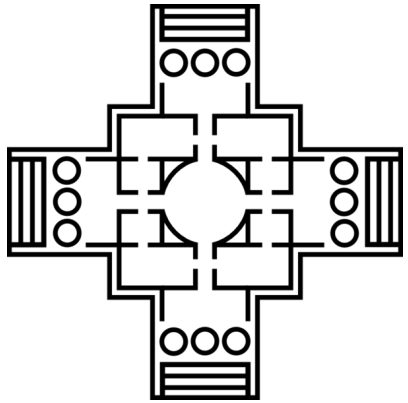
➢ Lesbarer, Kompakter Code

Quelle: https://martinfowler.com/bliki/FluentInterface.html

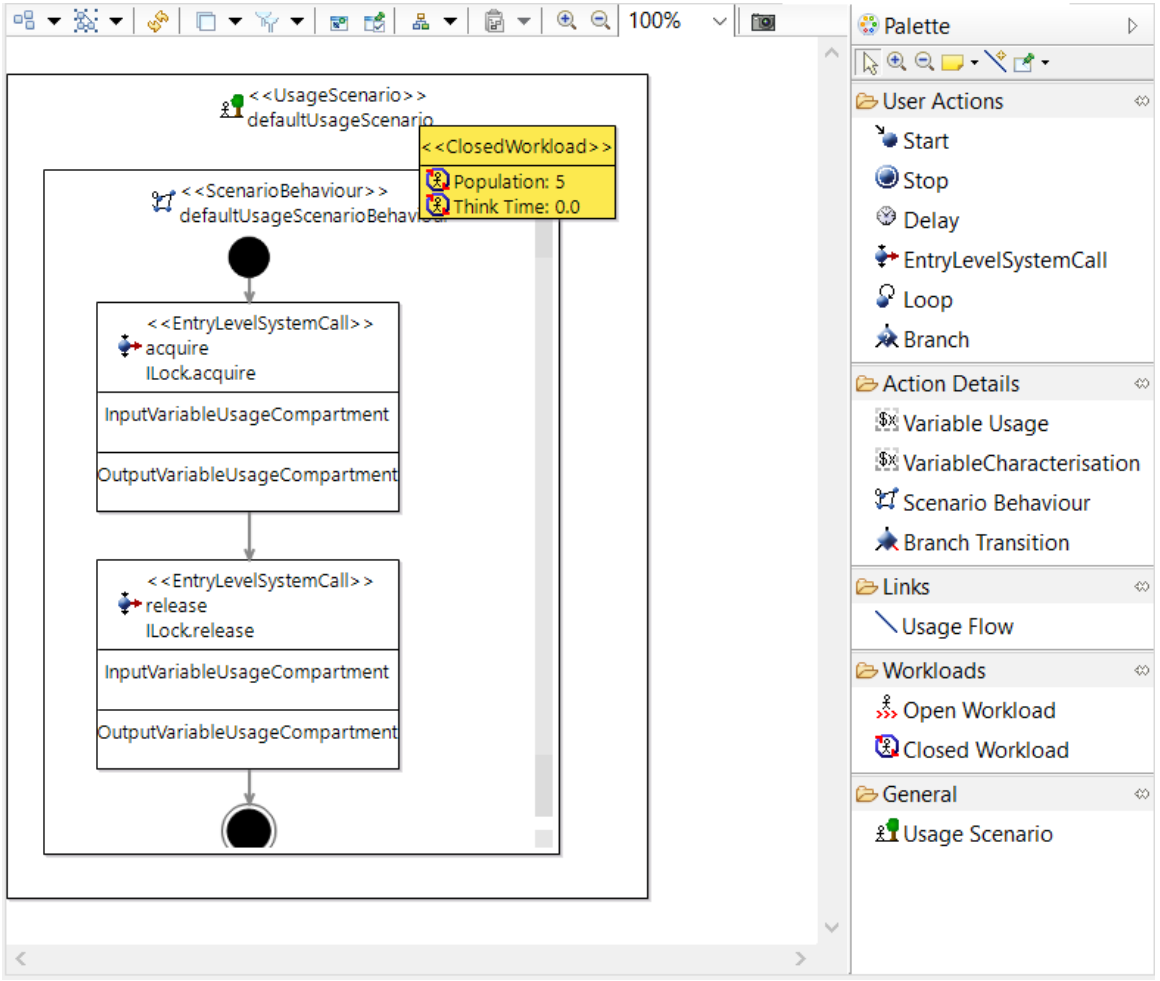
Palladio Component Model

- Palladio: Ansatz zur Software Architektur Simulation
- PCM: Domänspezifische Modellsprache mit Metamodell

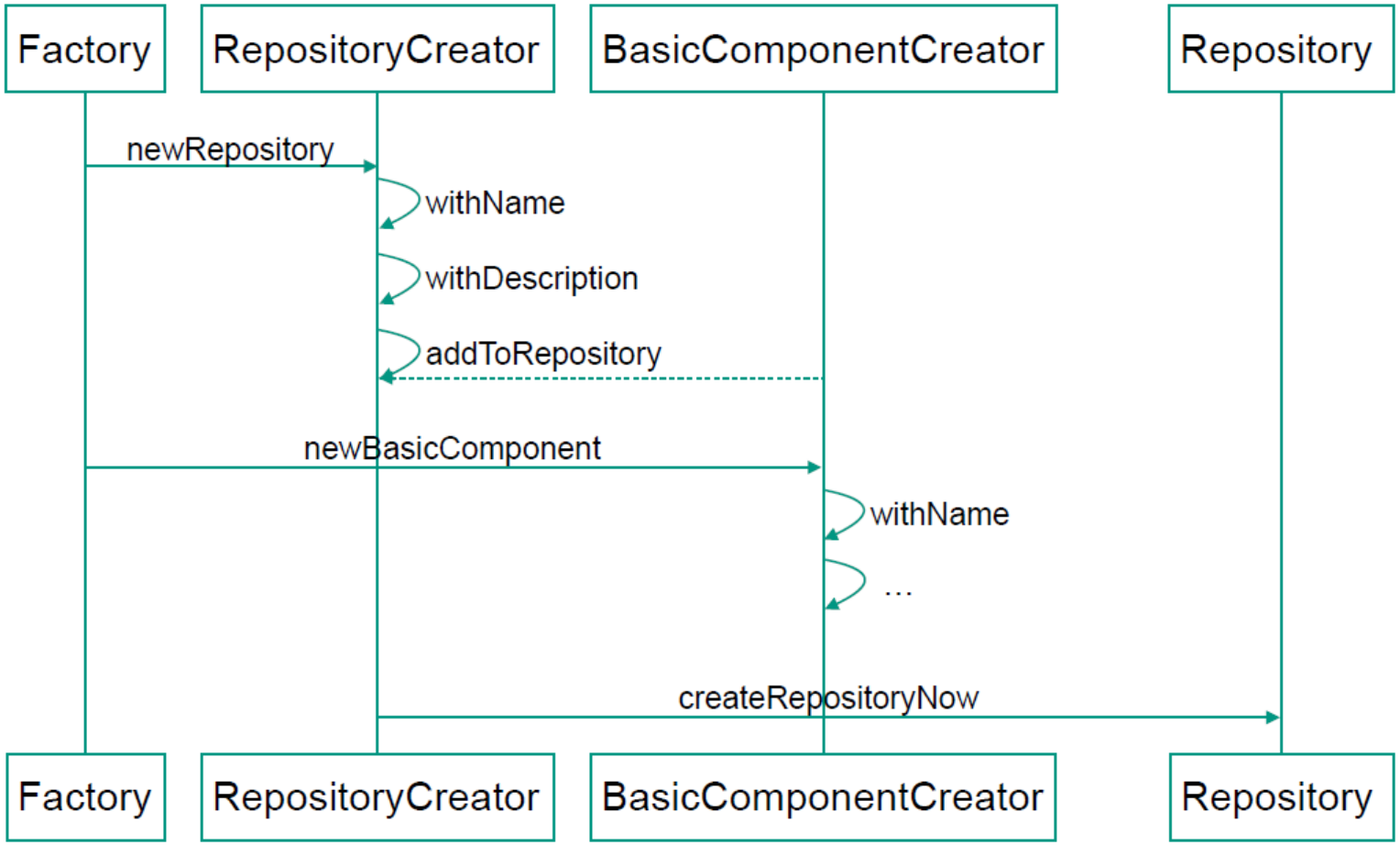




Palladio Component Model



Palladio Fluent API - Vorarbeiten

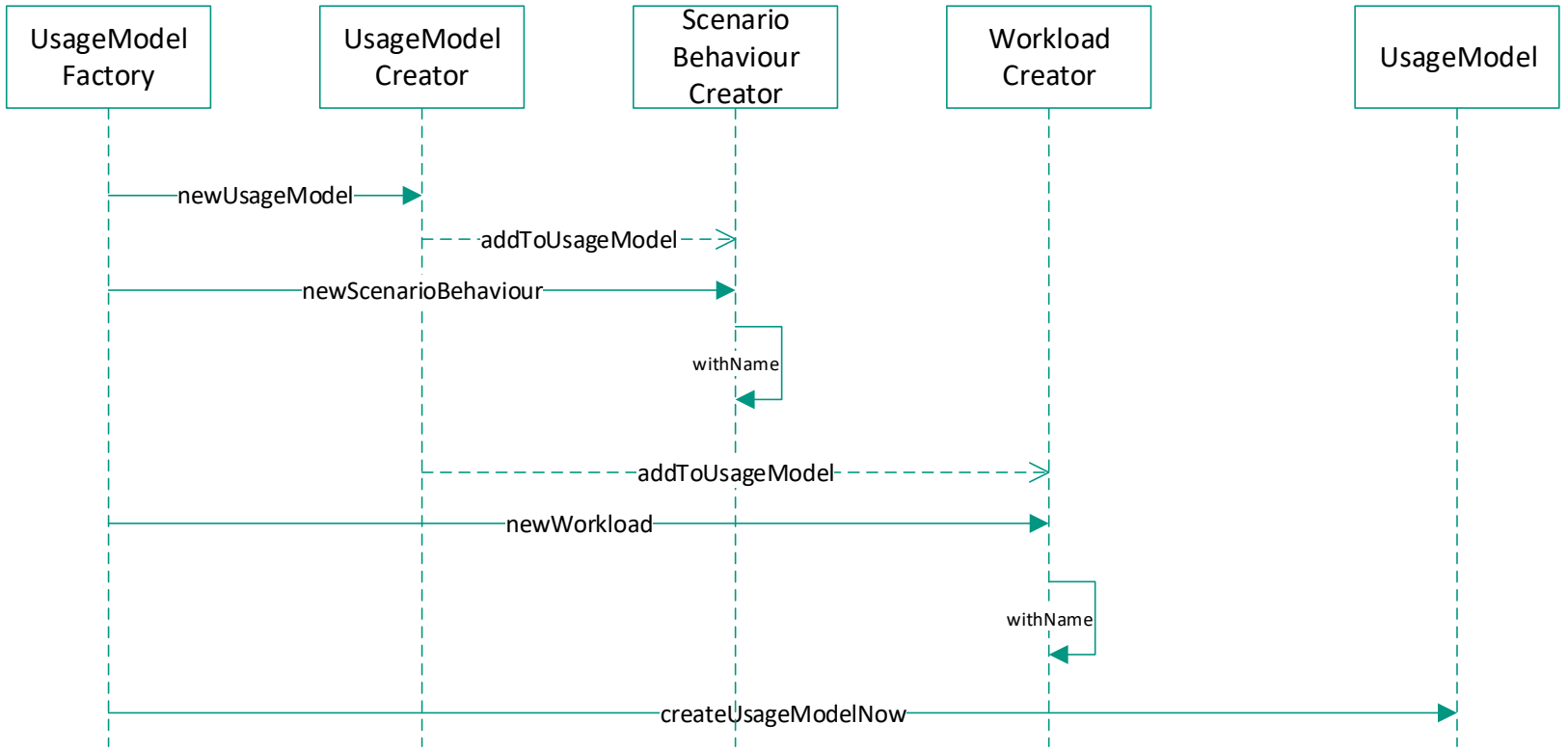


Methodenverkettung:
a().b().c()

Argument:
a(x)

Quelle: Folie von Louisa Lambrecht

Übertragung auf Usage Model



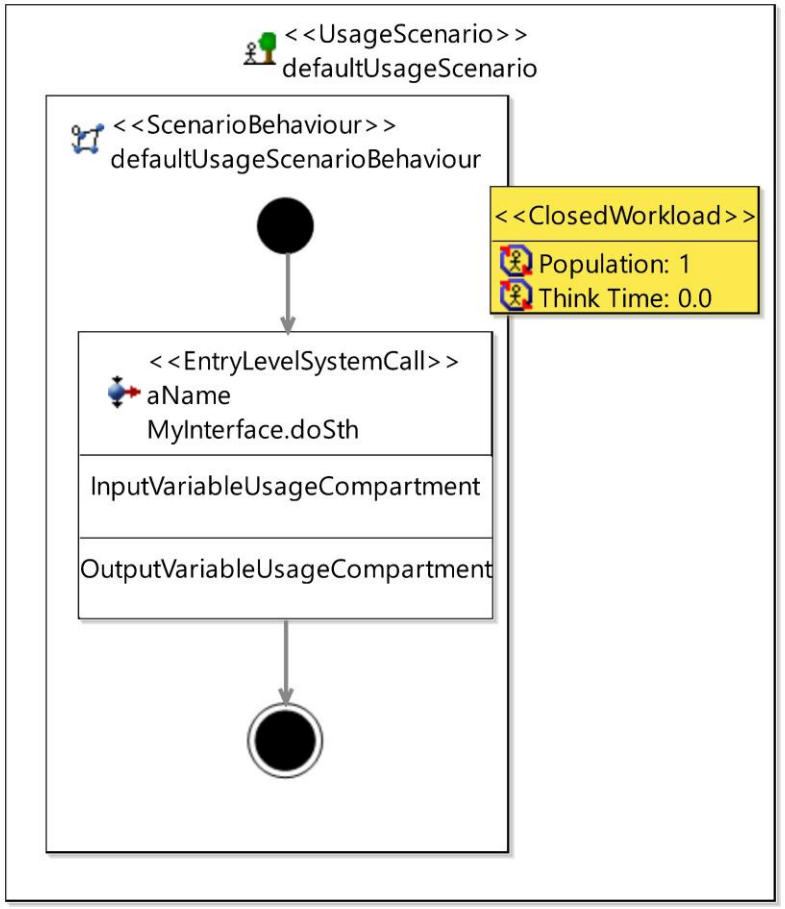
Methodenverkettung:
a().b().c()

Argument:
a(x)

Besonderheiten Usage Model

- Aufbau Fluent API Usage Model in eigener Factory
- Verkettung von Aktionen
  - Start/ Stop automatisiert hinzufügen
  - Reihenfolge wichtig
- Unterstützung der Verifikation durch Parameter
  - Etwas unübersichtlicher
  - Vermeidung von Fehlern
  - Einfach Erzeugung von gültigem Model

Usage Model - Beispiel

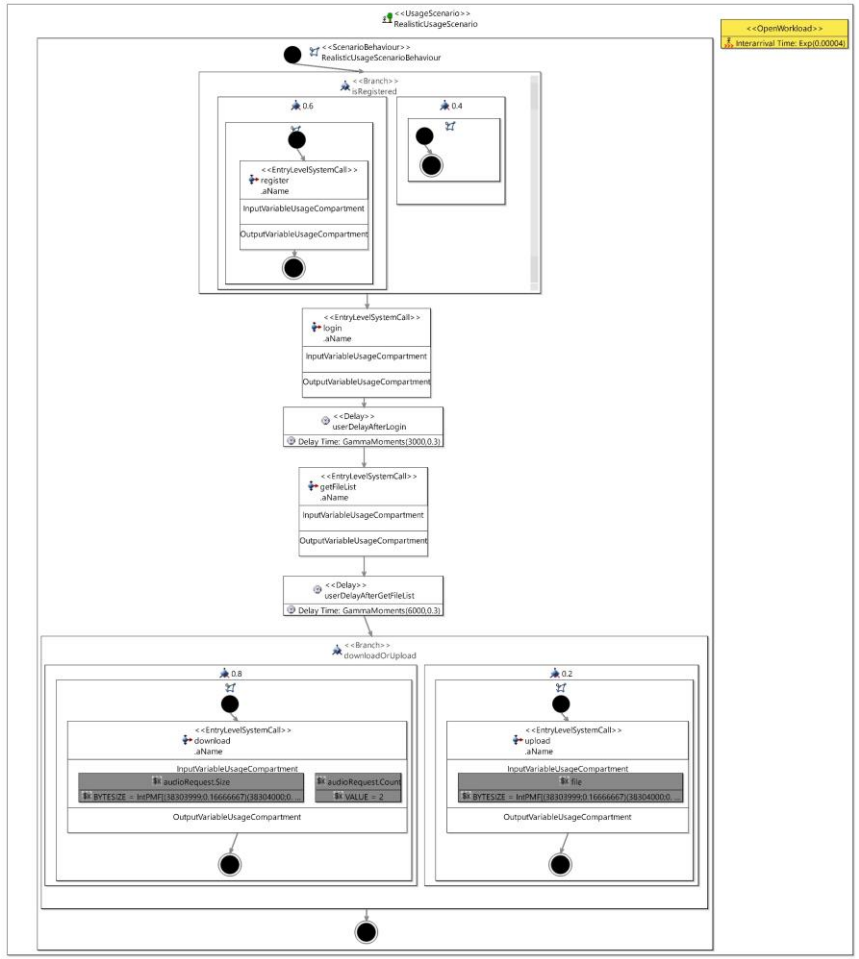


Usage Model – Weiteres Beispiel

```
UsageModel usgModel = this.create.addSystem(mediaStoreMockUp()).newUsageModel().addToUsageModel(
        this.create.newUsageScenario(
                this.create.newScenarioBehavior().withName("RealisticUsageScenarioBehaviour")
                .addToScenarioBehaviour(
                        this.create.newStartAction().withName("startUsage").withSuccessor(
                        this.create.newBranchAction().withName("isRegistered")
                            .addToBranchAction(this.create.newBranchTransition(this.create.newScenarioBehavior().addToScenarioBehaviour(
                                    this.create.newEntryLevelSystemCall(
                                            this.create.fetchOffOperationRoleAndSignature("defaultSystem","Provided_IWebGui", "register")).withName(
                                    .withName("needsToRegister")).withProbability(0.6))
                            .addToBranchAction(
                                    this.create.newBranchTransition(this.create.newScenarioBehavior().withName("isAlreadyRegistered")
                                    ).withProbability(0.4)).withSuccessor(
                        this.create.newEntryLevelSystemCall(
                                this.create.fetchOffOperationRoleAndSignature("defaultSystem","Provided_IWebGui","login"))
                                .withName("login").withSuccessor(
                        this.create.newDelayAction("GammaMoments(3000,0.3)").withName("userDelayAfterLogin").withSuccessor(
                        this.create.newEntryLevelSystemCall(
                                this.create.fetchOffOperationRoleAndSignature("defaultSystem","Provided_IWebGui","getFileList"))
                                .withName("getFileList").withSuccessor(
                        this.create.newDelayAction("GammaMoments(6000,0.3)").withName("userDelayAfterGetFileList").withSuccessor(
                        this.create.newBranchAction().withName("downloadOrUpload")
                            .addToBranchAction(this.create.newBranchTransition(this.create.newScenarioBehavior().withName("downloadCase")
                                    .addToScenarioBehaviour(this.create.newEntryLevelSystemCall(
                                            this.create.fetchOffOperationRoleAndSignature("defaultSystem","Provided_IWebGui","download"))
                                            .withName("download")
                                            .addToEntryLevelSystemCallInput(
                                                    this.create.newVariableUsage("audioRequest", "Size")
                                                    .withVariableCharacterisation("IntPMF[(38303999;0.16666667)(38304000;0.16666667)(40568000;0.166
                                            .addToEntryLevelSystemCallInput(
                                                    this.create.newVariableUsage("audioRequest", "Count")
                                                    .withVariableCharacterisation("2", VariableCharacterisationType.VALUE))
                                            )).withProbability(0.8))
                            .addToBranchAction(this.create.newBranchTransition(this.create.newScenarioBehavior().withName("uploadCase")
                                    .addToScenarioBehaviour(this.create.newEntryLevelSystemCall(
                                            this.create.fetchOffOperationRoleAndSignature("defaultSystem","Provided_IWebGui","upload"))
                                            .withName("upload").addToEntryLevelSystemCallInput(
                                                    this.create.newVariableUsage("file").withVariableCharacterisation("IntPMF[(38303999;0.16666667)(
                                            )).withProbability(0.2)).withSuccessor(
                        this.create.newStopAction().withName("stopUsage"))))))),

                this.create.newOpenWorkload("Exp(0.00004)")).withName("RealisticUsageScenario"))

        .createUsageModelNow();
```



Ohne Verifikation

- `UsageScenario`
  - Hat genau ein `ScenarioBehavior`
  - Hat genau einen `Workload` (mit Zeit)

```
UsageModel usgModel = create.newUsageModel().addToUsageModel(
    create.newUsageScenario()
        .addToUsageScenario(create.newScenarioBehavior())
        .addToUsageScenario(create.newOpenWorkload()
            .withArrivalTime("10")))
    .withName(name))
.createUsageModelNow();
```

Mit Verifikation

- `UsageScenario`
  - Hat genau ein `ScenarioBehavior`
  - Hat genau einen `Workload` (mit Zeit)

```
UsageModel usgModel = create.newUsageModel().addToUsageModel(
    create.newUsageScenario(
        create.newScenarioBehavior(),
        create.newOpenWorkload("10"))
    .withName(name))
.createUsageModelNow();
```

Verifikation

```
create.newBranchAction().withName(name).addToBranchAction(
    create.newBranchTransition(
        create.newScenarioBehavior().withName(scenName))
    .withProbability(prop))
```

- Problem: Wahrscheinlichkeit nicht überprüfbar

```
org.palladiosimulator.generator.fluent.shared.validate.ModelValidator
logResult
SCHWERWIEGEND: Validation for model "UsageModel": Diagnostic ERROR
source=org.palladiosimulator.pcm.usagemodel code=0 The
'allBranchProbabilitiesMustSumUpTo1' constraint is violated on
'org.palladiosimulator.pcm.usagemodel.impl.BranchImpl@4ef18604{#_02Mo5hcBEey7
T-wAh8Eu0g}' data=[Branch[TRANSIENT]]
```

Verkettung Aktionen

- `ScenarioBehaviour` hat Aktionen
  - Aktion: Vorgänger, Nachfolger
  - Erste: Immer Start
  - Letzte: Immer Stop

```
UsageModel usgModel = create.addSystem(createSimplifiedMediaStoreSystem())
        .newUsageModel().addToUsageModel(create.newUsageScenario(
          create.newScenarioBehavior()
              .addToScenarioBehaviour(
                  …
              ),
          create.newOpenWorkload("0")))
    .createUsageModelNow();
```

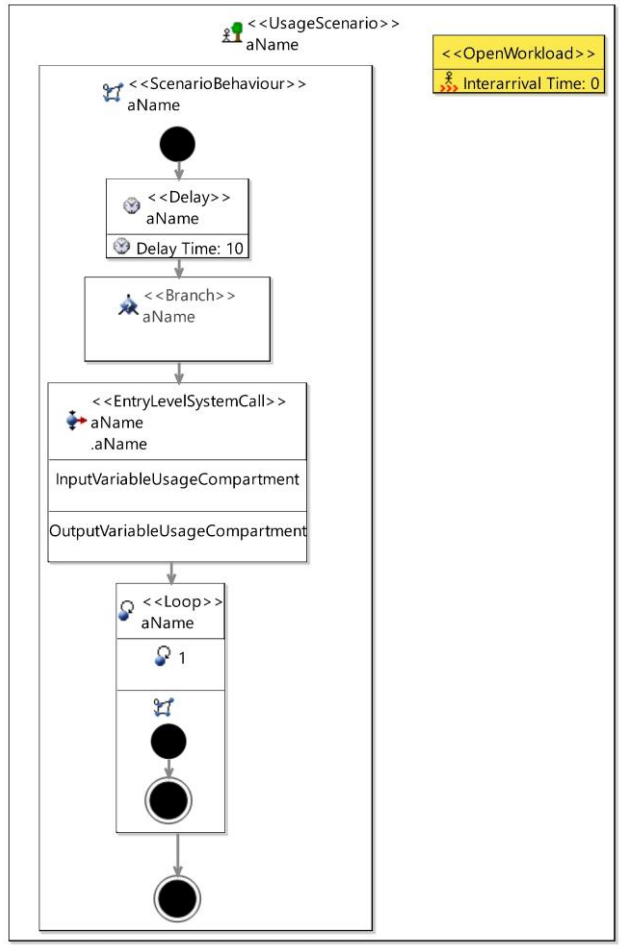
Verkettung Aktionen

```
.addToScenarioBehaviour(create.newStartAction())
.addToScenarioBehaviour(create.newDelayAction("10"))
.addToScenarioBehaviour(create.newBranchAction())
.addToScenarioBehaviour(create.newEntryLevelSystemCall(
    create.fetchOffOperationRoleAndSignature
    ("SimplifiedMediaStore System", provRoleName, operSigName))
.addToScenarioBehaviour(create.newLoopAction("1",
    create.newScenarioBehavior())
.addToScenarioBehaviour( create.newStopAction())
```

- Fehlend: Verkettung
- Problem: Verweis

Verkettung Aktionen

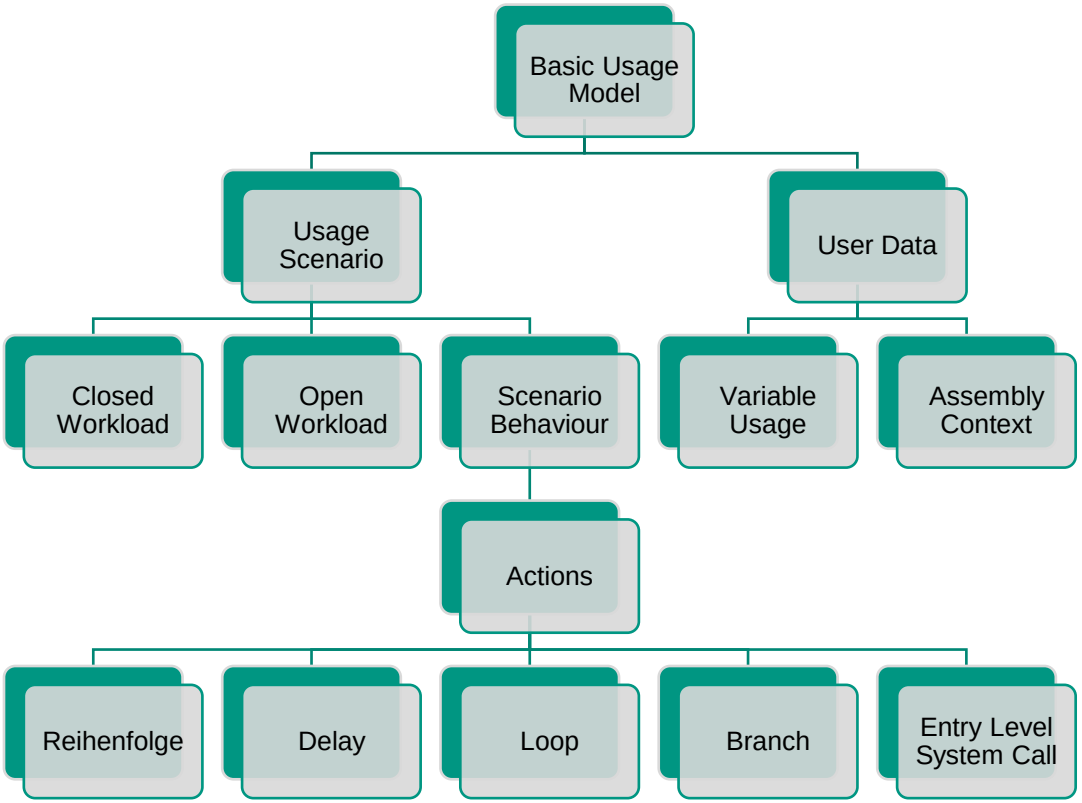
```
create.newStartAction().withSuccessor(
    create.newDelayAction("10").withSuccessor(
        create.newBranchAction().withSuccessor(
            create.newEntryLevelSystemCall(
                create.fetchOffOperationRoleAndSignature
                ("System",provRoleName,operSigName))
                 .withSuccessor(
                    create.newLoopAction("1",
                    create.newScenarioBehavior())
                     .withSuccessor(
                        create.newStopAction()))))
))
```



- Prüfung auf Start und Stop
- Vorgänger markieren

Test

- Vorher
  - Modell per Fluent API aufbauen
  - Modell normal aufbauen
  - Vergleichen
- Jetzt
  - Modelle per Fluent API aufbauen
  - Unit Test von Bereichen und deren Parametern
  - Zusätzlich großer Test: „Realistic Media Store“



Fragen?