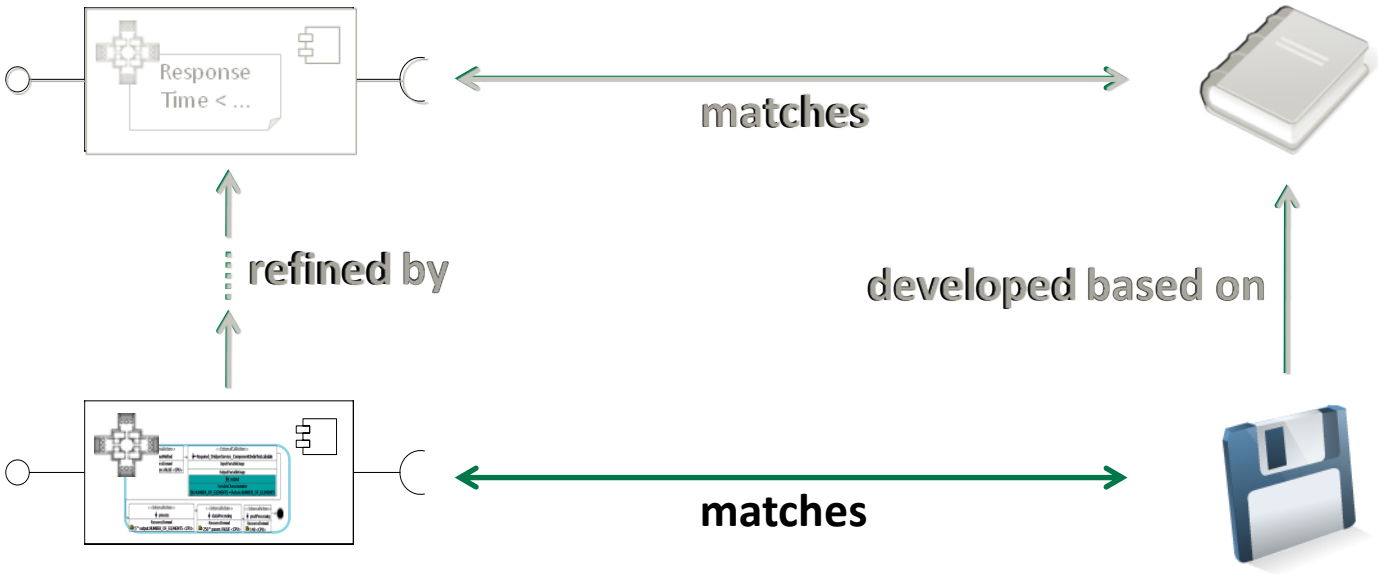
Usage Profile and Platform Independent Automated Validation of Service Behavior Specifications

Henning Groenda,
QUASOSS,
04.10.2010

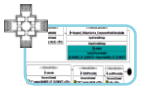
Context
Round-trip Performance Engineering



Legend

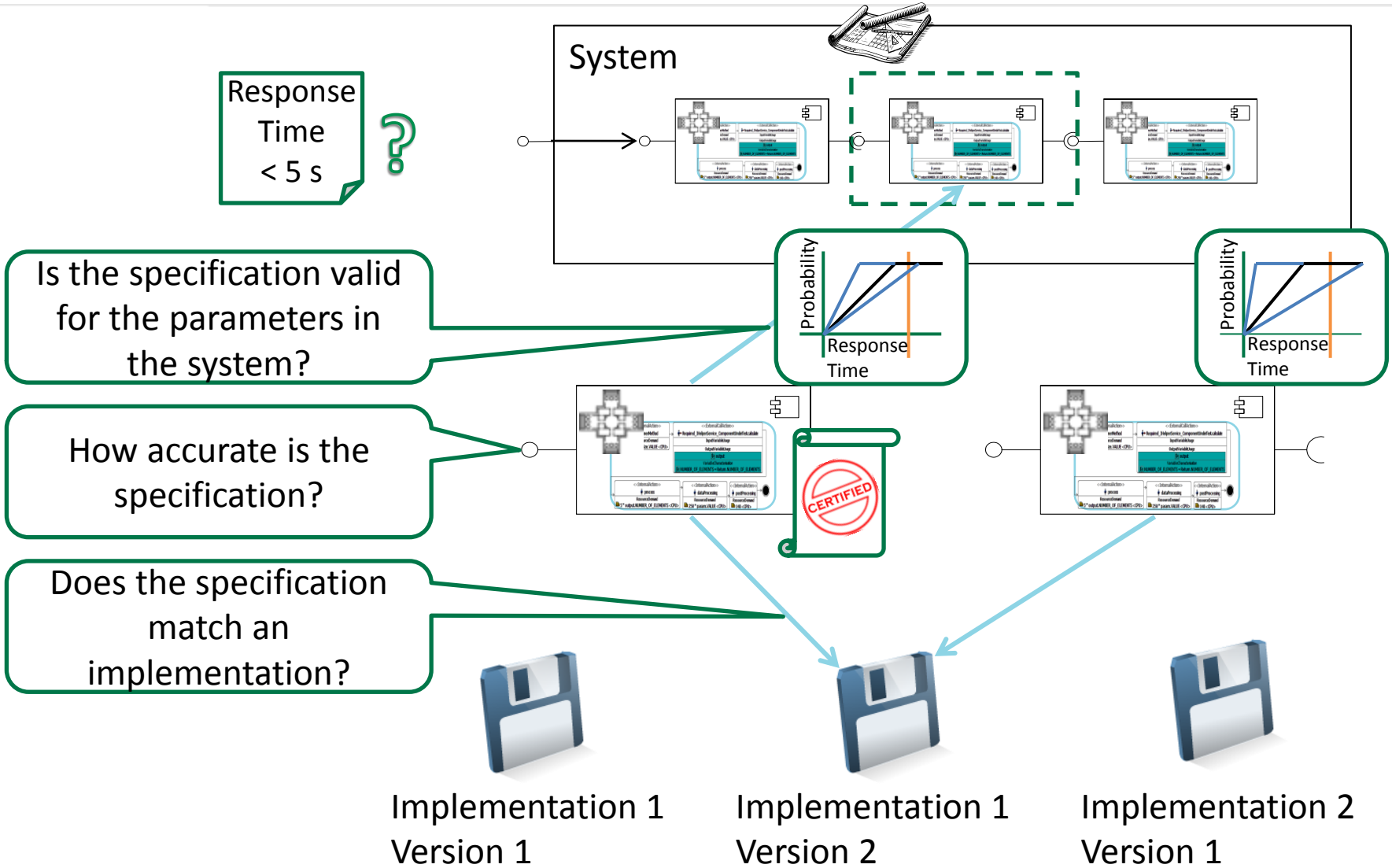

Requirements


Implementation

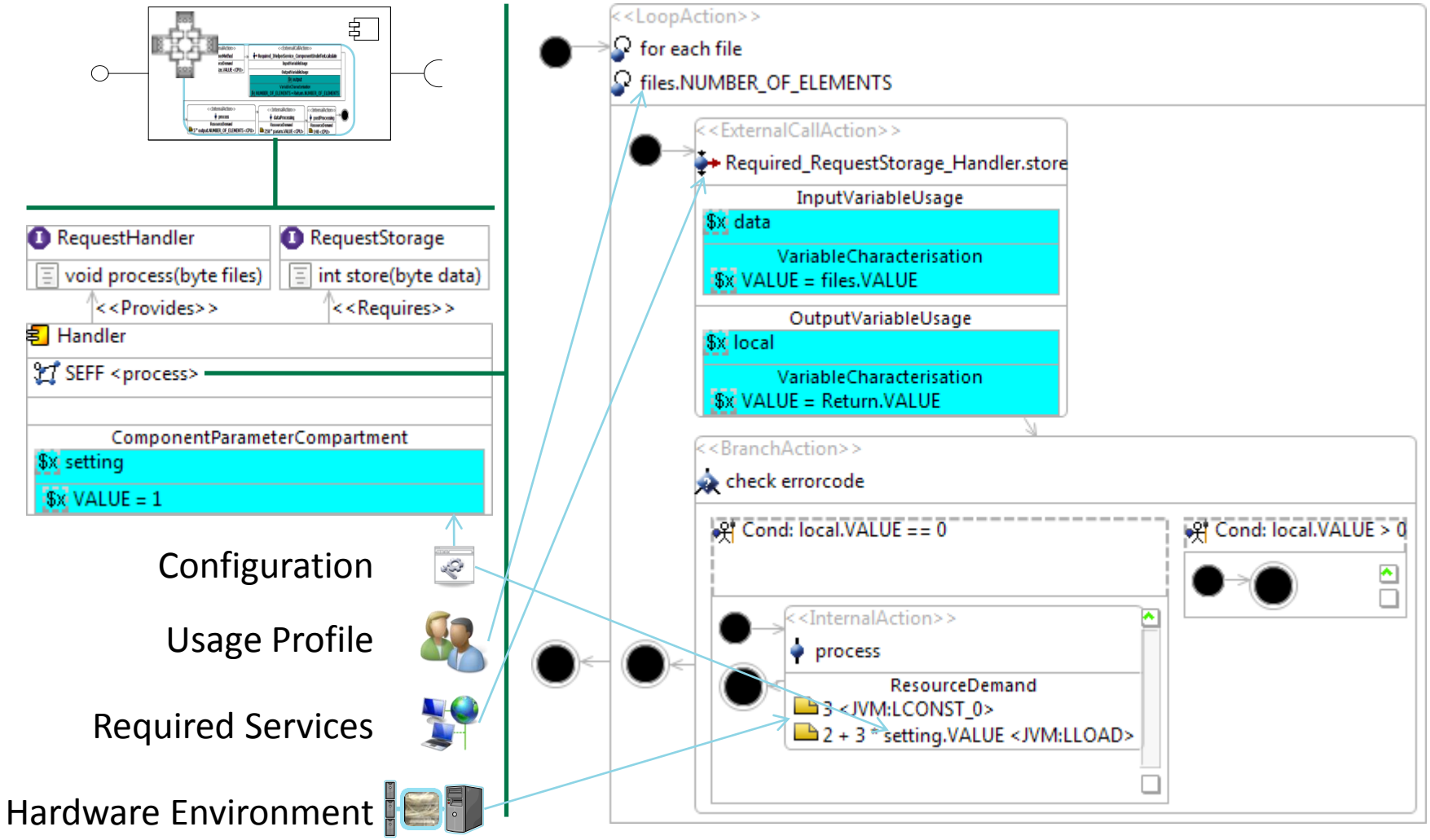

Descriptive Performance Specifications

Normative Performance Specification

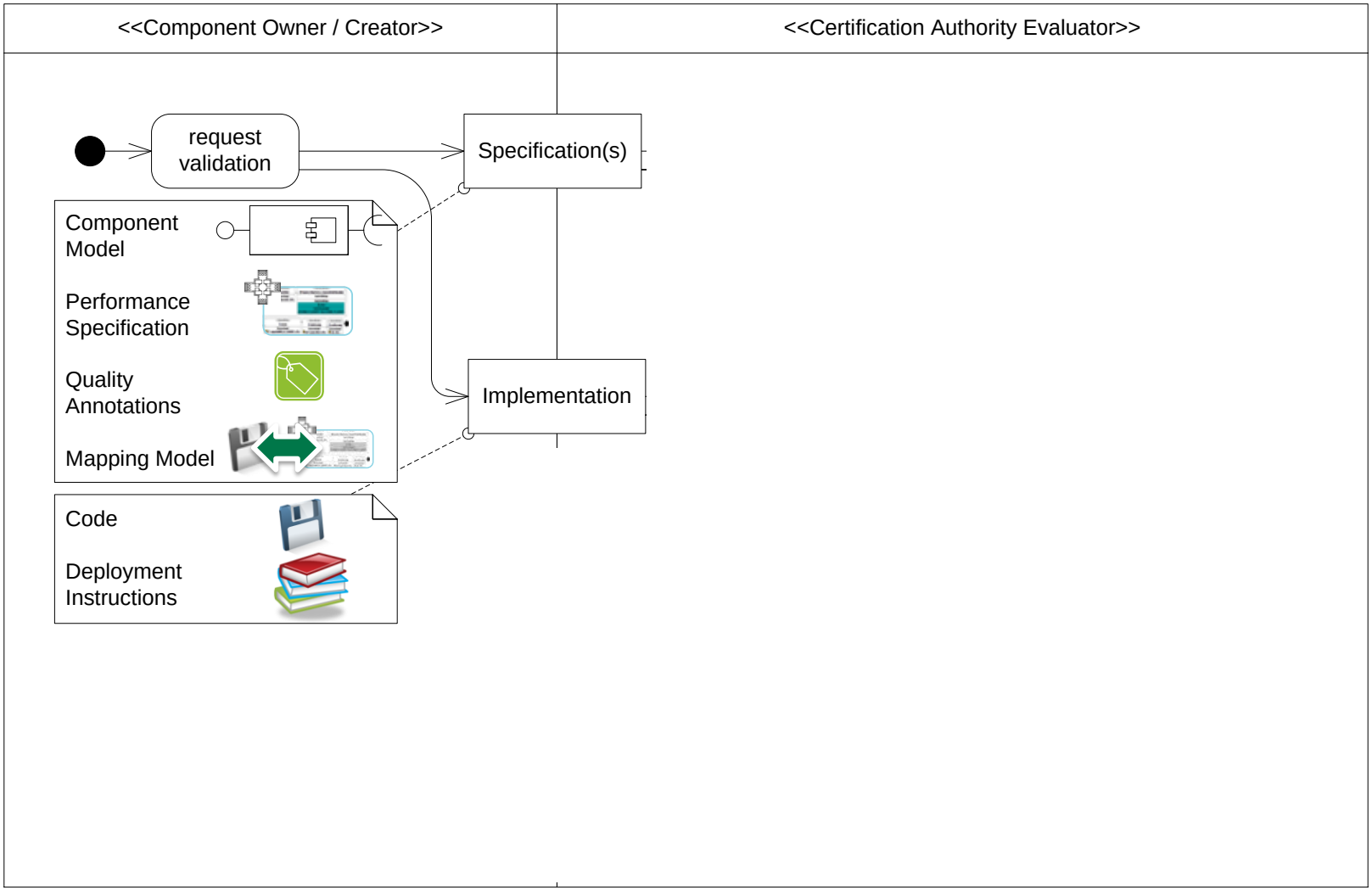
Motivation


System
Response Time < 5 s
Is the specification valid for the parameters in the system?
How accurate is the specification?
Does the specification match an implementation?
Probability
Response Time
Probability
Response Time
CERTIFIED
Implementation 1 Version 1
Implementation 1 Version 2
Implementation 2 Version 1

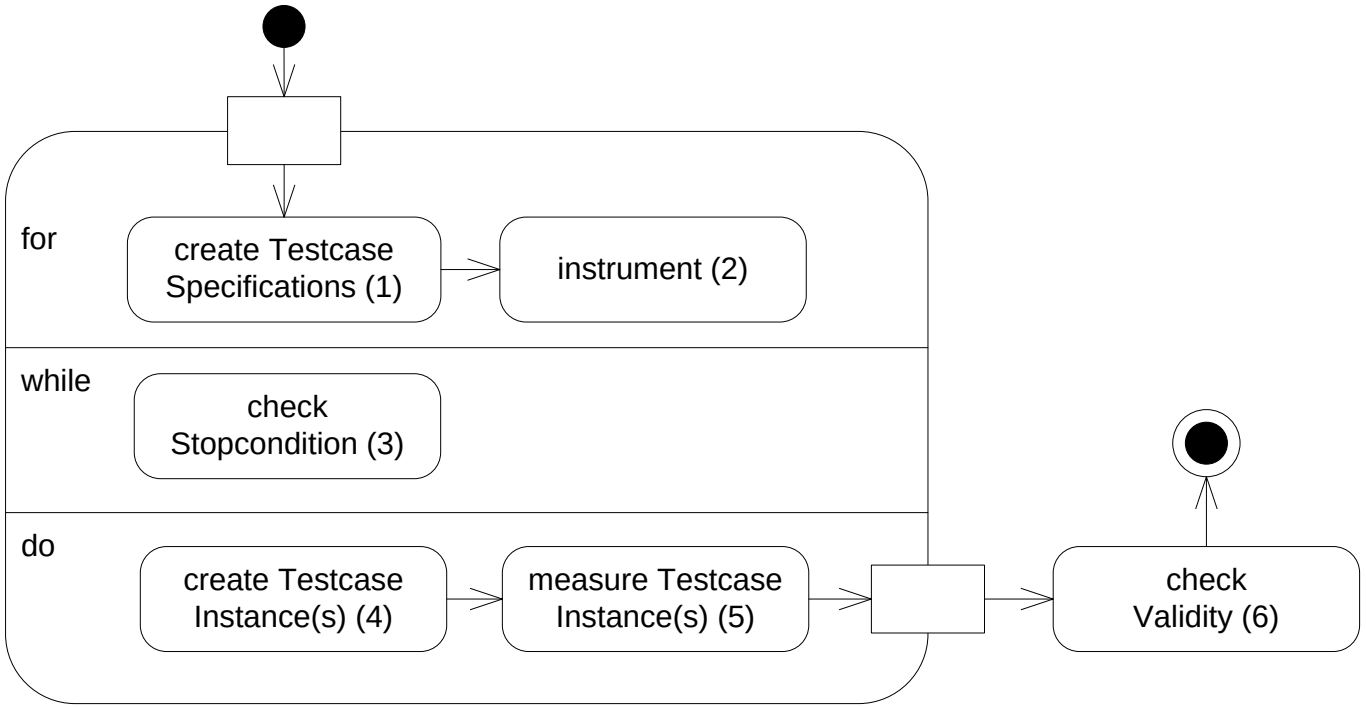
Parameterized Specification Example



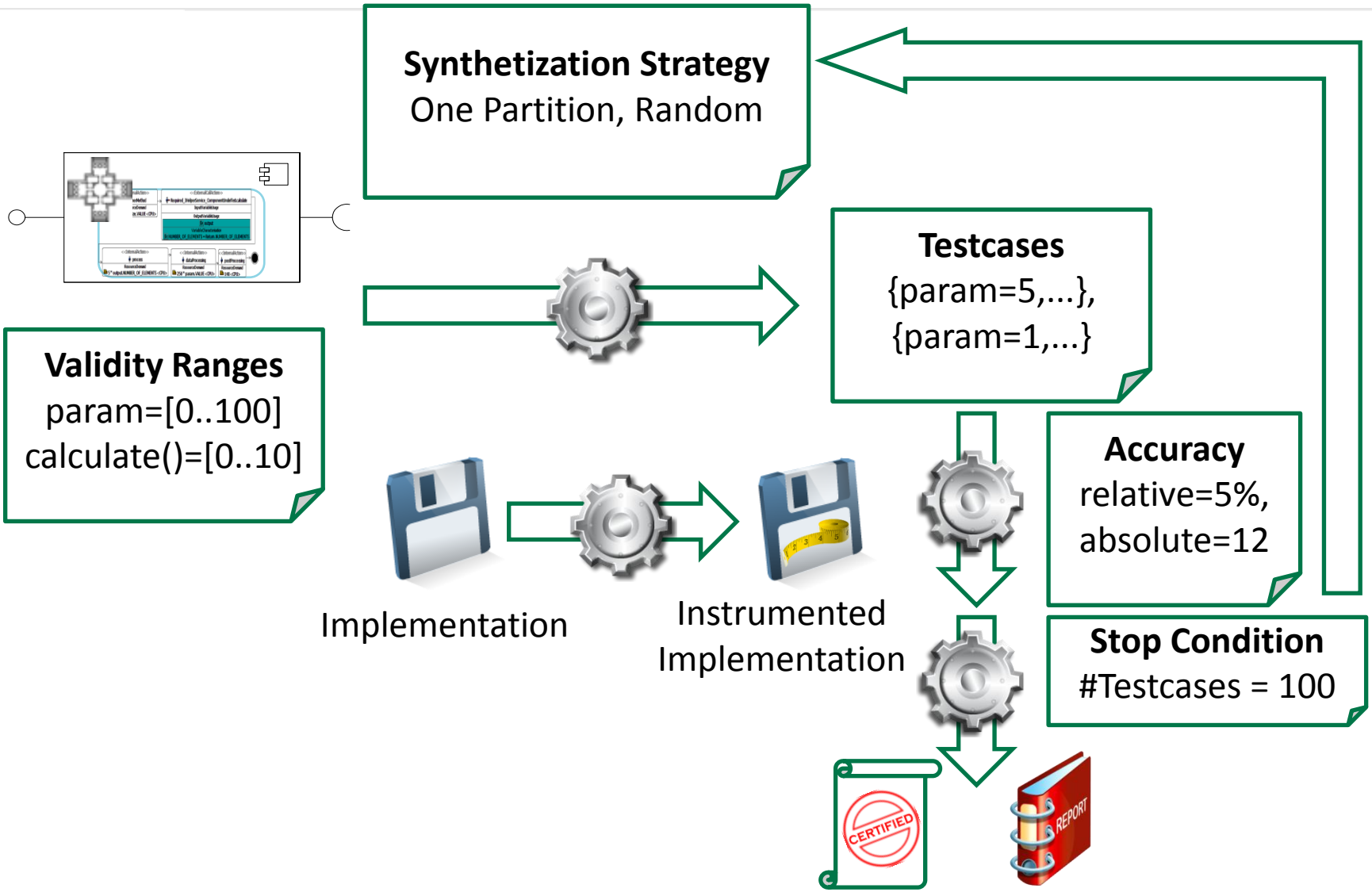
Specification Validation Process



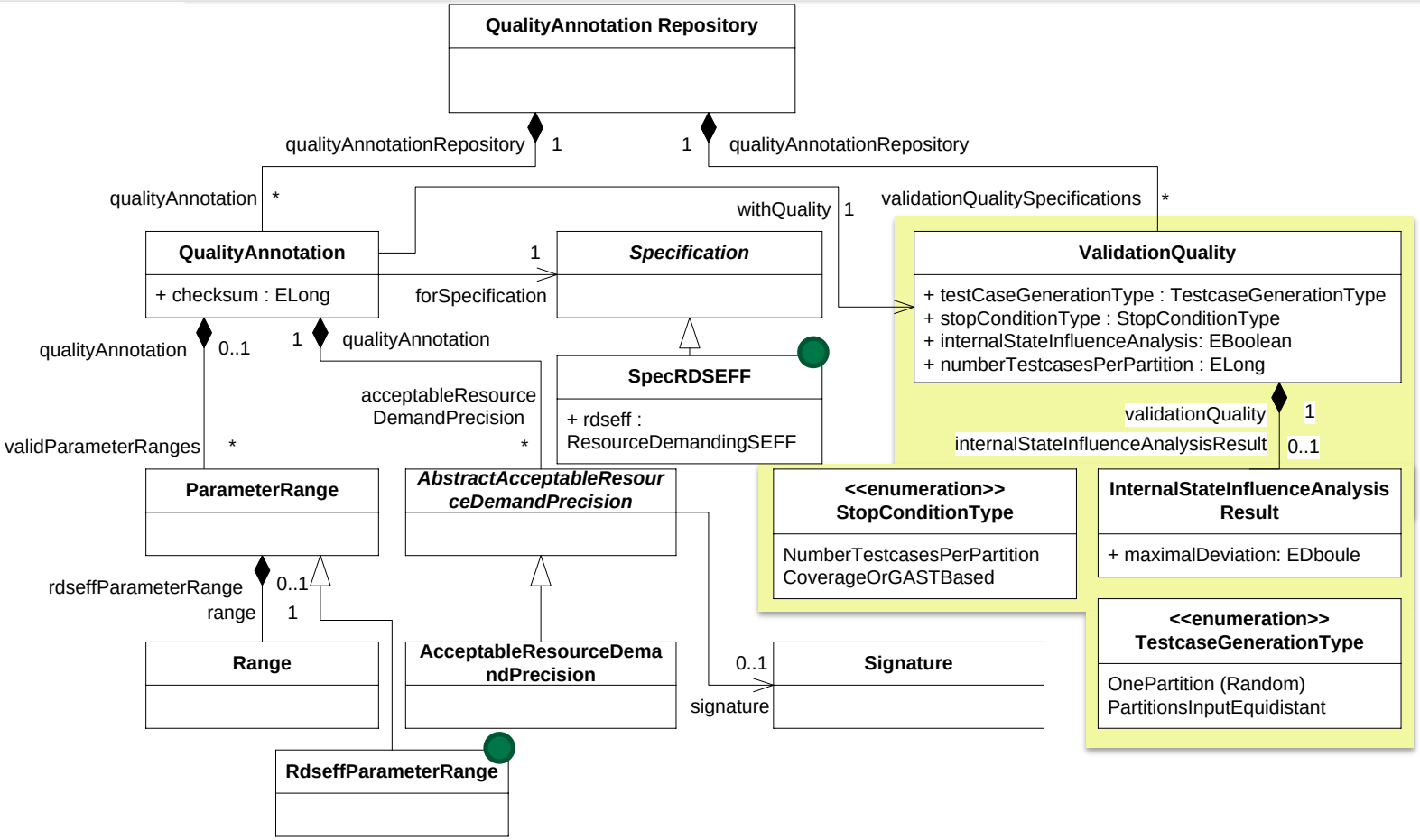
Assessment Step Activity Diagram



Validation Process Example



Accuracy Information Annotation Model (AIAM)



Quality Annotation (for PCM)

Validation Quality

Validation

- Accuracy of validation:
  Performance validation based on Bytecode

- Instrumentation not mixed with source code
  Use annotation model

- Mapping specification elements and implementation
  Use General Abstract Syntax Tree (GAST)

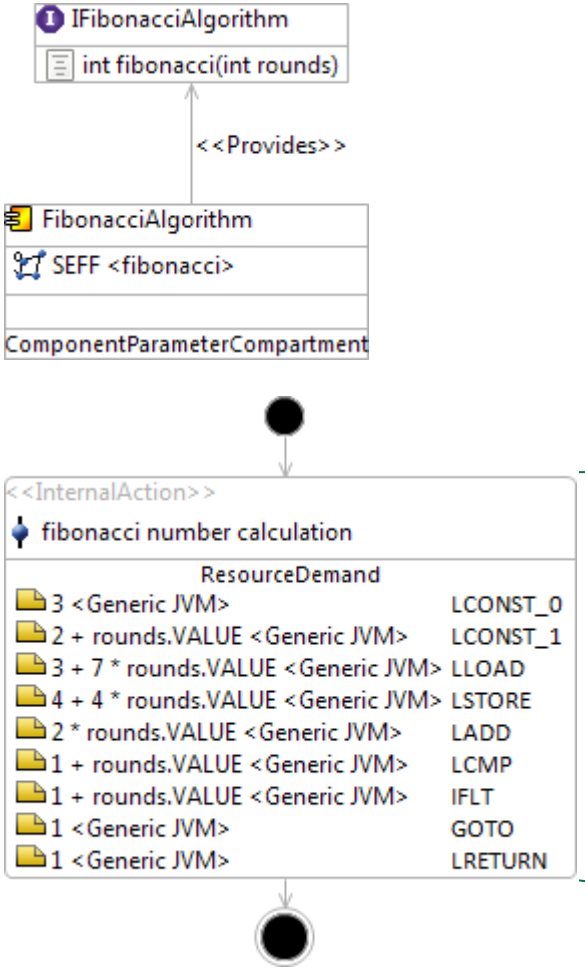
Implementation

```
public class FibonacciAlgorithm {

	/**Calculates a Fibonacci number given the number
of rounds the algorithm should run.
	 * @param rounds Number of calculation rounds.
	 * @return Fibonacci number.
	 */
	public long fibonacci(long rounds) {
		long i1 = 0;
		long i2 = 1;
		long i3 = 0;
		// normalized loop
		for (
				long i = 0;
				i < rounds;
				i++) {
			i3 = i1 + i2;
			i2 = i1;
			i1 = i3;
		}
		return i3;
	}
}
```

GAST

- Package de
  - Package de.fzi
    - Package de.fzi.ByCounter
      - Package de.fzi.ByCounter.example
        - Package de.fzi.ByCounter.example.Fibonacci
          - GAST Class de.fzi.ByCounter.example.Fibonacci.FibonacciAlgorithm
            - Comment _-I8V8UYUEd-iYb1IWQK_sg
            - Position FILE:C:
            - Constructor _-h4VUEYUEd-iYb1IWQK_sg
            - Method fibonacci(long):long
              - Comment _-I8V8kYUEd-iYb1IWQK_sg
              - Comment _-I8V80YUEd-iYb1IWQK_sg
              - Position FILE:C:
              - Block Statement _-iKpQ0YUEd-iYb1IWQK_sg
              - Local Variable _-jjwZEYUEd-iYb1IWQK_sg
              - Local Variable _-jjwZkYUEd-iYb1IWQK_sg
              - Local Variable _-jjwaEYUEd-iYb1IWQK_sg
              - Local Variable _-jjwakYUEd-iYb1IWQK_sg
              - Declaration Type Access _-h3uR0YUEd-iYb1IWQK_sg long
              - Formal Parameter _-jkXgEYUEd-iYb1IWQK_sg
            - Inheritance Type Access _-kWalkYUEd-iYb1IWQK_sg java.lang.Object

Case Study
Specification, Quality, and Mapping



Internal State Influence Analysis	false
Number Testcases Per Partition	100
Stop Condition Type	NumberTestcasesPerPartition
Test Case Generation Type	OnePartitionRandom
Acceptable Absolute Deviance	0
Acceptable Relative Deviance	0.0

Parameter	Parameter rounds
Resource Demanding SEFF	Resource Demanding SEFF _VNfWQDD7Ed-JI8LGk6UTnA
From	0
To	1000

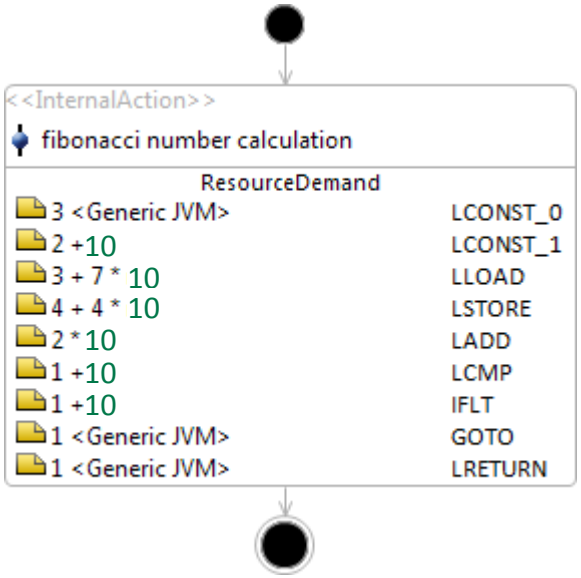
```
public class FibonacciAlgorithm {

    /**Calculates a Fibonacci number given the number of rounds the algorithm should run.
     * @param rounds Number of calculation rounds.
     * @return Fibonacci number.
     */
    public long fibonacci(long rounds) {
        long i1 = 0;
        long i2 = 1;
        long i3 = 0;
        // normalized loop
        for (
                long i = 0;
                i < rounds;
                i++) {
            i3 = i1 + i2;
            i2 = i1;
            i1 = i3;
        }
        return i3;
    }
}
```

Case Study
Validity Check Example

Parameter

- rounds: 10

Specification



Measurement

3	LCONST_0
11	LCONST_1
73	LLOAD
44	LSTORE
20	LADD
11	LCMP
11	IFLT
1	GOTO
1	LRETURN

Case Study Runtime

- No need to compensate warmup and teardown effects
- No need to identify and exclude outliers

rounds	Response Time (ns) pure	Response Time (ns) instrumented
1	6 161	141 197 445
10	3 920	105 979 160
100	3 360	105 463 856
1000	10 082	137 462 045

- Significant overhead for validation
  - Vs. re-use of specifications
  - Vs. creating a new specification for each use-case
  - Vs. integrating and measuring a service in a test system

Assumptions & Limitations

- Focus on
  - Business services
  - CPU resource demand

- Internal state analysis not implemented yet
- Runtime-weaving is not supported yet
- Currently supports validation of internal actions

Conclusion Future Work

- Automated validation of performance specifications
  - Exact measurements without disturbances
  - Validation of parameterized specifications
  - Support for fine-grained regions of validation
  - Supports validation of more than one method
  - Documented specification quality: Trust, error propagation

- Future Work
  - Support the use of required services
  - Improved test case synthetization strategies
    - Block coverage (on specification)
    - Condition coverage (on specification)

BACKUP

Next Steps

- Theory
  - WP 5: Test Strategies for Performance Specifications
    - Overview on control-flow based strategies
    - Discoverable error types on functional level
    - Semantic translation to performance specifications
    - Value of applicability on performance specifications

- Testbed
  - Support for external calls
  - Support for component parameters
  - Support for complex parameters
  - Support for parameter characterizations
  - Dynamic testcase synthetization (pre- vs on-demand-creation)

Example Input

Code

```
13 public class FibonacciAlgorithm {
14
15     /**Calculates a Fibonacci number given the number of rour
16      * @param rounds Number of calculation rounds.
17      * @return Fibonacci number.
18      */
19     public long fibonacci(long rounds) {
20         long i1 = 1;
21         long i2 = 1;
22         long i3 = 0;
23         // normalized loop
24         for (
25                 long i = 0;
26                 i < rounds;
27                 i++) {
28             i3 = i1 + i2;
29             i2 = i1;
30             i1 = i3;
31         }
32         return i3;
33     }
34 }
```

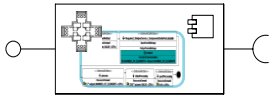
GAST

- Package de
  - Package de.fzi
    - Package de.fzi.ByCounter
      - Package de.fzi.ByCounter.example
        - Package de.fzi.ByCounter.example.Fibonacci
          - GAST Class de.fzi.ByCounter.example.Fibonacci.RunExample
          - GAST Class de.fzi.ByCounter.example.Fibonacci.FibonacciAlgorithm
            - Comment _-l8V8UYUEd-iYb1IWQK_sg
            - Position FILE:C:\\data\\wrk\\Projekte\\Part III - Certification of Performance Specifications\\ru
            - Constructor _-h4VUEYUEd-iYb1IWQK_sg
            - Method fibonacci(long):long
              - Comment _-l8V8kYUEd-iYb1IWQK_sg
              - Comment _-l8V80YUEd-iYb1IWQK_sg
              - Position FILE:C:\\data\\wrk\\Projekte\\Part III - Certification of Performance Specifications\
              - Block Statement _-iKpQ0YUEd-iYb1IWQK_sg
              - Local Variable _-jjwZEYUEd-iYb1IWQK_sg
              - Local Variable _-jjwZkYUEd-iYb1IWQK_sg
              - Local Variable _-jjwaEYUEd-iYb1IWQK_sg
              - Local Variable _-jjwakYUEd-iYb1IWQK_sg
              - Declaration Type Access _-h3uR0YUEd-iYb1IWQK_sg long
              - Formal Parameter _-jkXgEYUEd-iYb1IWQK_sg
            - Inheritance Type Access _-kWalkYUEd-iYb1IWQK_sg java.lang.Object

Example Input

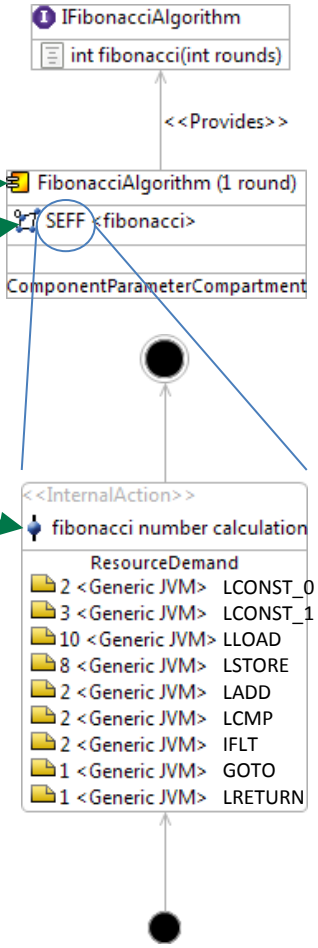
GAST Model

Link Model

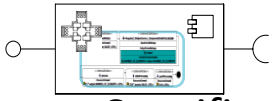


Performance Specification Model

- Package de
  - Package de.fzi
    - Package de.fzi.ByCounter
      - Package de.fzi.ByCounter.example
        - Package de.fzi.ByCounter.example.Fibonacci
          - GAST Class de.fzi.ByCounter.example.Fibonacci.RunExample
          - GAST Class de.fzi.ByCounter.example.Fibonacci.FibonacciAlgorithm
            - Comment _-l8V8UYUEd-iYb1IWQK_sg
            - Position FILE:C:\\data\\wrk\\Projekte\\Part III - Certification of Performance Specifications\\ru
            - Constructor _-h4VUEYUEd-iYb1IWQK_sg
            - Method fibonacci(long):long
              - Comment _-l8V8kYUEd-iYb1IWQK_sg
              - Comment _-l8V80YUEd-iYb1IWQK_sg
              - Position FILE:C:\\data\\wrk\\Projekte\\Part III - Certification of Performance Specifications'
              - Block Statement _-iKpQ0YUEd-iYb1IWQK_sg
              - Local Variable _-jjwZEYUEd-iYb1IWQK_sg
              - Local Variable _-jjwZkYUEd-iYb1IWQK_sg
              - Local Variable _-jjwaEYUEd-iYb1IWQK_sg
              - Local Variable _-jjwakYUEd-iYb1IWQK_sg
              - Declaration Type Access _-h3uR0YUEd-iYb1IWQK_sg long
              - Formal Parameter _-jkXgEYUEd-iYb1IWQK_sg
            - Inheritance Type Access _-kWalkYUEd-iYb1IWQK_sg java.lang.Object

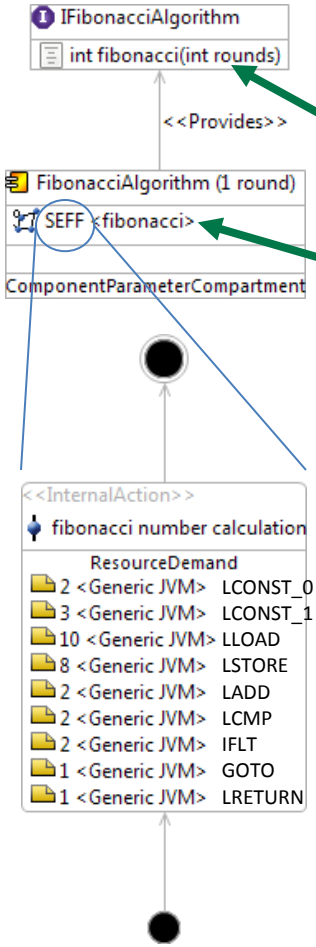


Example
Quality Annotations



Performance Specification Model

Quality Annotation



Validation Quality

Internal State Influence Analysis	false
Number Testcases Per Partition	100
Stop Condition Type	NumberTestcasesPerPartition
Test Case Generation Type	OnePartitionRandom
Acceptable Absolute Deviance	0
Acceptable Relative Deviance	0.0

Quality Annotation

For Rdseff	Resource Demanding SEFF _VNfWQDD7Ed-JI8LGk6UTnA
Parameter	Parameter rounds
Resource Demanding SEFF	Resource Demanding SEFF _VNfWQDD7Ed-JI8LGk6UTnA
From	0
Id	_mSuH8UlcEd-_Vd2k9TvUbw
To	1000

Example
Test case partitions / Test case instance

Validation Quality

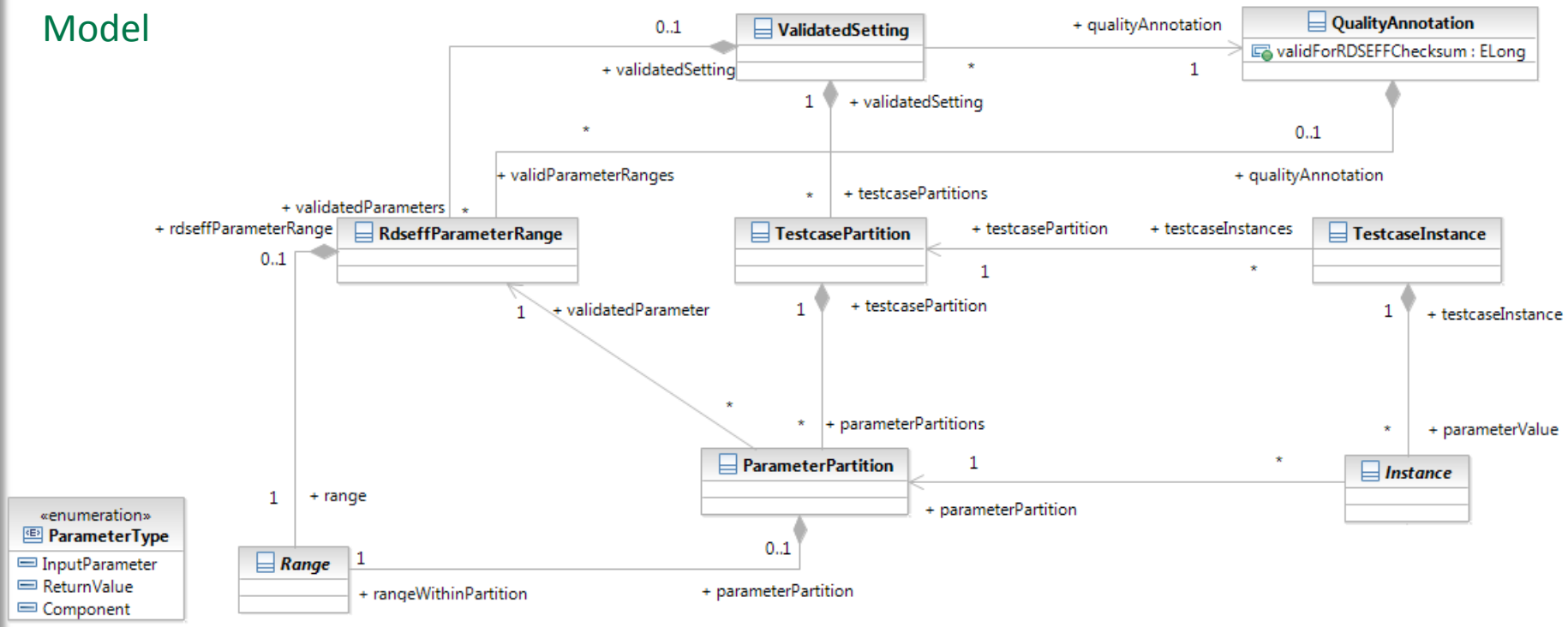
Internal State Influence Analysis	false
Number Testcases Per Partition	100
Stop Condition Type	NumberTestcasesPerPartition
Test Case Generation Type	OnePartitionRandom
Acceptable Absolute Deviance	0
Acceptable Relative Deviance	0.0

parameter 2

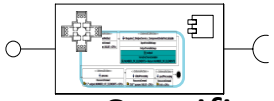
parameter 1	partition 1 [10,20]		partition 2 [30,40]	
partition 1 [0,5]	TCP ([0,5],[10,20])	TCI (3,10) TCI (1,17)	TCP ([0,5],[30,40])	TCI (4,33) TCI (3,36)
partition 2 [5,10]	TCP ([5,10],[10,20])	TCI (5,11) TCI (9,13)	TCP ([5,10],[30,40])	TCI (5,30) TCI(6,38)

Example

Model



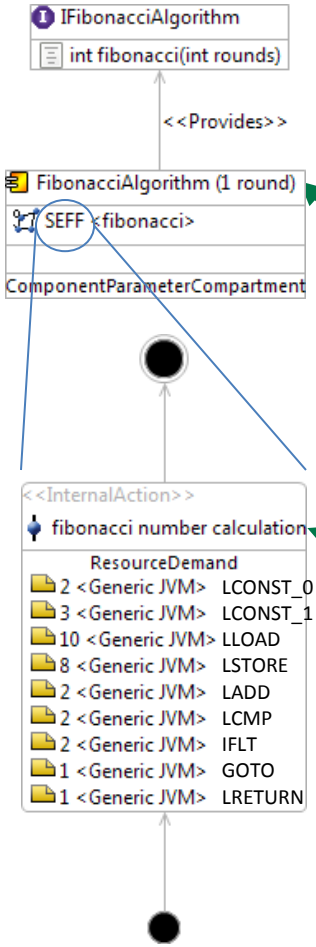
Example Instrumentation



Performance Specification Model

Link Model

Instrumentation Model



Instrumentation Profile

Aggregate Internal Calls Transparently	true
Aggregation Type	ONLINE
Id	_F6_VsFRfEd-rhZ7fUacBAw
Instrument Recursively	true
Instrument Using Basic Blocks	true
Instrument Using Long Counters	true
Persist Instrumented Classes	
Temporary Results Type	IN_MEMORY
Trace And Identify Requests	false

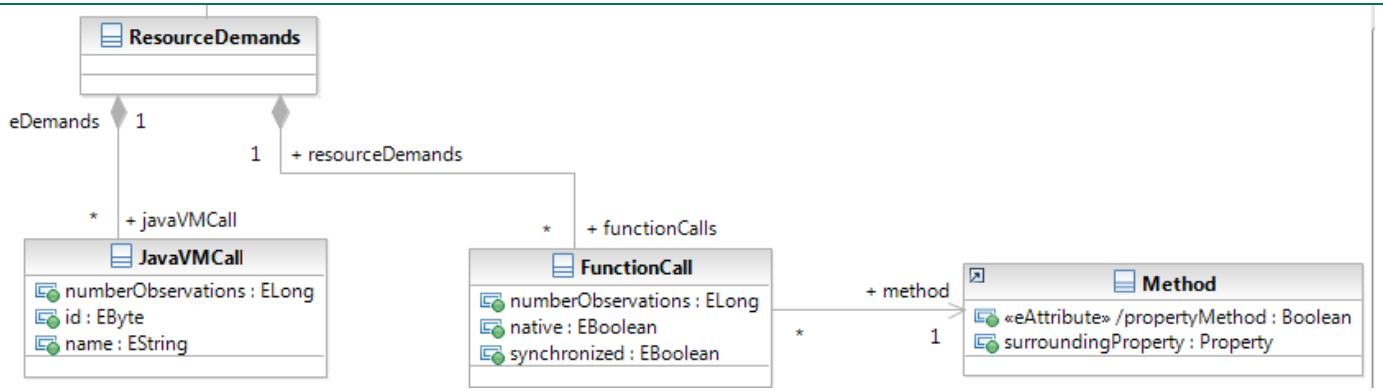
- Logical Set _yVGYIFdnEd-Gwt9_2ESV5g
  - GAST Class de.fzi.ByCounter.example.Fibonacci.FibonacciAlgorithm
- Instrumented Code Area _yYx-QFdnEd-Gwt9_2ESV5g

Example
Test case execution result

2x LCONST_0
3x LCONST_1
10x LLOAD
8x LSTORE
2x LADD
2x LCMP
2x IFLT
1x GOTO
1x LRETURN

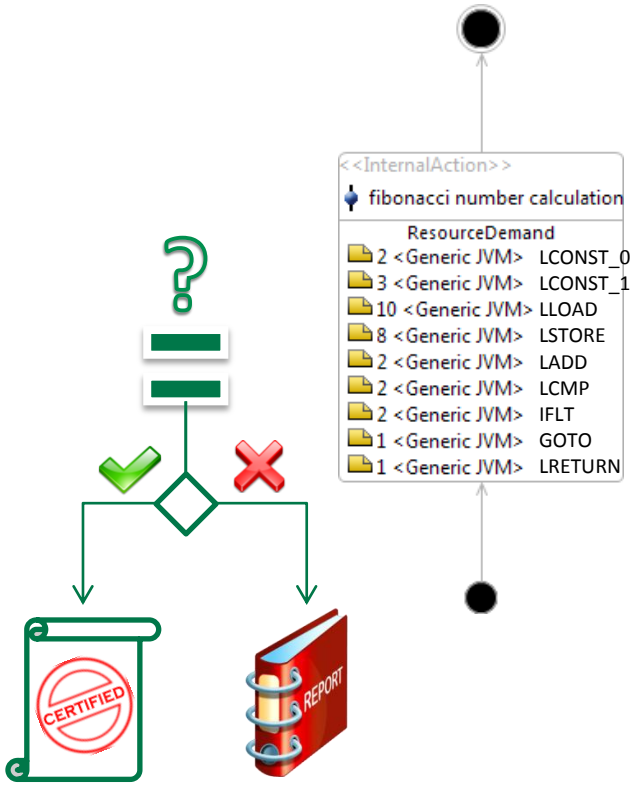
Example

Model



Example Comparison

2x LCONST_0
3x LCONST_1
10x LLOAD
8x LSTORE
2x LADD
2x LCMP
2x IFLT
1x GOTO
1x LRETURN

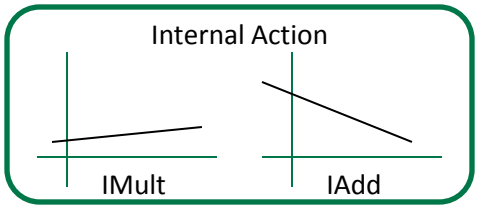


Next Steps

- Additional parameter types
  - Component
  - Return-value of external calls
- Parameter characterizations
  - Mapping of technical and performance-relevant interface
  - Structural-based information, e.g. nodes and edges of graph
  - Non-structural-based information, e.g. compression or clustering ratio
  - Support for synthetization and characterization detection
- Internal state
  - Detect size of internal state influence: Evaluate by re-executing with the same parameters but in other order.

Next Steps

- Testcase generation
  - Random
  - Equidistant partitions (input or output)
  - Output partitions with equal change
    - Linear
    - Integral
  - Weight depending on magnitude of change
  - Based on implementation analysis



- Stop-Conditions
  - Number of test cases per partition
  - Choose Metric, e.g. Coverage-/GAST-Analysis
    - Estimated error probability / confidence

Next Steps

- Case Study: Fibonacci algorithm
  - Evaluate for all values within range; Extend range until validation fails
  - Measure effort, show chosen values
  - Random vs. equidistant input partitions

Challenges

- How to validate parameterizes specifications which are based on control-flow abstractions?
- How can usage-profile and hardware environment independent validations be supported?
- How can the quality of a specification be quantified? Which metrics are useful?
- How can the quality of the validation of a specification be evaluated? Which test case generation strategies and stop conditions allow to cover which error types?
- How can certification be integration into a component-based development process?
- Which instrumentation granularity should be chosed to validate parameterized specifications?