

SoftTest: A Framework for **Software** **Testing** of Java Programs

B. Childers, M. L. Soffa, J. Beaver, L. Ber, K.
Cammarata, J. Litman, J. Misurda



Department of Computer Science
University of Pittsburgh
<http://www.cs.pitt.edu/copa>

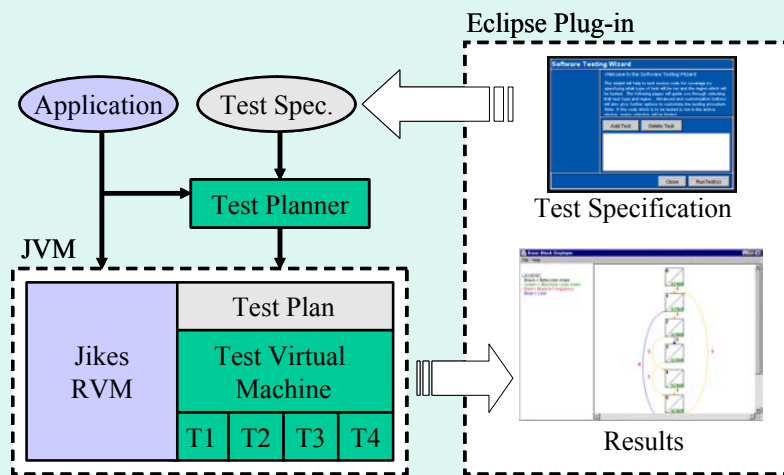
Structural Software Testing

- Assure quality, robust software
 - Collect coverage information about the program
 - E.g., branch coverage records which edges are hit
 - Uses program instrumentation
- Static instrumentation approaches
 - Not scalable: Instrumentation left in place
 - Inflexible: Only certain tests, languages, platforms

Goals and Approach

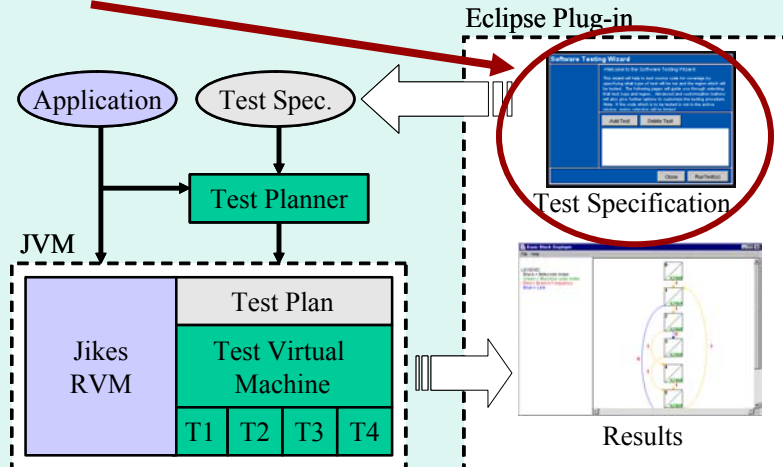
- A scalable & flexible framework for structural testing
 - Automatically apply multiple test strategies
 - Multiple languages and platforms
 - Handle large programs
- Approach
 - **Specification driven:** User written test
 - **Test plans:** Recipe of how & where to test
 - **Path specific:** Instrument only what is needed
 - **Dynamic:** Insert & remove instrumentation

Test Framework Instance for Java



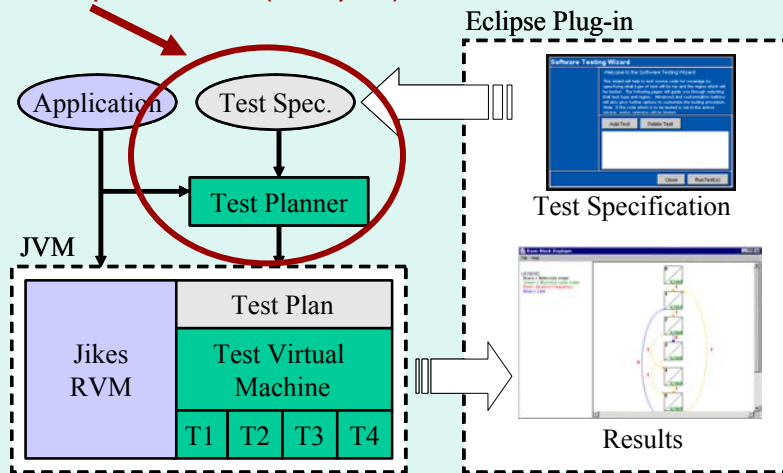
Test Framework Instance for Java

Wizard for creating the test specification



Test Framework Instance for Java

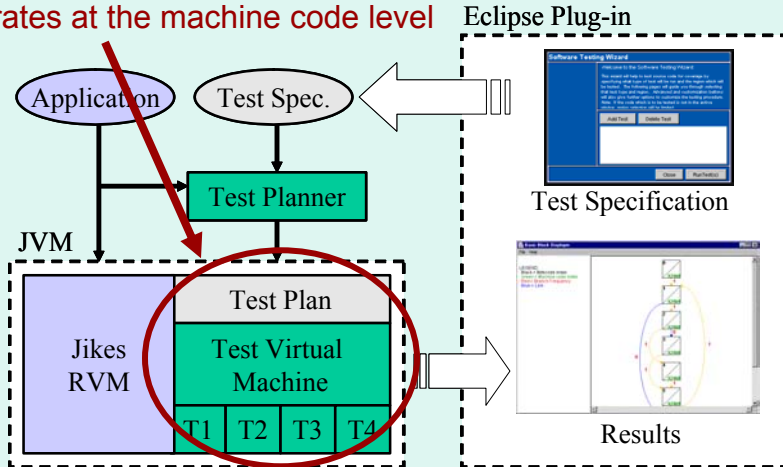
From test specification (*testspec*) determine how to test



Test Framework Instance for Java

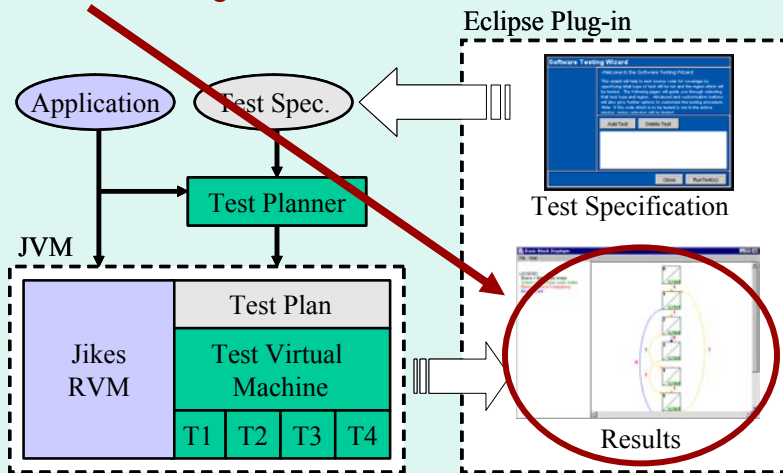
Test plan uses *instrumentation primitives*

Operates at the machine code level



Test Framework Instance for Java

Source level coverage results



Test Planner

- **Test plan:** Where and how to test program
- **Test table**
 - Where to instrument in the target machine code
 - Test information (results)
 - Branch coverage: when to insert & remove instrumentation, records which edges are hit/not hit
- **Test payload**
 - What to do at an instrumentation point
 - Target machine code
 - Branch coverage: Update edge coverage

Branch Coverage Example

Test Table

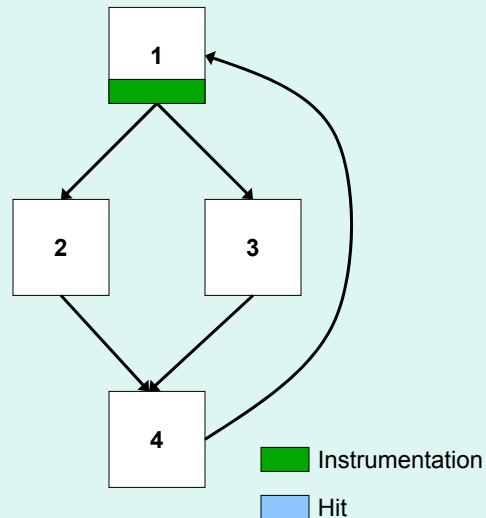
<u>Block</u>	<u>Next</u>	<u>Hit</u>
1	2,3	N,N
2	4	N
3	4	N
4	1	N

Test Payload

Mark edge hit

Insert at next point

Remove instrumentation



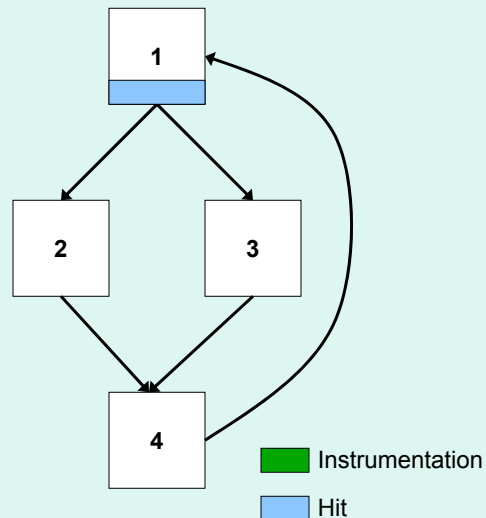
Branch Coverage Example

Test Table

<u>Block</u>	<u>Next</u>	<u>Hit</u>
1	2,3	N,N
2	4	N
3	4	N
4	1	N

Test Payload

Mark edge hit
Insert at next point
Remove instrumentation



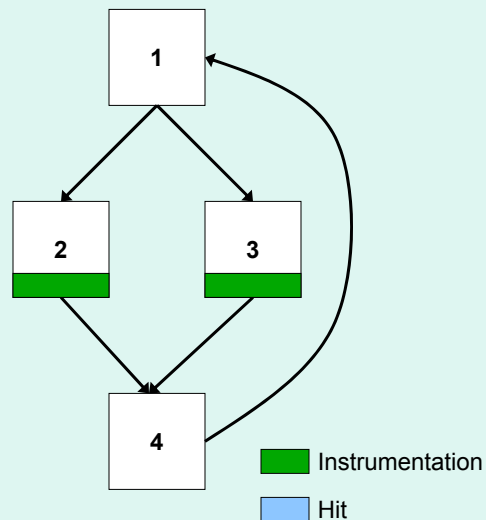
Branch Coverage Example

Test Table

<u>Block</u>	<u>Next</u>	<u>Hit</u>
1	2,3	N,N
2	4	N
3	4	N
4	1	N

Test Payload

Mark edge hit
Insert at next point
Remove instrumentation



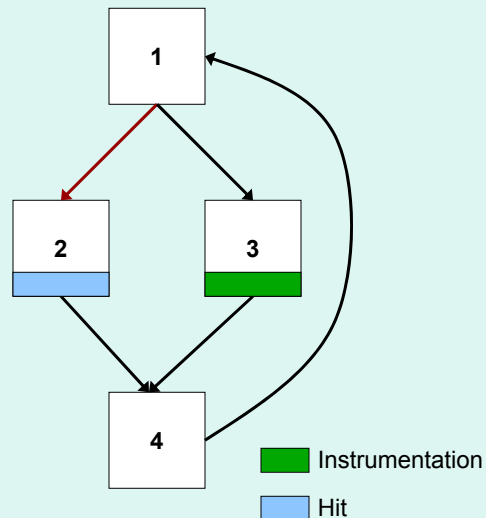
Branch Coverage Example

Test Table

<u>Block</u>	<u>Next</u>	<u>Hit</u>
1	2,3	Y,N
2	4	N
3	4	N
4	1	N

Test Payload

Mark edge hit
Insert at next point
Remove instrumentation



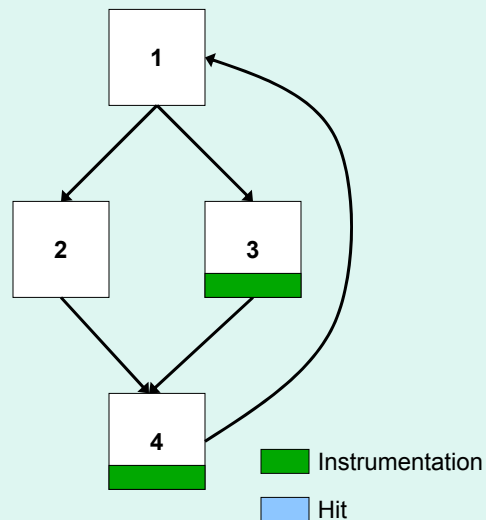
Branch Coverage Example

Test Table

<u>Block</u>	<u>Next</u>	<u>Hit</u>
1	2,3	Y,N
2	4	N
3	4	N
4	1	N

Test Payload

Mark edge hit
Insert at next point
Remove instrumentation



Branch Coverage Example

Test Table

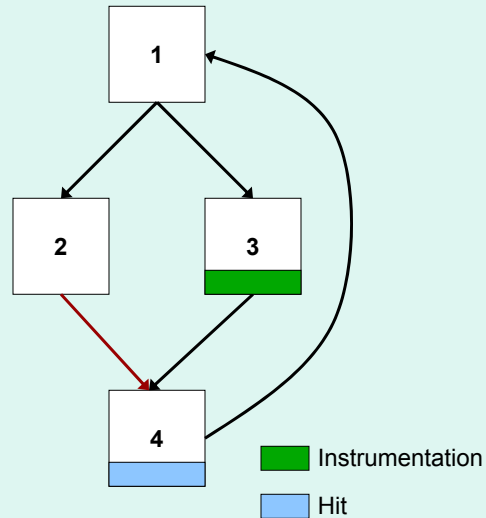
<u>Block</u>	<u>Next</u>	<u>Hit</u>
1	2,3	Y,N
2	4	Y
3	4	N
4	1	N

Test Payload

Mark edge hit

Insert at next point

Remove instrumentation



Branch Coverage Example

Test Table

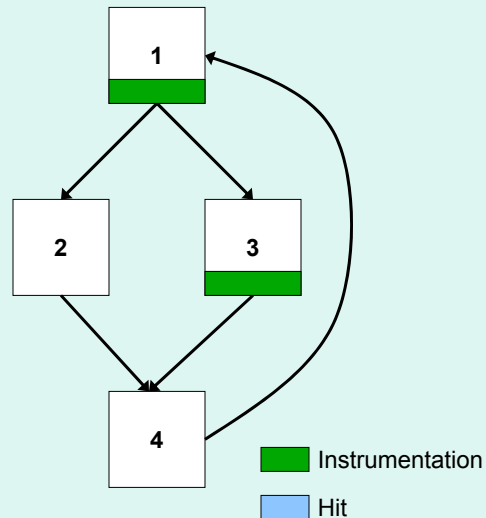
<u>Block</u>	<u>Next</u>	<u>Hit</u>
1	2,3	Y,N
2	4	Y
3	4	N
4	1	N

Test Payload

Mark edge hit

Insert at next point

Remove instrumentation



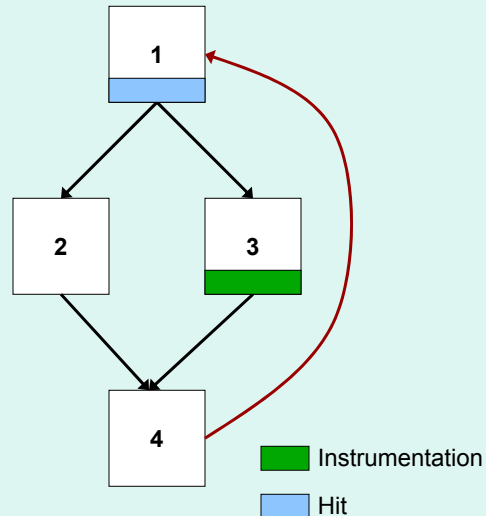
Branch Coverage Example

Test Table

<u>Block</u>	<u>Next</u>	<u>Hit</u>
1	2,3	Y,N
2	4	Y
3	4	N
4	1	Y

Test Payload

Mark edge hit
Insert at next point
Remove instrumentation



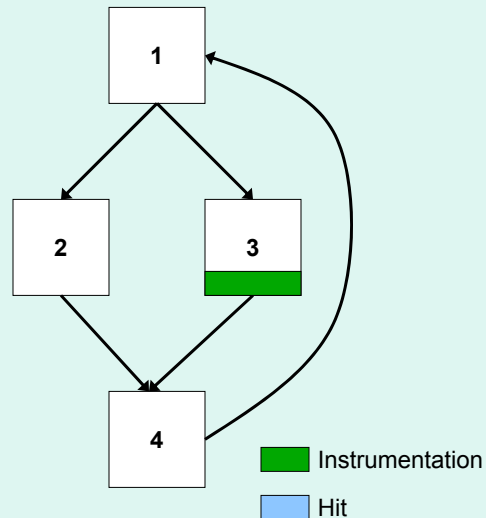
Branch Coverage Example

Test Table

<u>Block</u>	<u>Next</u>	<u>Hit</u>
1	2,3	Y,N
2	4	Y
3	4	N
4	1	Y

Test Payload

Mark edge hit
Insert at next point
Remove instrumentation



Branch Coverage Example

Test Table

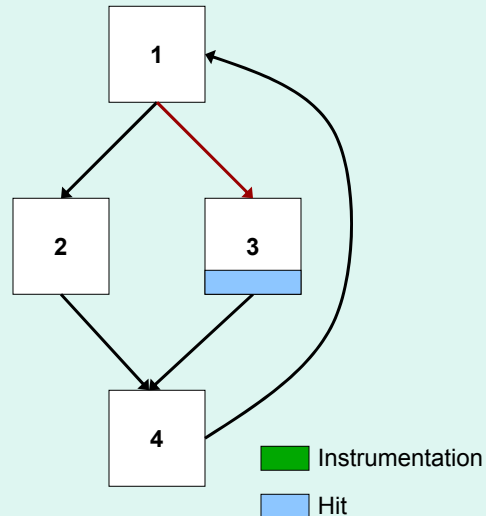
<u>Block</u>	<u>Next</u>	<u>Hit</u>
1	2,3	Y,Y
2	4	Y
3	4	N
4	1	Y

Test Payload

Mark edge hit

Insert at next point

Remove instrumentation



Branch Coverage Example

Test Table

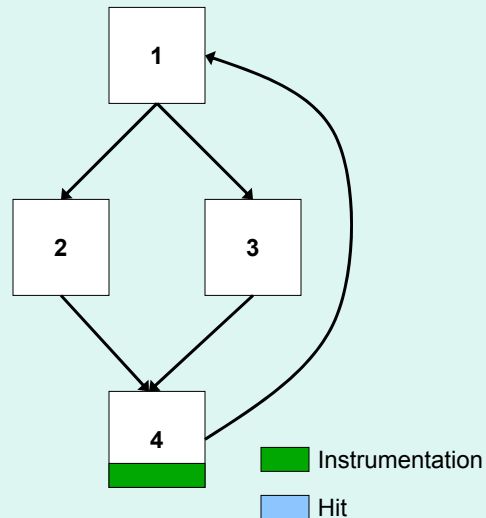
<u>Block</u>	<u>Next</u>	<u>Hit</u>
1	2,3	Y,Y
2	4	Y
3	4	N
4	1	Y

Test Payload

Mark edge hit

Insert at next point

Remove instrumentation



Branch Coverage Example

Test Table

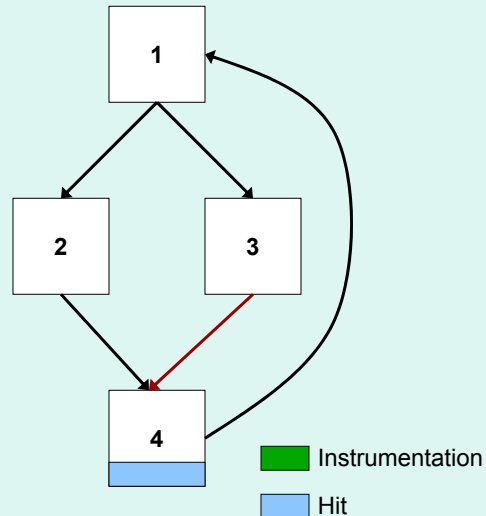
<u>Block</u>	<u>Next</u>	<u>Hit</u>
1	2,3	Y,Y
2	4	Y
3	4	Y
4	1	Y

Test Payload

Mark edge hit

Insert at next point

Remove instrumentation



Branch Coverage Example

Test Table

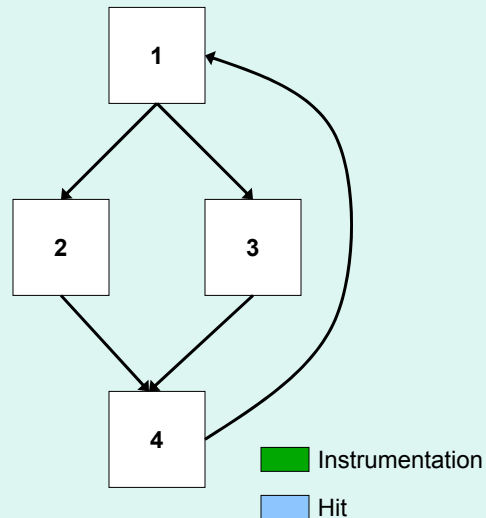
<u>Block</u>	<u>Next</u>	<u>Hit</u>
1	2,3	Y,Y
2	4	Y
3	4	Y
4	1	Y

Test Payload

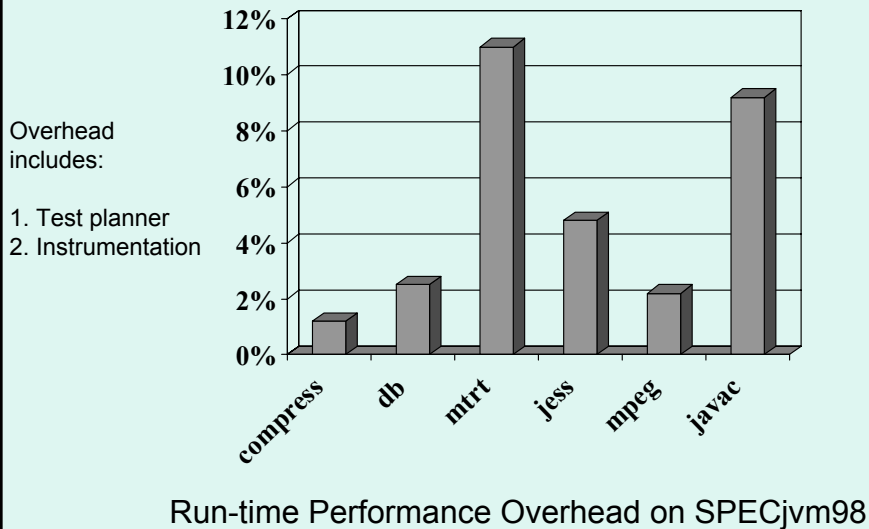
Mark edge hit

Insert at next point

Remove instrumentation



Preliminary Results: Branch Coverage Testing



Summary & Future Work

- Initial framework instance has low overhead (1-11%)
- On-going development of prototype
 - Def-use, statement, method, etc.
- Reduce number of instrumentation points
- Handle optimized target machine code
- <http://www.cs.pitt.edu/copa/eclipse.html>