Efficient and Precise Dynamic Impact Analysis Using Execute-After Sequences

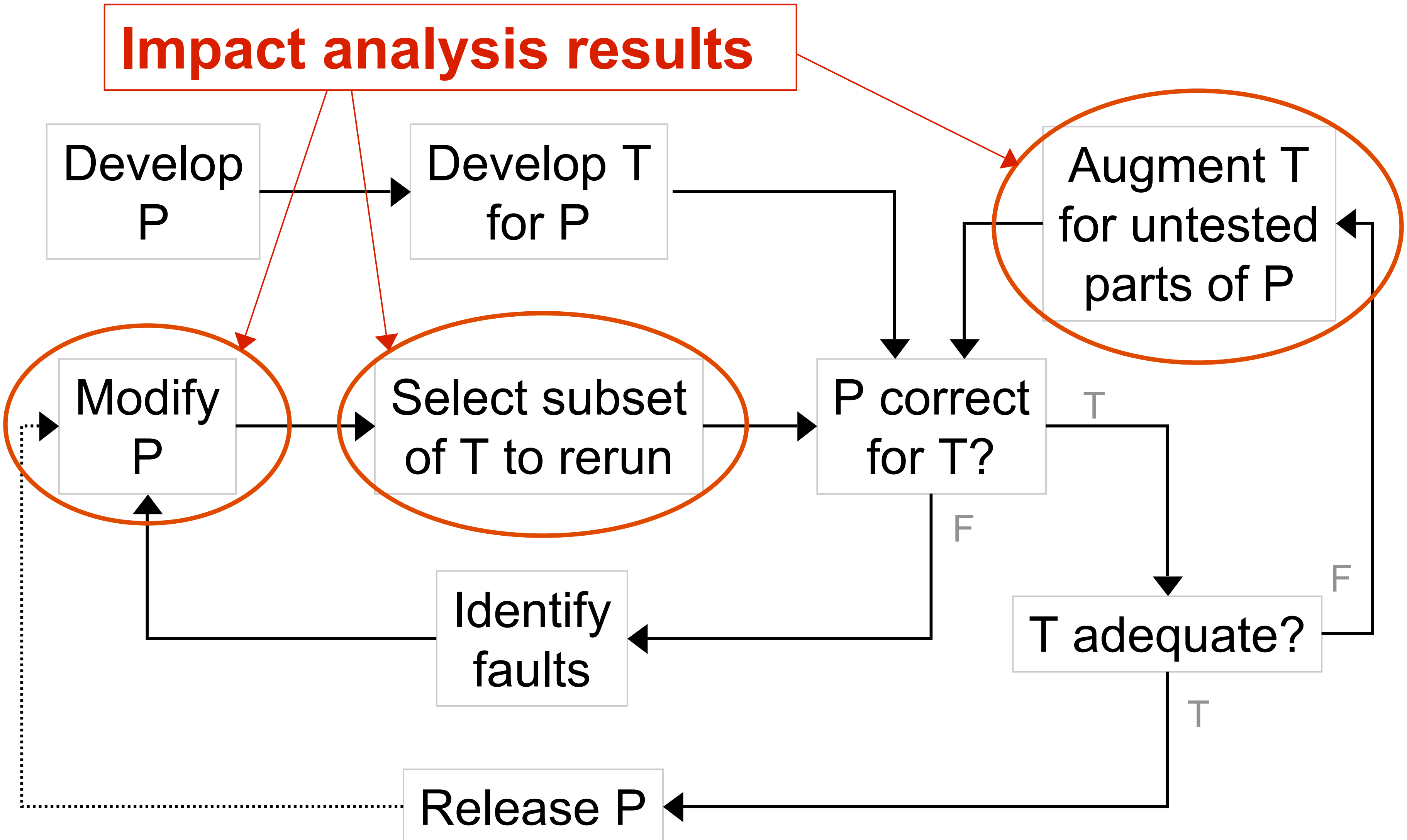
Taweesup (Term) Apiwattanapong

Alessandro Orso
Mary Jean Harrold

College of Computing
Georgia Institute of Technology

National Science Foundation awards CCR-0306372, CCR-0205422, CCR-0209322, SBE-0123532, and EIA-0196145 to Georgia Tech, Tata Consultancy Services

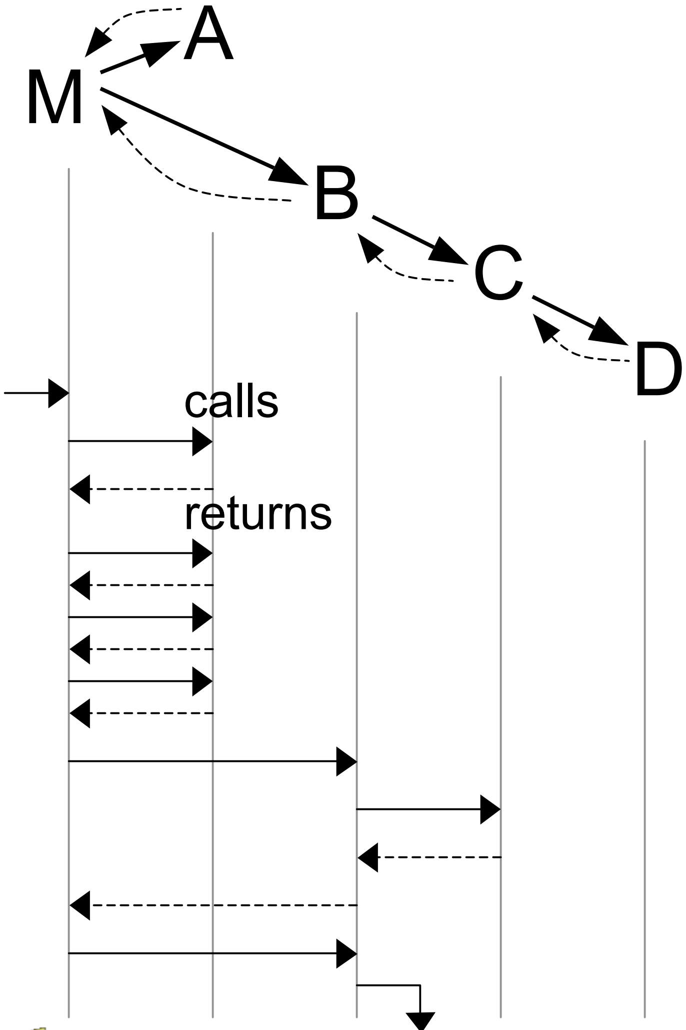
Development and Maintenance



Outline

- Dynamic Impact Analysis
- Existing Techniques
- Our Technique
- Empirical Studies
- Related Work
- Conclusion

Impact Analysis



Change = {C}

Static Impact Analysis

Impact Set = {M, A, B, C, D}

Dynamic Impact Analysis

Impact Set = {M, A, B, C}

Dynamic Impact Analysis

Impact-analysis techniques that

- Are based on dynamic information (e.g., test suites, field executions)
- Are conservative w.r.t. dynamic information

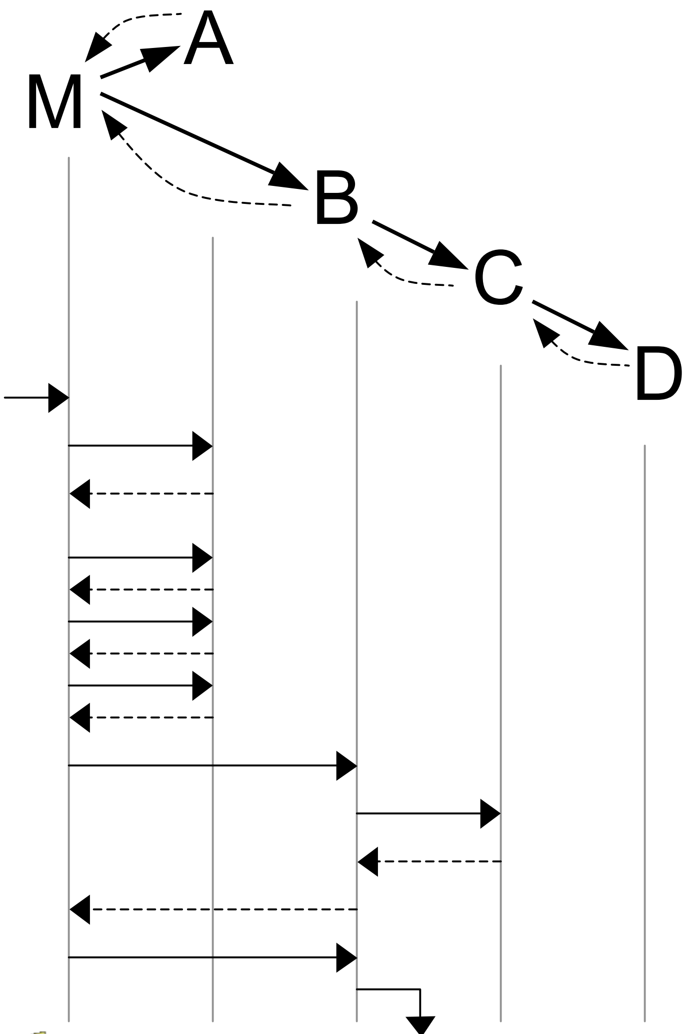
Quality of the dynamic information is key!

- representativeness of actual usage
  - → collect actual usage → efficiency is important

Two existing dynamic impact-analysis techniques

- PathImpact
- CoverageImpact

Existing Techniques



Change = {C}

PathImpact

M	A	r	…	A	r	B	C	r	r	B	x

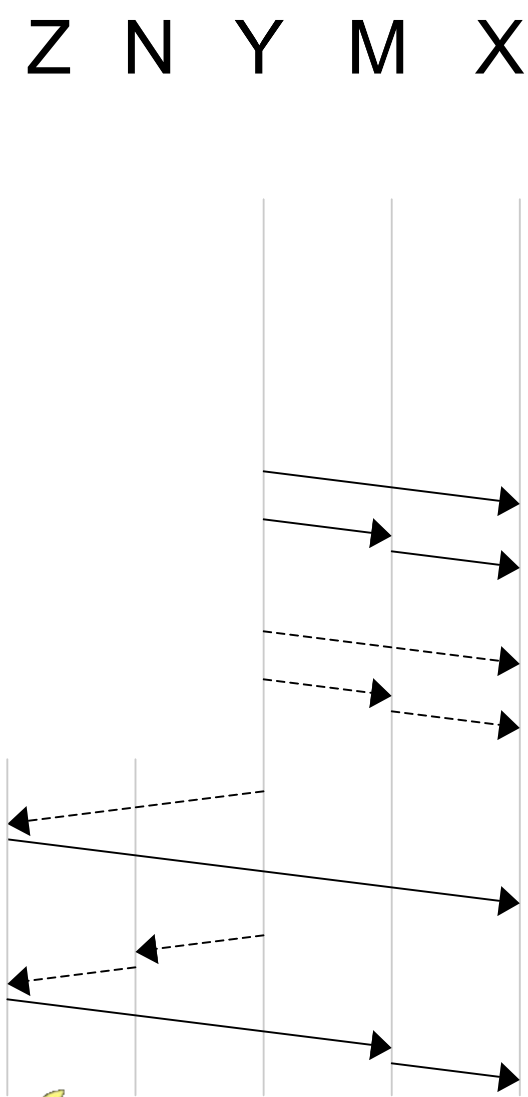
Impact Set = {M, B, C}

CoverageImpact

M	A	B	C	D
1	1	1	1	0

Impact Set = {M, A, B, C}

Execute-After (EA) Relation



Essential information is

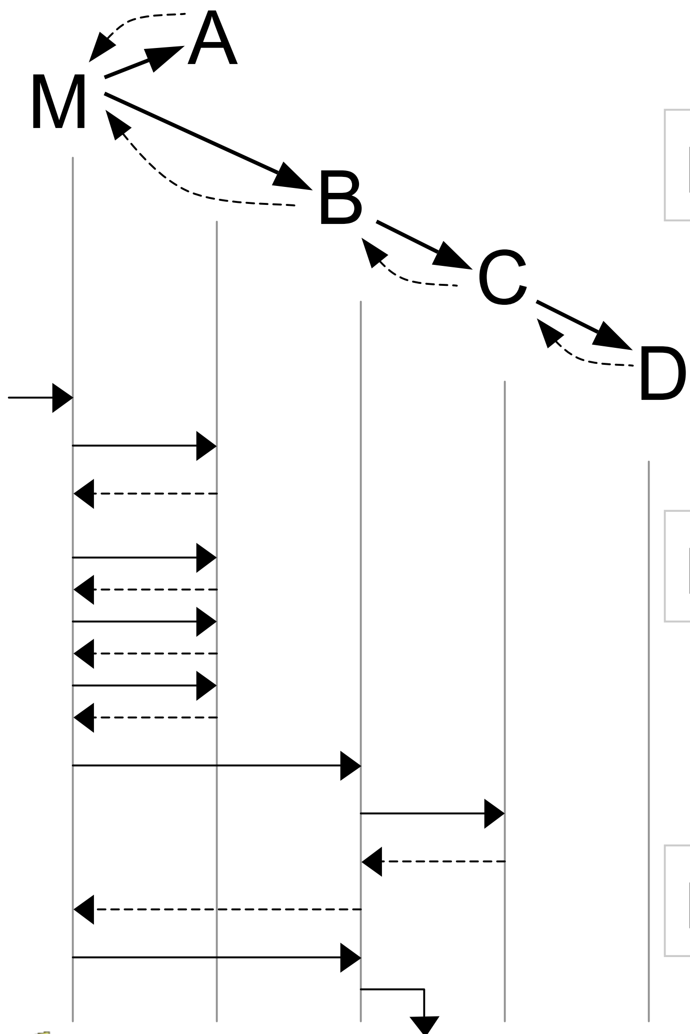
"Execute-After Relation"

Definition

Given a program *P*, a set of executions *E*, and two methods *X* and *Y* in *P*, $(X,Y) \in EA$ for *E* if and only if, in at least one execution in *E*,

1. *Y* calls *X* (directly or transitively (d/t)),
2. *Y* returns into *X* (d/t), or
3. *Y* returns into a method *Z* (d/t), and *Z* later calls *X* (d/t).

Computing EA Relation



Use method-entry and method-return-into events

$M_e \quad A_e \quad M_i \ \ldots \ A_e \quad M_i \quad B_e \quad C_e \quad B_i \quad M_i \quad B_e$

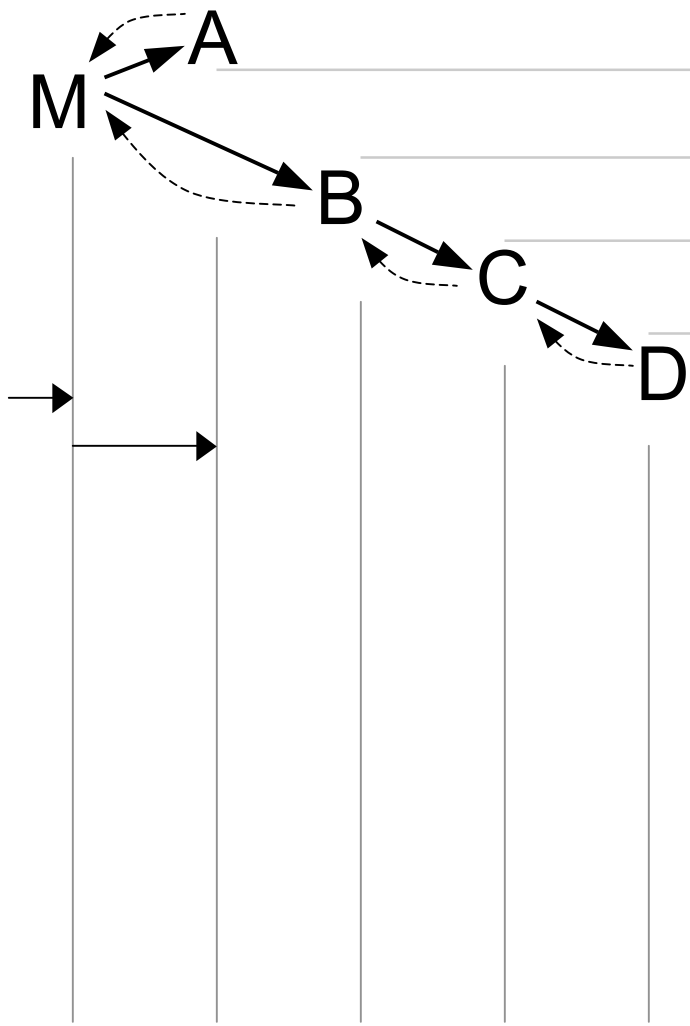
Use only the first and last events of each method

$M_e \quad A_e \quad A_e \quad B_e \quad C_e \quad M_i \quad B_e$

Execute-After (EA) Sequence

$M_f \quad A_f \quad A_l \quad B_f \quad C_f \quad C_l \quad M_l \quad B_l$

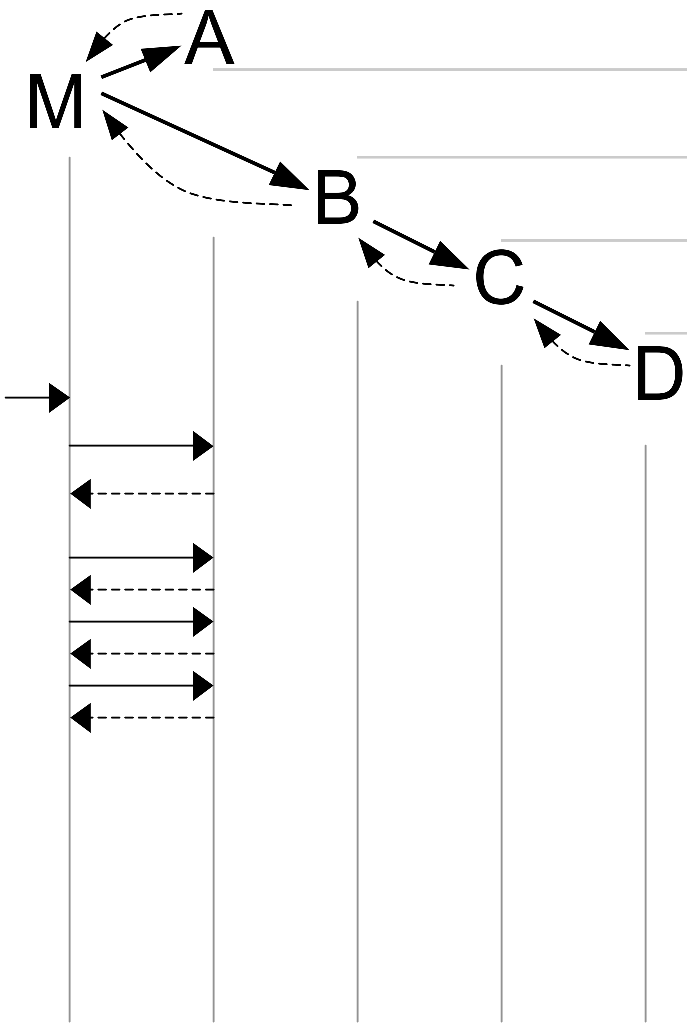
Our Technique



F	L
2	2
1	1

Current Timestamp: 3

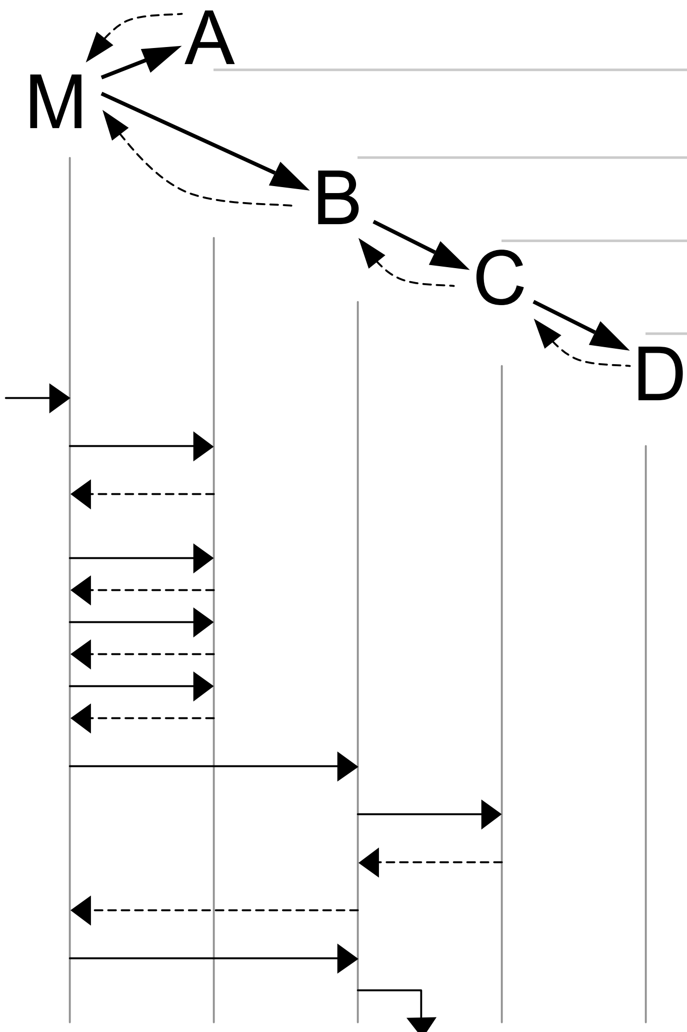
Our Technique



F	L
2	8
1	9

Current Timestamp: **10**

Our Technique



F	L
2	8
1	13
10	14
11	11

Current Timestamp: **15**

Sort the entries to obtain an Execute-After (EA) Sequence

M_f A_f A_l B_f C_f C_l M_l B_l

Analytical Complexity

Techniques	Space	Time (per method call)
PathImpact	$O(t)$	$O(t)$
CoverageImpact	m bits	$O(1)$
Execute-After	2m integers	$O(1)$

Program size: 30 KLOC

Trace size: 2 GB

t = the size of the trace
m = the number of methods

Outline

- Dynamic Impact Analysis
- Existing Techniques
- Our Technique
- Empirical Studies
- Related Work
- Conclusion

Empirical Studies

Studies

- Efficiency
- Precision

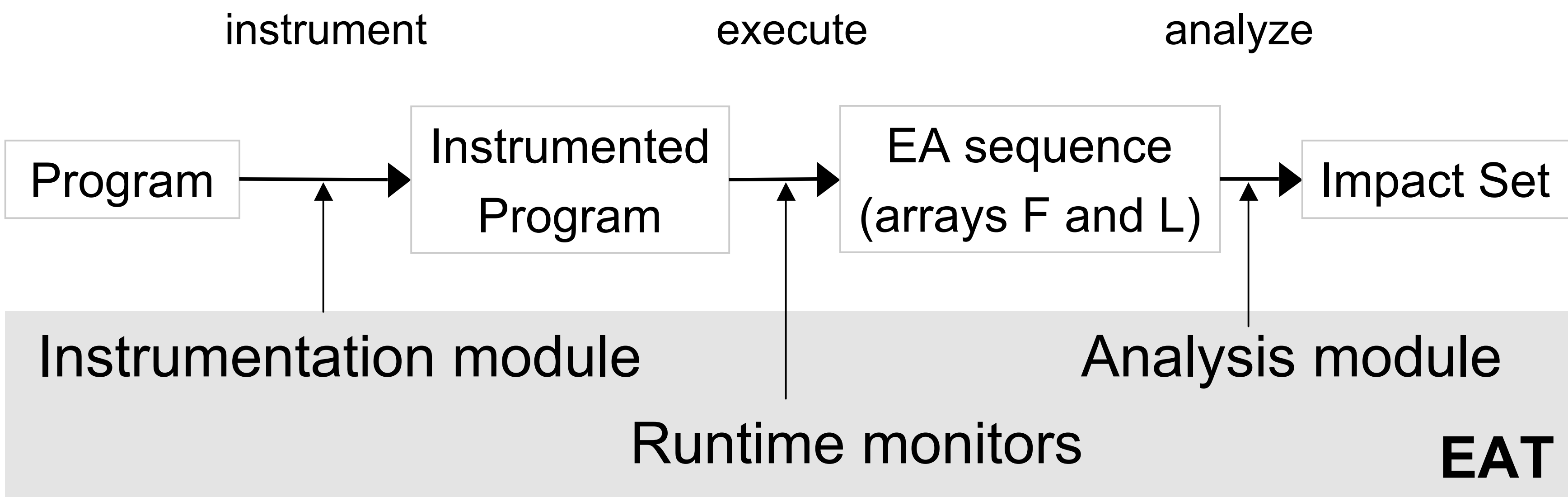
Experimental Setup

Tool: EAT (Execute-After Tool)

Subject Programs:

Program	Versions	Classes	Methods	LOC	Test Cases
Siena	8	24	219	3674	564
Jaba	11	355	2695	33183	125
Jaba-long	11	355	2695	33183	90

EAT (Execute-After Tool)



Collecting method-return-into events

- Normal return
- Exceptional return into a catch block
- Exceptional return into a finally block

Study 1: Efficiency

Goal: To evaluate relative execution costs for Execute-After (EA) wrt CoverageImpact (CI) and PathImpact (PI)

Method: Measure time to execute programs on test cases, gather dynamic data, and output information to disk.

	Running time (ms)				Overhead (%)		
Program	Uninst.	CI	EA	PI	CI	EA	PI
Siena	53	108	110	~263	104	108	~396
Jaba	432	463	486	~54,000	7.18	12.50	~12,400
Jaba-long	5257	5617	5861	-	6.85	11.49	-

Study 2: Precision

Goal: To compare the precision of the three techniques

Method: Measure the relative sizes of impact sets computed by the techniques.

Related Work

PathImpact (Law and Rothermel)

- based on lightweight dynamic forward slicing

CoverageImpact (Orso et al.)

- based on compressed program traces

Online impact analysis (Breech et al.)

- compute impact sets online
- space complexity: x^2
- time complexity (per method call): $O(x)$
- (x is the number of methods executed)

Conclusion

Summary

- identify essential information for dynamic impact analysis
- present a new, efficient, and precise technique to collect and analyze that information
- present a set of empirical studies which show the efficiency and effectiveness of our technique

Future directions

- perform studies using the technique in the field
- perform client-analysis
- generalization of the technique
  - levels of granularity
  - programming languages
- apply the technique to other dynamic analyses
  - reverse-engineering
  - recovery of feature interaction

Thank you.

Questions?