



Using Testing Trace for Automatic User Categorization

J. Jenny Li and David M. Weiss



Outline

- Motivations
- Trace-based user classification method
- Implementation
- Study results
- Conclusions

Large Amount of Testing Data

- Testing accounts for 50% of development
- Many kinds of testing:
 - Unit testing
 - Integration testing
 - System testing
 - Performance testing
- Large amount of test data can be used for?
 - Find bugs
 - Prediction
 - Others?

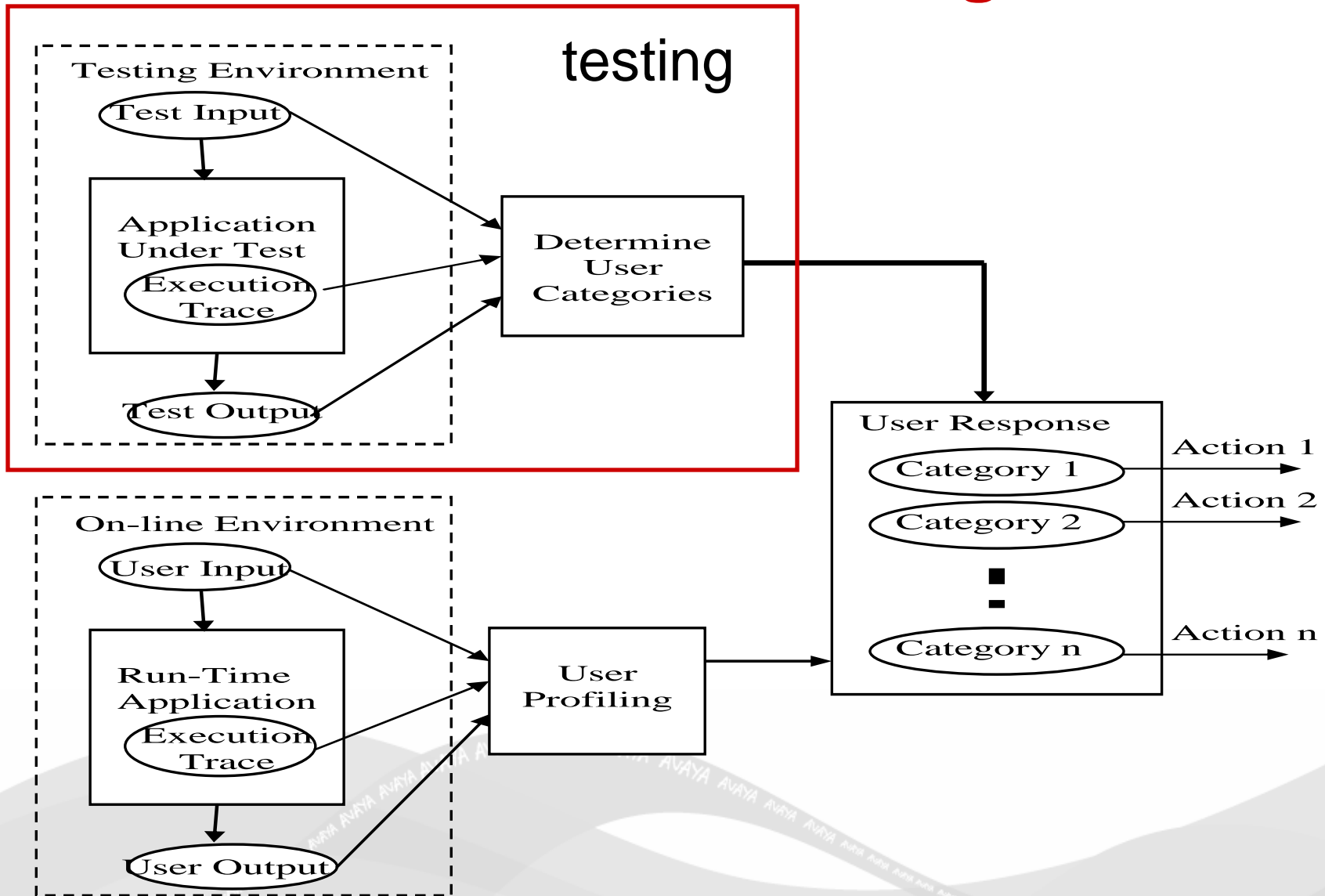
Additional Usage of Testing Data

- Artifacts:
 - Program under testing
 - Change frequency of the program
 - Program metrics such as complexity and size
- Traces:
 - Which feature is the most frequently used
 - Sequence of feature usage
 - Usage similarity
- Can predict field usage
 - Otherwise bad testing

Summarize Test Data for Field Usage

- Anticipated Usage
 - Scenarios never been tested before: savvy or random users
 - Some features are used more often than others
 - Prioritizing regression test sets gives a sense of confidence
- How to use it in service to user
 - Use test data as training set
 - User profile → user categories
 - Services tailored to user type
- Assumption:
 - All test artifacts and traces are available
 - No 100% data coverage
 - Each test is independent

Test-Trace-Based User Categorization



Step 2: Run-time Profiling

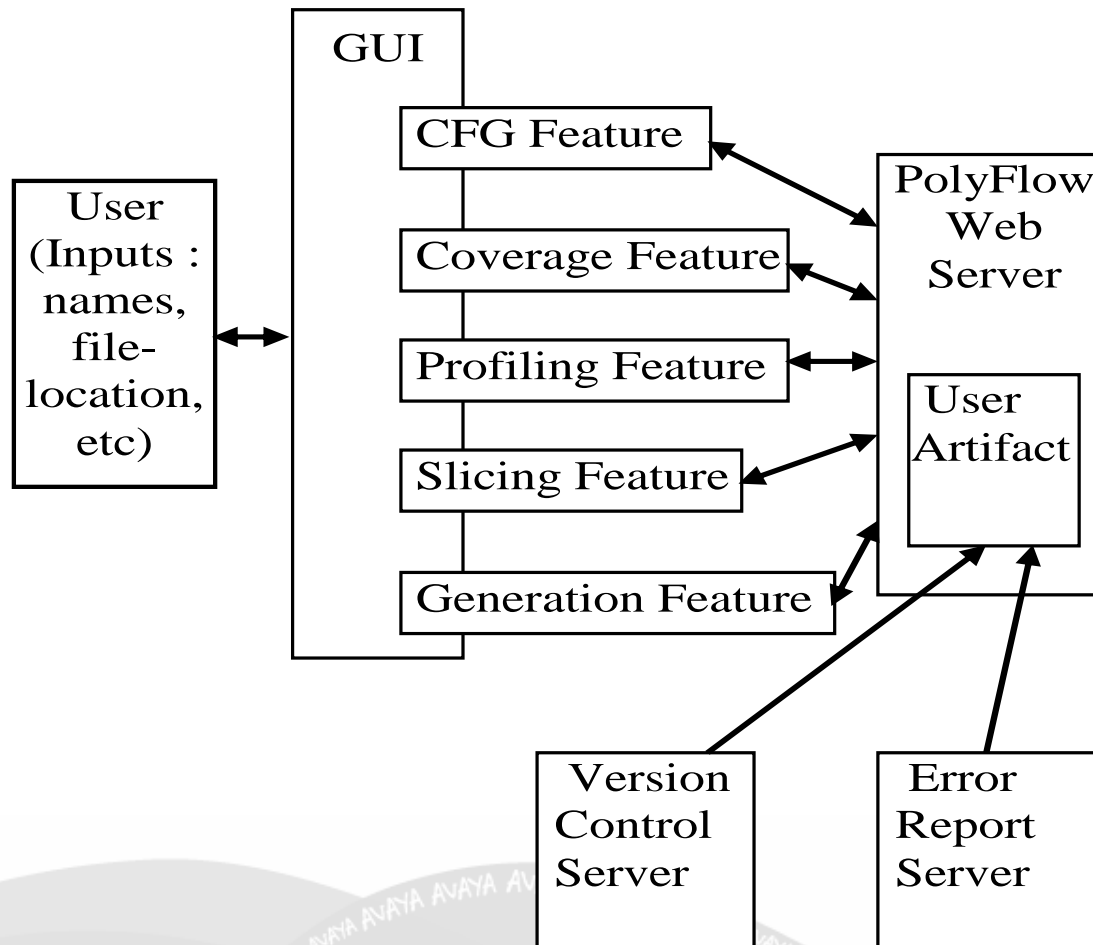
- Insert probes to monitor execution
 - Feature trace
 - Metrics of artifacts
 - User behavior

- **No need to store profiles**

Step 3: Tailored Responses

- Notify users if they need help
- Put the same type of users into a group chat
- Connect to high-level users for help
- Other tailored services
 - Voice/chat live help
 - Appropriate documents
 - Right-level samples and tutorials
- **Most innovative and important step**

Case Study: Software Testing Web Service

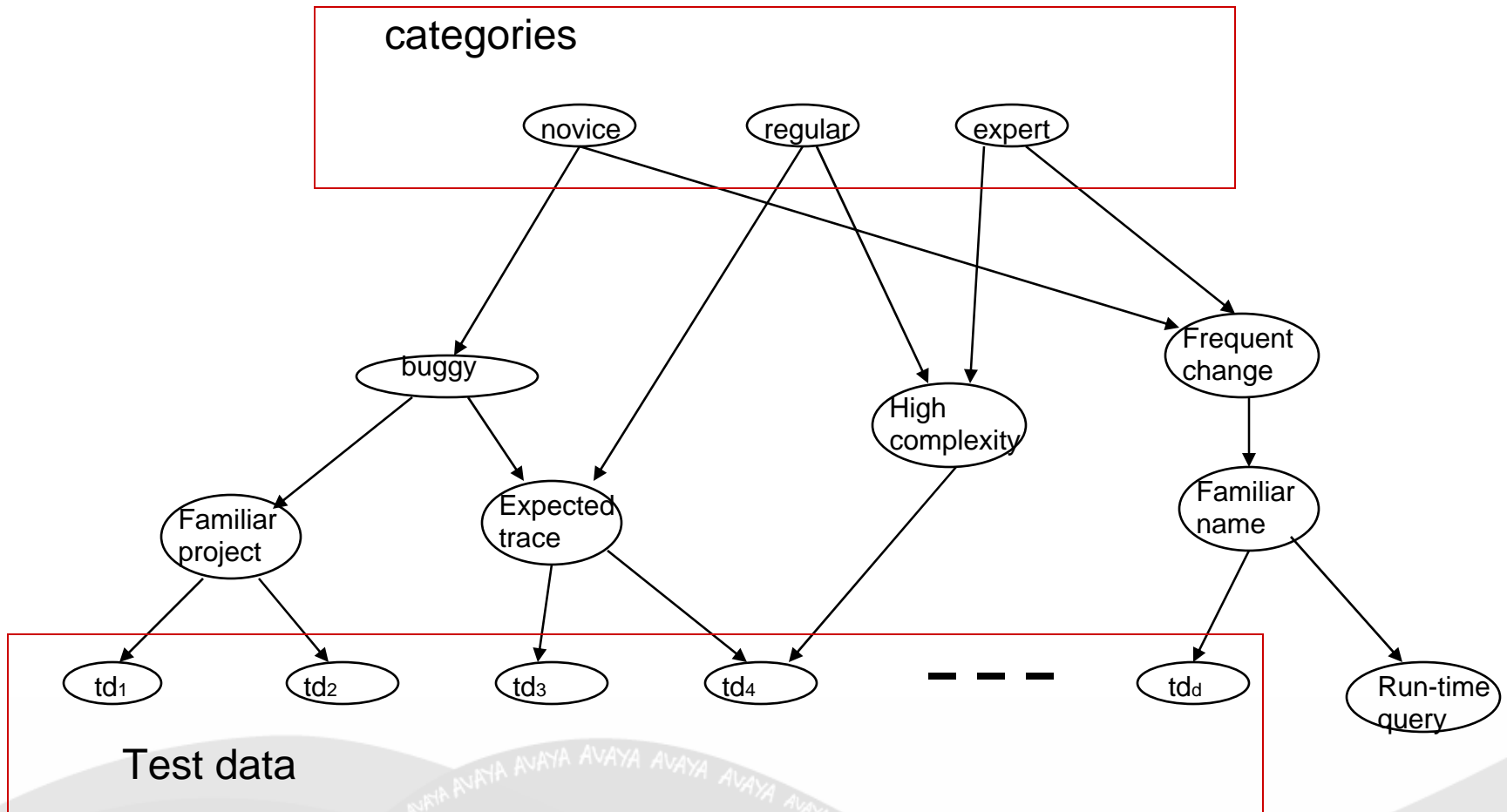


Each Test Data

User Name	Header
Project Name	
Method-begin	Execution trace
..... Node-in-method	
..... Method-begin	
..... Last-method-end	
Code complexity	Result
CFG complexity	
.....	

- $d ==$ number of test cases

User Category Classifier



<http://ti.arc.nasa.gov/project/autoclass/autoclass-c/>

Tailored User Assistance

- Novice user:
 - Red-flag
 - Connect to network for help
- Regular user:
 - Yellow-flag
 - Connect to additional training
- Expert user:
 - Green-flag
 - If can offer help to others
 - Feedback/suggestion on additional new features
- **Most innovative and important step**

Summary

- Take advantage of the large amount of testing data
- Create a user category classifier based on test data
- Run-time classification and tailored service
 - No need to store user profiles
- Future work:
 - Other usage of testing data
 - Security detection: outliner?

AVAYA *labs*

Questions and answers



PolyFlow Product Line

- A product line of testing services
 - Recover program structure
 - Static analysis
 - Run test
 - Coverage
 - testGen
 - Fault localization
- Performance
 - Tradeoff: analysis vs execution-time
 - Todo: find a balance