Building Practical Automation for Software Engineers

Kevin Moran, Ph.D.
Assistant Professor
Director of the SAGE Research Lab
University of Central Florida

CECS Virtual Seminar Series

Friday, September 29th, 2023

My Academic Background

My Academic Background

Completed
Undergrad
(Physics Degree)

2013

My Academic Background

Completed
Undergrad
(Physics Degree)

2013

2015

WILLIAM & MARY
CHARTERED 1693

Completed
M.S. Degree in CS

UCF

My Academic Background

Completed Undergrad (Physics Degree)

2013

2015

WILLIAM & MARY
CHARTERED 1693

Completed M.S. Degree in CS

Completed Ph.D. in CS

WILLIAM & MARY
CHARTERED 1693

2018

UCF

My Academic Background

Completed Undergrad (Physics Degree)

2013

2015

WILLIAM & MARY
CHARTERED 1693

Completed M.S. Degree in CS

Completed Ph.D. in CS

WILLIAM & MARY
CHARTERED 1693

2018

2020

WILLIAM & MARY
CHARTERED 1693

Completed Post-Doc

UCF

My Academic Background

Completed Undergrad (Physics Degree)

2013

2015

Completed M.S. Degree in CS

Completed Ph.D. in CS

2018

2020

Completed Post-Doc

Started as Assistant Prof

2020

My Academic Background

Completed Undergrad (Physics Degree)

2013

2015

Completed M.S. Degree in CS

2018

Completed Ph.D. in CS

2020

Completed Post-Doc

2020

Started as Assistant Prof

2023

Started as Assistant Prof

About Myself & the SAGE Lab

About Myself & the SAGE Lab

- Kevin Moran, Ph.D.
- Assistant Professor
- Department of Computer Science
- Member of the UCF *Cyber Security and Privacy Cluster* (CyberSP)
- kpmoran@ucf.edu
- https://www.kpmoran.com/

About Myself & the SAGE Lab

- Kevin Moran, Ph.D.
- Assistant Professor
- Department of Computer Science
- Member of the UCF *Cyber Security and Privacy Cluster* (CyberSP)
- kpmoran@ucf.edu
- https://www.kpmoran.com/

- ***Research Interests:***
  - Software Engineering, Maintenance, & Evolution
  - Machine Learning
  - Software Security
- ***Teaching Interests:***
  - Programming (currently teaching COP-3223)
  - Software Design & Testing
- ***Key Achievements/Awards:***
  - ACM SigSoft Distinguished Paper Award at *ESEC/FSE'19*
  - ACM SigSoft Distinguished Paper Award at *ICSE'20*
  - Best Paper Award at *CODASPY'19*
  - Rising Star Award at MobileSoft'23
  - 5 Distinguished Reviewer Awards

About Myself & the SAGE Lab

- Kevin Moran, Ph.D.
- Assistant Professor
- Department of Computer Science
- Member of the UCF *Cyber Security and Privacy Cluster* (CyberSP)
- kpmoran@ucf.edu
- https://www.kpmoran.com/

- ***Research Interests:***
  - Software Engineering, Maintenance, & Evolution
  - Machine Learning
  - Software Security
- ***Teaching Interests:***
  - Programming (currently teaching COP-3223)
  - Software Design & Testing
- ***Key Achievements/Awards:***
  - ACM SigSoft Distinguished Paper Award at *ESEC/FSE'19*
  - ACM SigSoft Distinguished Paper Award at *ICSE'20*
  - Best Paper Award at *CODASPY'19*
  - Rising Star Award at MobileSoft'23
  - 5 Distinguished Reviewer Awards

About Myself & the SAGE Lab

- Kevin Moran, Ph.D.
- Assistant Professor
- Department of Computer Science
- Member of the UCF *Cyber Security and Privacy Cluster* (CyberSP)
- kpmoran@ucf.edu
- https://www.kpmoran.com/

- ***Research Interests:***
  - Software Engineering, Maintenance, & Evolution
  - Machine Learning
  - Software Security
- ***Teaching Interests:***
  - Programming (currently teaching COP-3223)
  - Software Design & Testing
- ***Key Achievements/Awards:***
  - ACM SigSoft Distinguished Paper Award at *ESEC/FSE'19*
  - ACM SigSoft Distinguished Paper Award at *ICSE'20*
  - Best Paper Award at *CODASPY'19*
  - Rising Star Award at MobileSoft'23
  - 5 Distinguished Reviewer Awards

SAGE Lab (https://sagelab.io/)

- The **S**oftware **A**utomation, **G**eneration, and **E**ngineering Research Lab
- 6 PhD Students Across UCF and GMU

 Junayed Mahmud

 Arun Krishnavajjala

 Atish Dipongkor

 Sabiha Salma

 SM Hasan Mansur

 Safwat Ali Khan

- ***Research Funding & Collaborations***
  - The Lab has been awarded over ***$4.2 million*** in research funding during it's first three years from government and industry sources
  - We have strong ties to industry, ***to prove out our research in practice***

Software Engineering

Software Engineering

The methods and techniques by which developers design, create, test, and manage software

Software Engineering

The methods and techniques by which developers design, create, test, and manage software

My research goal: Design tailored *automated approaches* to help facilitate *developer needs* throughout the software development and maintenance lifecycle.

How Can We Design Practical Automation?

Mining Software Repositories

Mining Software Repositories

Mining Software Repositories

Textual Software Artifacts

Source Code Files

Software Documentation

Bug Reports

Mining Software Repositories

Textual Software Artifacts

Source Code Files

Software Documentation

Bug Reports

Graphical Software Artifacts

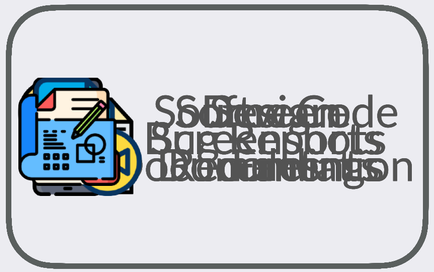
Design Documents

Screenshots

Screen Recordings

UCF

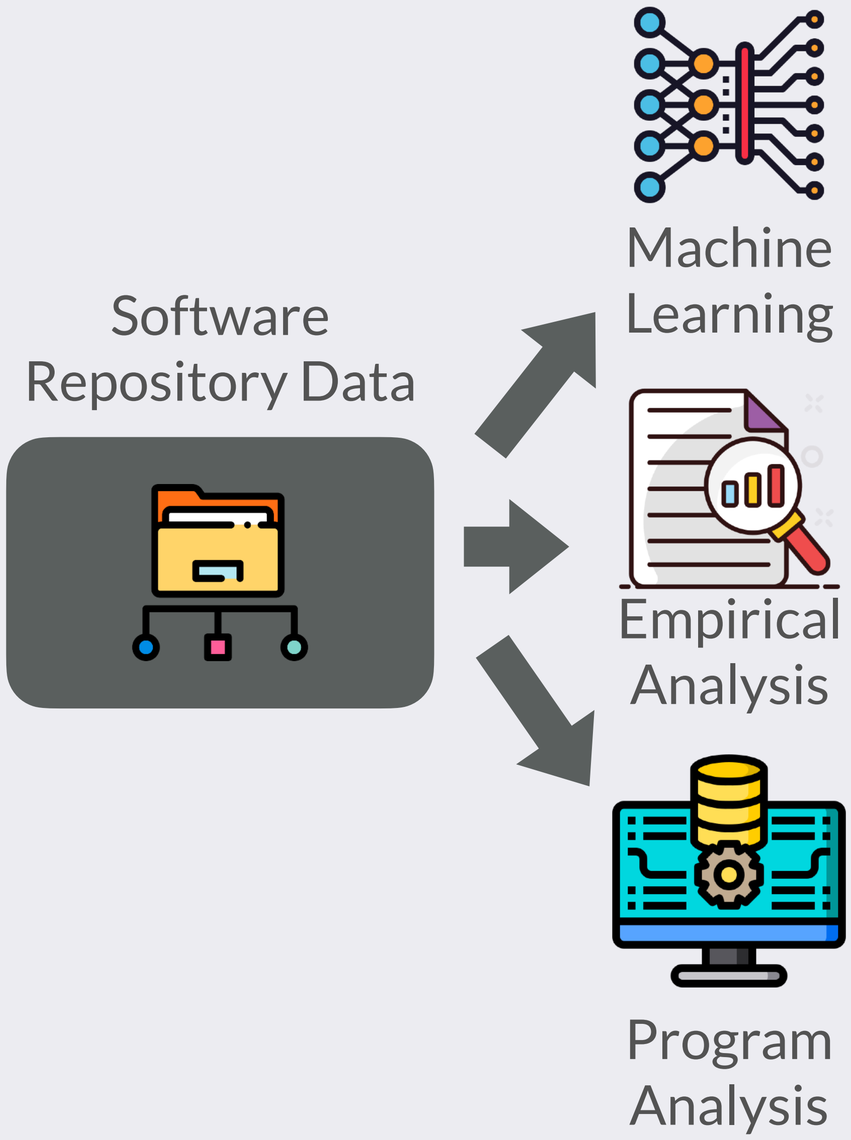
Learning Patterns from Software Data



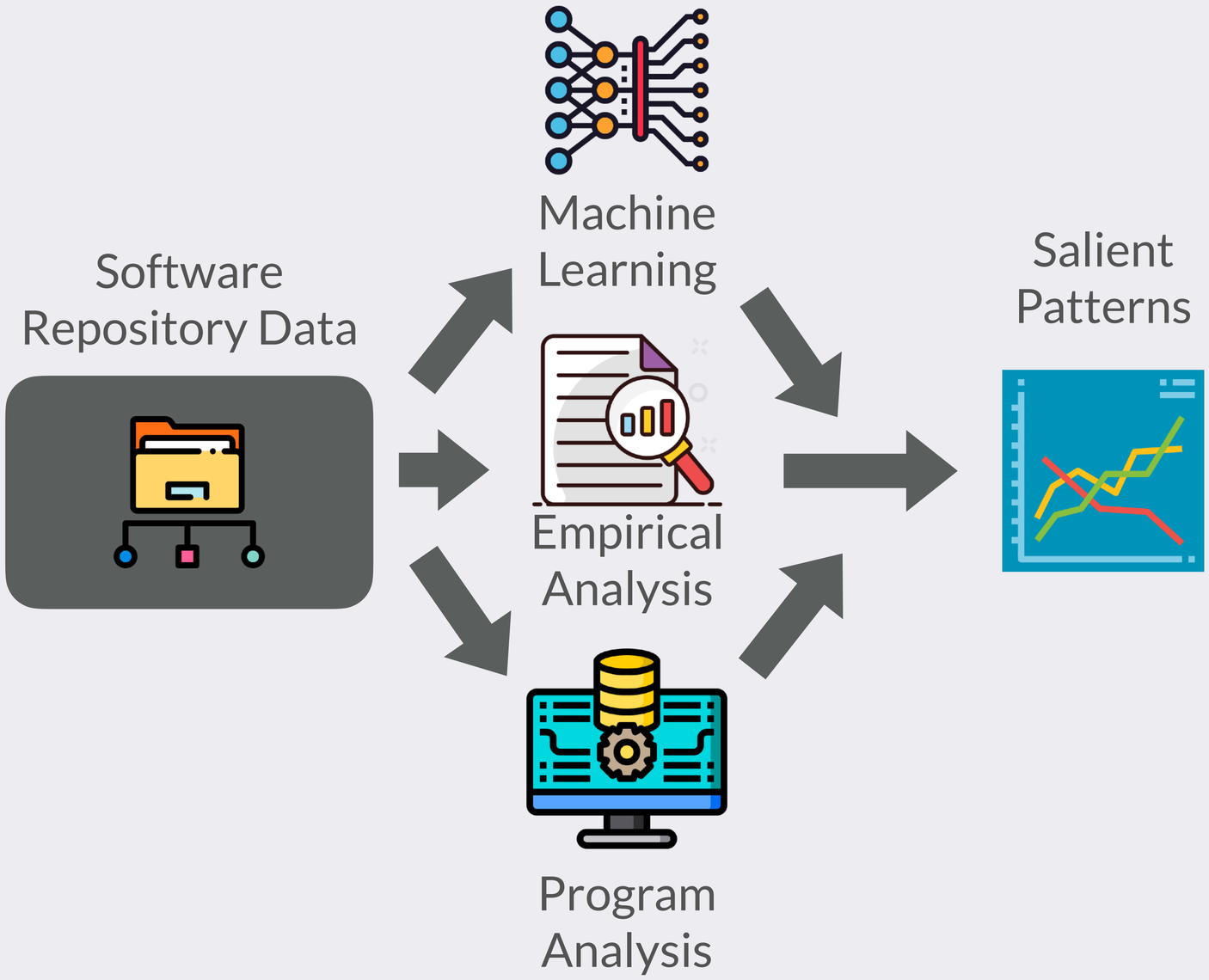
Learning Patterns from Software Data

Software
Repository Data

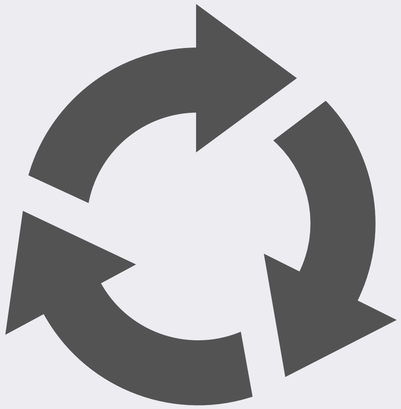
Learning Patterns from Software Data

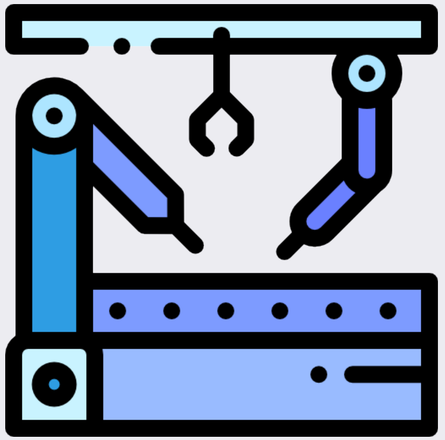


Learning Patterns from Software Data



Developer-Centric Automation





Practical Significance

Blend *scientific discovery* with *practical significance*

Practical Significance

Blend *scientific discovery* with *practical significance*

Overview of Research Areas

Research Title:

Automated UI Analysis

- ***Motivation***
  - *User Interfaces (UIs)* contain a wealth of information about software systems
  - This information is typically *orthogonal* to the information contained in code and other software artifacts
  - We can *learn patterns* from this information to automate Software Development Tasks

Objectives/Approach

Impact, Outcomes, and Achievements

Overview of Research Areas

Research Title:

Automated UI Analysis

- ***Motivation***
  - *User Interfaces (UIs)* contain a wealth of information about software systems
  - This information is typically *orthogonal* to the information contained in code and other software artifacts
  - We can *learn patterns* from this information to automate Software Development Tasks

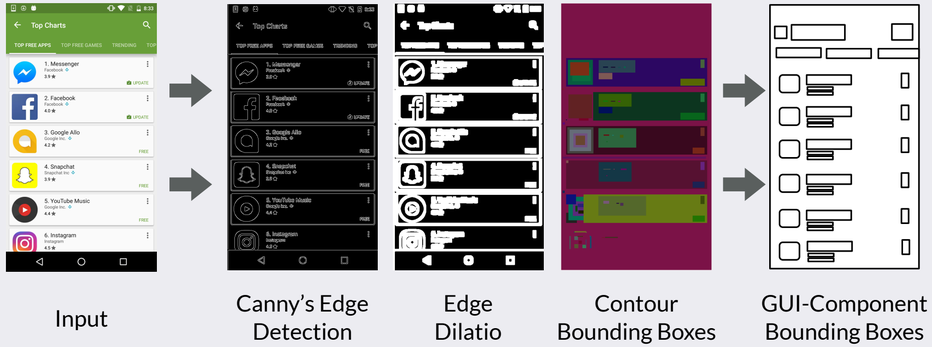


Fig. 1: Overview of the ReDraw Technique for Automated Prototyping of UIs

Objectives/Approach

Impact, Outcomes, and Achievements

Overview of Research Areas

Research Title:

Automated UI Analysis

- ***Motivation***
  - *User Interfaces (UIs)* contain a wealth of information about software systems
  - This information is typically *orthogonal* to the information contained in code and other software artifacts
  - We can *learn patterns* from this information to automate Software Development Tasks

Objectives/Approach

Impact, Outcomes, and Achievements

Overview of Research Areas

Research Title:

Automated UI Analysis

- ***Motivation***
  - *User Interfaces (UIs)* contain a wealth of information about software systems
  - This information is typically *orthogonal* to the information contained in code and other software artifacts
  - We can *learn patterns* from this information to automate Software Development Tasks

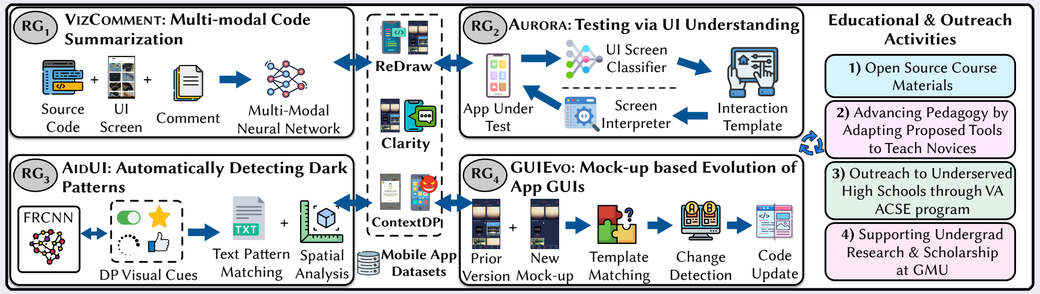


Fig. 2: Overview of theUI Analysis Research Agenda

Objectives/Approach

Impact, Outcomes, and Achievements

Overview of Research Areas

Research Title:

Automated UI Analysis

- ***Motivation***
  - *User Interfaces (UIs)* contain a wealth of information about software systems
  - This information is typically *orthogonal* to the information contained in code and other software artifacts
  - We can *learn patterns* from this information to automate Software Development Tasks

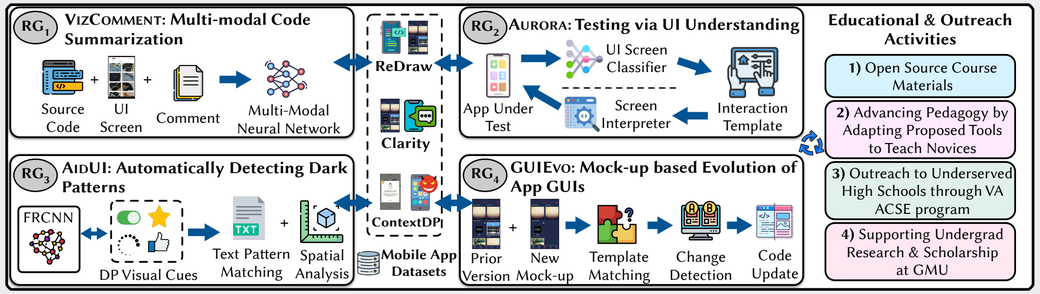


Fig. 2: Overview of theUI Analysis Research Agenda

Objectives/Approach

- ***Objectives:***
  - Determine what information can be *reliably learned from UIs*, and design tailored *deep learning techniques*
  - Automate the following *tasks*: UI design instantiation, UI Testing, Software Documentation, Bug Reporting
- ***Approach:***
  - We have developed several *bespoke deep learning techniques* to detect and learn patterns from UIs
  - *Multimodal transformer*-based models

Impact, Outcomes, and Achievements

UCF

Overview of Research Areas

Research Title:

Automated UI Analysis

- ***Motivation***
  - *User Interfaces (UIs)* contain a wealth of information about software systems
  - This information is typically *orthogonal* to the information contained in code and other software artifacts
  - We can *learn patterns* from this information to automate Software Development Tasks

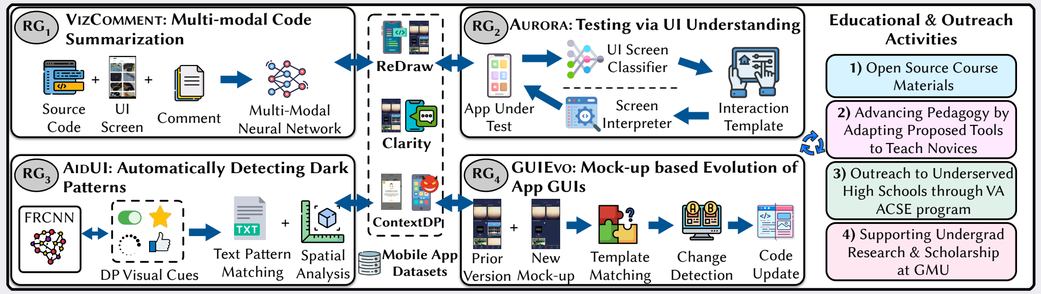


Fig. 2: Overview of theUI Analysis Research Agenda

Objectives/Approach

- ***Objectives:***
  - Determine what information can be *reliably learned from UIs*, and design tailored *deep learning techniques*
  - Automate the following *tasks*: UI design instantiation, UI Testing, Software Documentation, Bug Reporting
- ***Approach:***
  - We have developed several *bespoke deep learning techniques* to detect and learn patterns from UIs
  - *Multimodal transformer*-based models

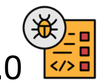
Impact, Outcomes, and Achievements

- ***Academic Achievements:***
  - 17 publications, 1 distinguished paper award
  - 11 Prototype research tools and 3 Datasets developed (open source)
- ***Practical Impact:***
  - The GVT tool for detecting UI display issues was deployed at a large multinational telecommunications company, and *is used by hundreds of developers/designers*
  - The V2S tool for record/replay testing has been deployed at a software testing company

UCF

Overview of Research Areas

Automated UI Analysis 

Research Title: Bug Reporting 2.0 

- ***Motivation***
  - Increasing Complexity of Software
  - Issue Trackers *have not evolved* in two decades
  - Empirical Studies illustrate bug reports are often *low-quality*
  - Tasks related to *bug report management* dominate Software Maintenance

Objectives/Approach

Impact, Outcomes, and Achievements

Overview of Research Areas

Automated UI Analysis

Research Title: Bug Reporting 2.0

- ***Motivation***
  - Increasing Complexity of Software
  - Issue Trackers *have not evolved* in two decades
  - Empirical Studies illustrate bug reports are often *low-quality*
  - Tasks related to *bug report management* dominate Software Maintenance

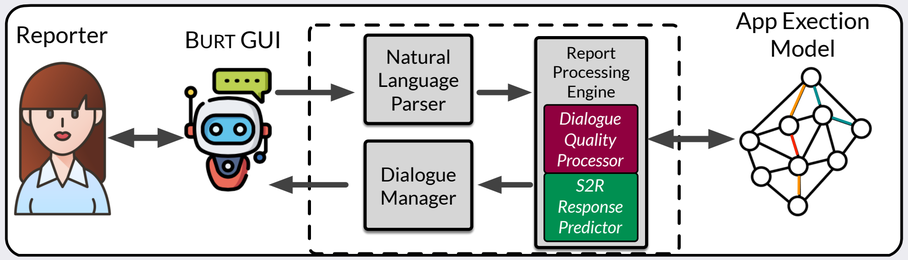


Fig. 1: Overview of the BURT Bug Reporting System

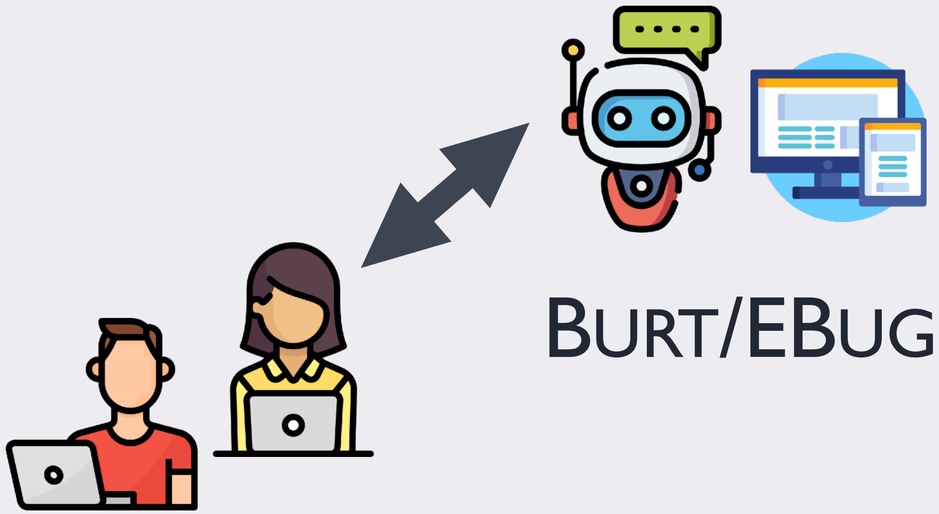
Objectives/Approach

Impact, Outcomes, and Achievements

BURT and EBUG: Enabling Interactive Bug Reporting

Reporters

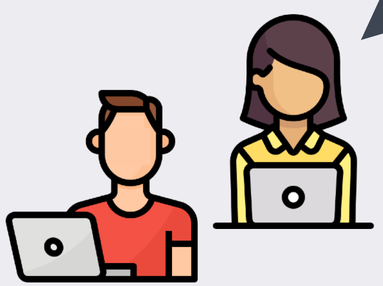
BURT and EBUG: Enabling Interactive Bug Reporting



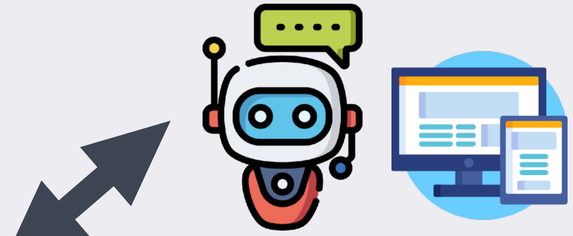
BURT/EBUG

Reporters

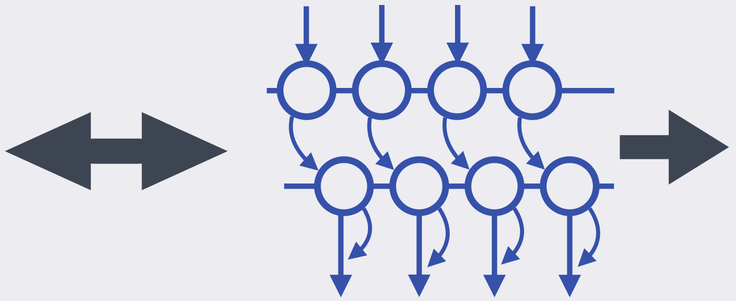
BURT and EBUG: Enabling Interactive Bug Reporting



Reporters



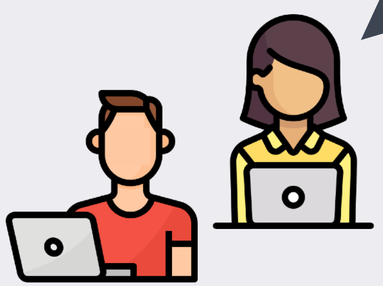
BURT/EBUG



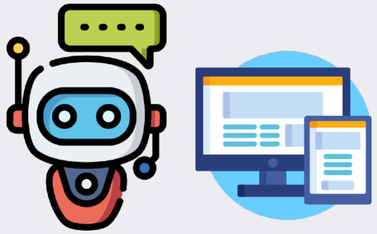
Natural Language Parser

S_1: [press] [‘Go to…’]
S_2: [type] [number]
S_3: [press] [‘Create’]
…

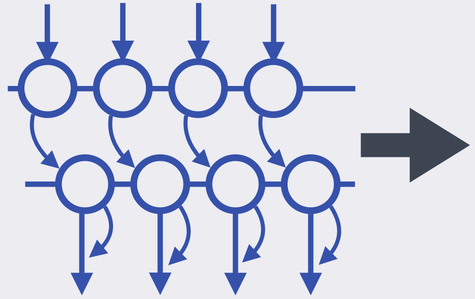
BURT and EBUG: Enabling Interactive Bug Reporting



Reporters

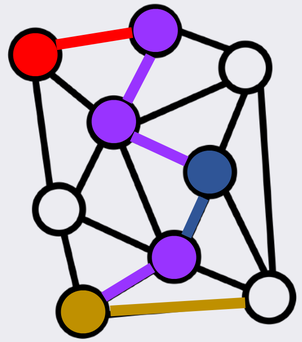


BURT/EBUG



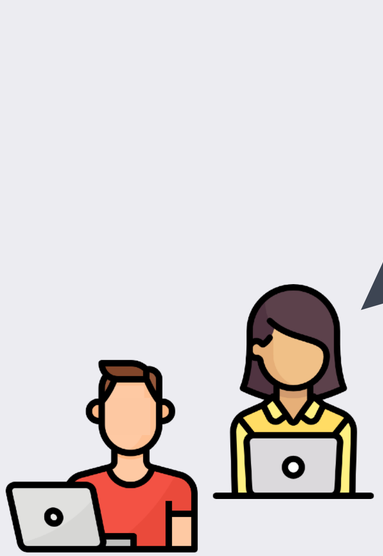
Natural Language Parser

S_1: [press] [‘Go to…’]
S_2: [type] [number]
S_3: [press] [‘Create’]
…

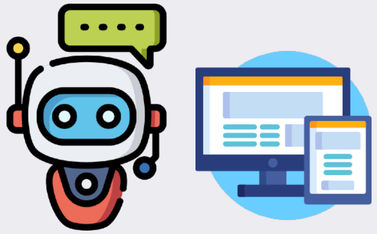


Application State Mapping

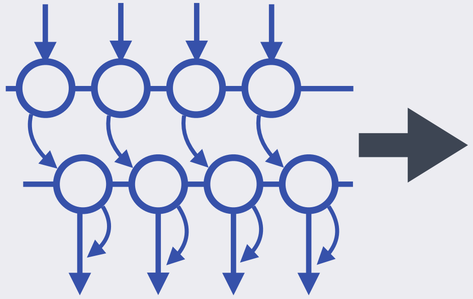
BURT and EBUG: Enabling Interactive Bug Reporting



Reporters

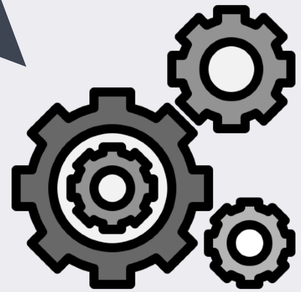


BURT/EBUG

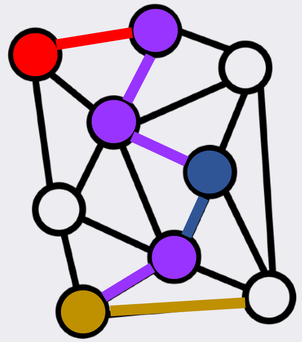


Natural Language Parser

S_1: [press] [‘Go to…’]
S_2: [type] [number]
S_3: [press] [‘Create’]
…



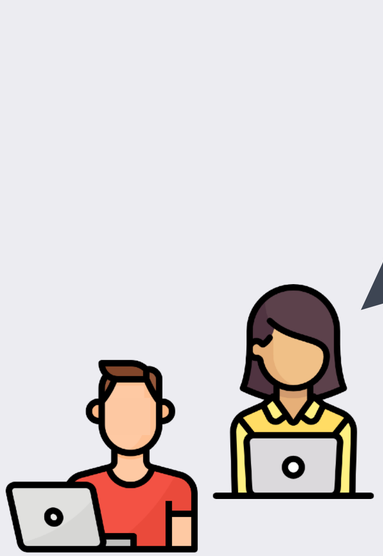
Prediction Engine



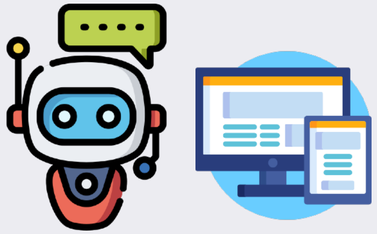
Application State Mapping

UCF

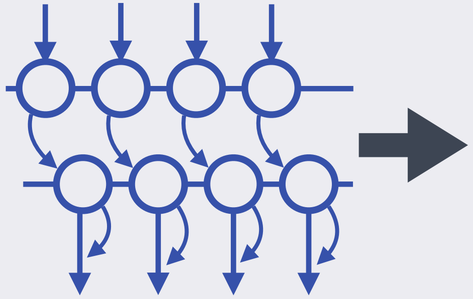
BURT and EBUG: Enabling Interactive Bug Reporting



Reporters

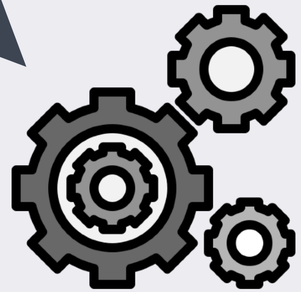


BURT/EBUG

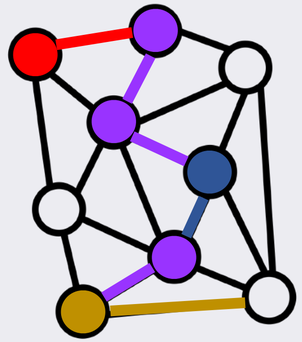


Natural Language Parser

S_1: [press] [‘Go to…’]
S_2: [type] [number]
S_3: [press] [‘Create’]
…



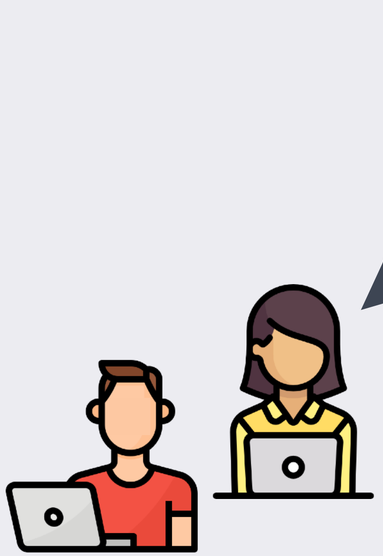
Prediction Engine



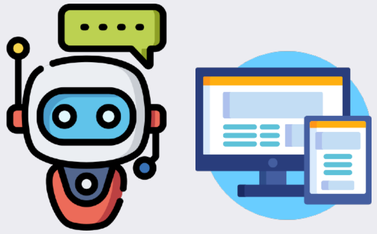
Application State Mapping

UCF

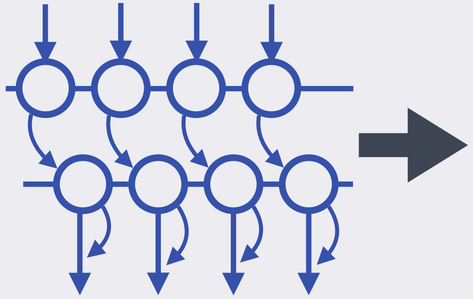
BURT and EBUG: Enabling Interactive Bug Reporting



Reporters

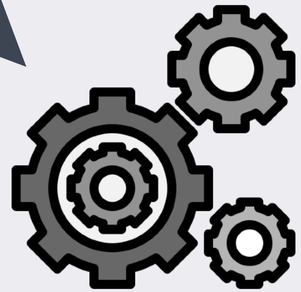


BURT/EBUG

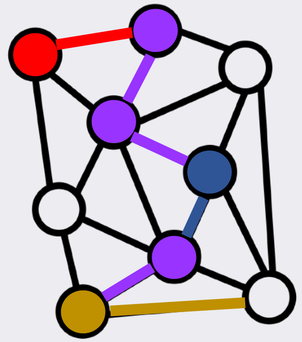


Natural Language Parser

S_1: [press] [‘Go to…’]
S_2: [type] [number]
S_3: [press] [‘Create’]
…



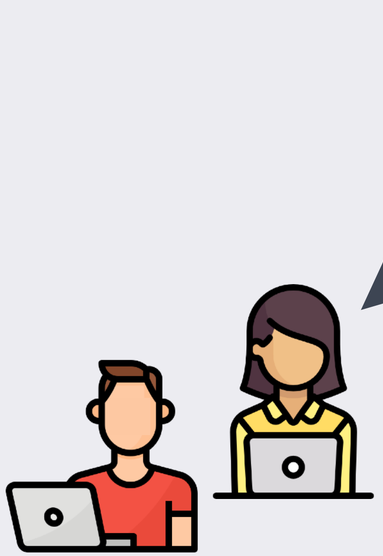
Prediction Engine



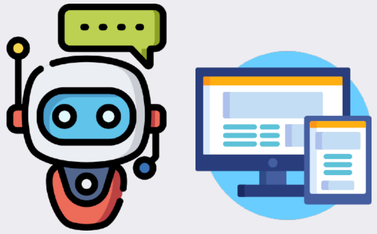
Application State Mapping

UCF

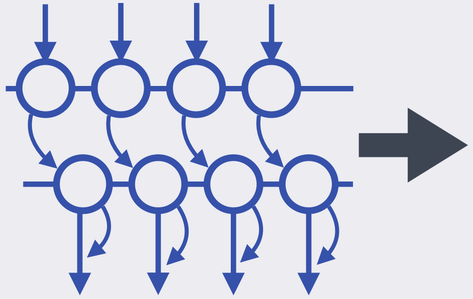
BURT and EBUG: Enabling Interactive Bug Reporting



Reporters

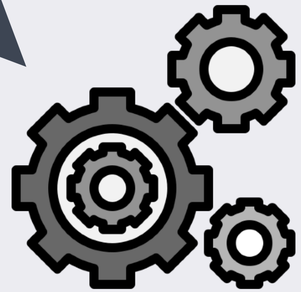


BURT/EBUG

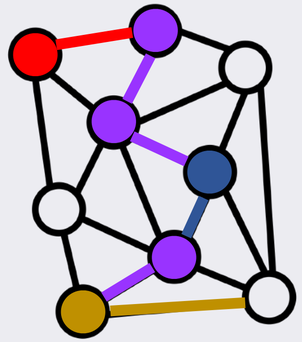


Natural Language Parser

S_1: [press] [‘Go to…’]
S_2: [type] [number]
S_3: [press] [‘Create’]
…



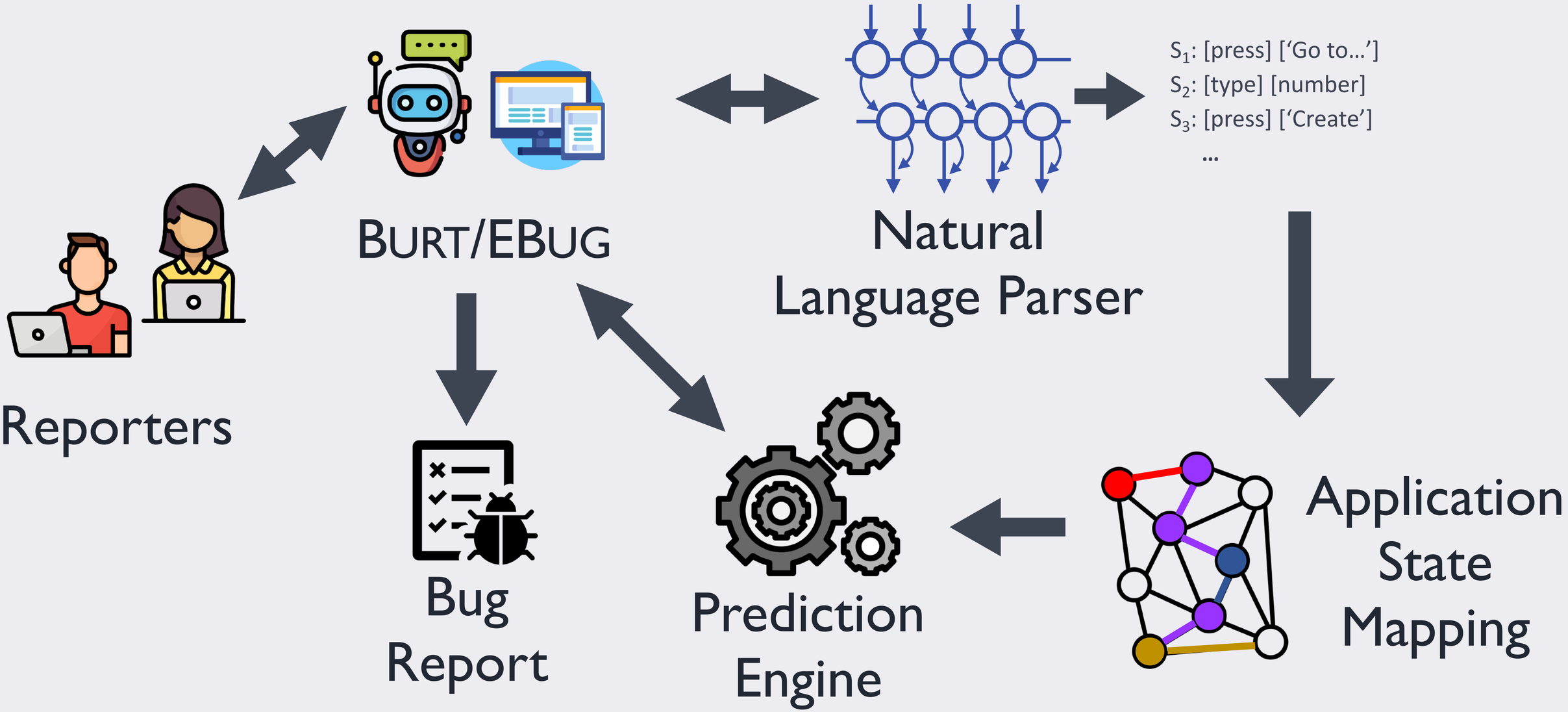
Prediction Engine



Application State Mapping

UCF

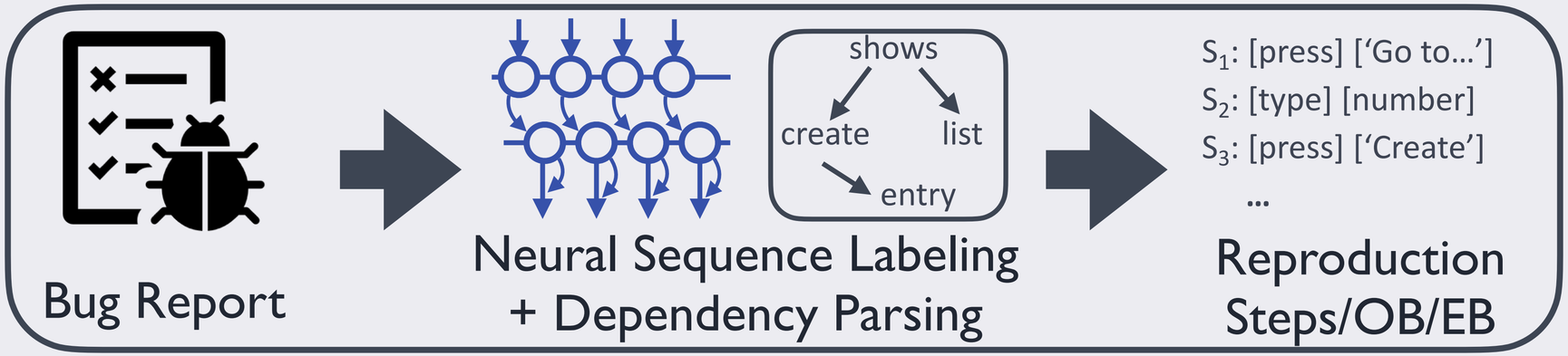
BURT and EBUG: Enabling Interactive Bug Reporting



BURT and EBUG: Component Workflow

BURT and EBUG: Component Workflow

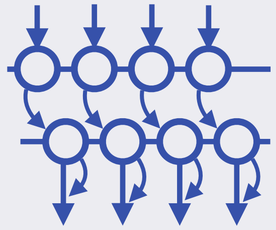
1) Bug Element Identification

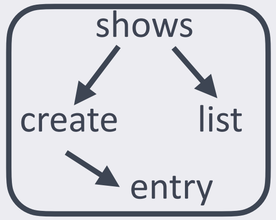


BURT and EBUG: Component Workflow

1) Bug Element Identification

Bug Report





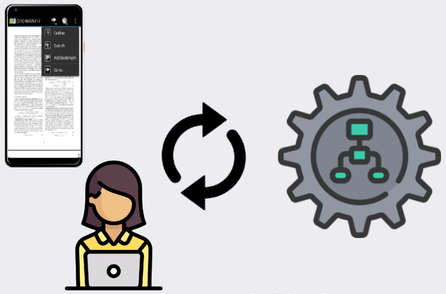
Neural Sequence Labeling
+ Dependency Parsing

S_1: [press] [‘Go to…’]
S_2: [type] [number]
S_3: [press] [‘Create’]
…

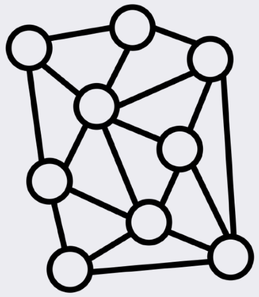
Reproduction
Steps/OB/EB

2) Execution Model Generation

Android App



Automated/Manual
Exploration



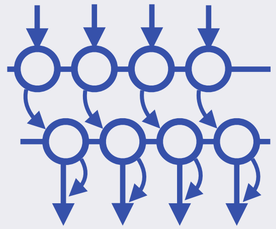
Execution Model(s)

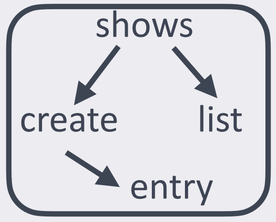
UCF

BURT and EBUG: Component Workflow

1) Bug Element Identification

Bug Report





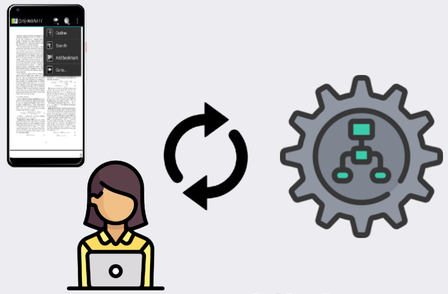
Neural Sequence Labeling
+ Dependency Parsing

S_1: [press] [‘Go to…’]
S_2: [type] [number]
S_3: [press] [‘Create’]
…

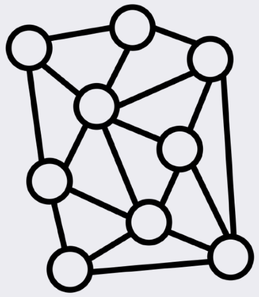
Reproduction
Steps/OB/EB

2) Execution Model Generation

Android App



Automated/Manual
Exploration



Execution Model(s)

3)Report Processing

Dialogue Quality Processor

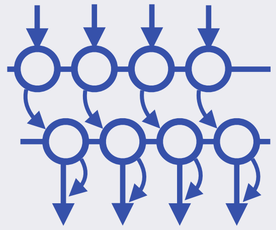
Bug Element Response Predictor

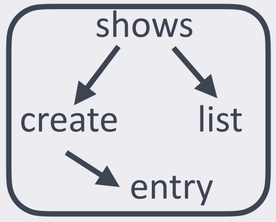
UCF

BURT and EBUG: Component Workflow

1) Bug Element Identification

Bug Report





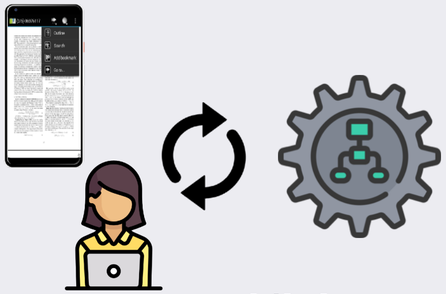
Neural Sequence Labeling
+ Dependency Parsing

S_1: [press] ['Go to…']
S_2: [type] [number]
S_3: [press] ['Create']
…

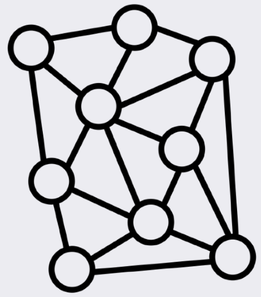
Reproduction Steps/OB/EB

2) Execution Model Generation

Android App



Automated/Manual Exploration

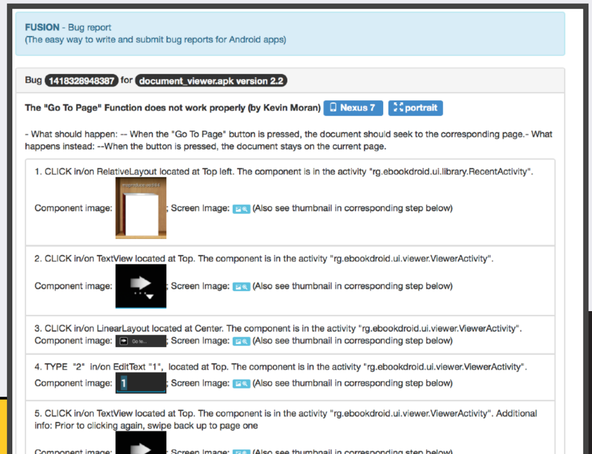


Execution Model(s)

3)Report Processing

Dialogue Quality Processor

Bug Element Response Predictor



FUSION - Bug report
(The easy way to write and submit bug reports for Android apps)

Bug 1418328948387 for document_viewer.apk version 2.2

The "Go To Page" Function does not work properly (by Kevin Moran) Nexus 7 portrait

- What should happen: -- When the "Go To Page" button is pressed, the document should seek to the corresponding page.- What happens instead: --When the button is pressed, the document stays on the current page.

1. CLICK in/on RelativeLayout located at Top left. The component is in the activity "rg.ebookdroid.ui.library.RecentActivity".
Component image: Screen Image: (Also see thumbnail in corresponding step below)

2. CLICK in/on TextView located at Top. The component is in the activity "rg.ebookdroid.ui.viewer.ViewerActivity".
Component image: Screen Image: (Also see thumbnail in corresponding step below)

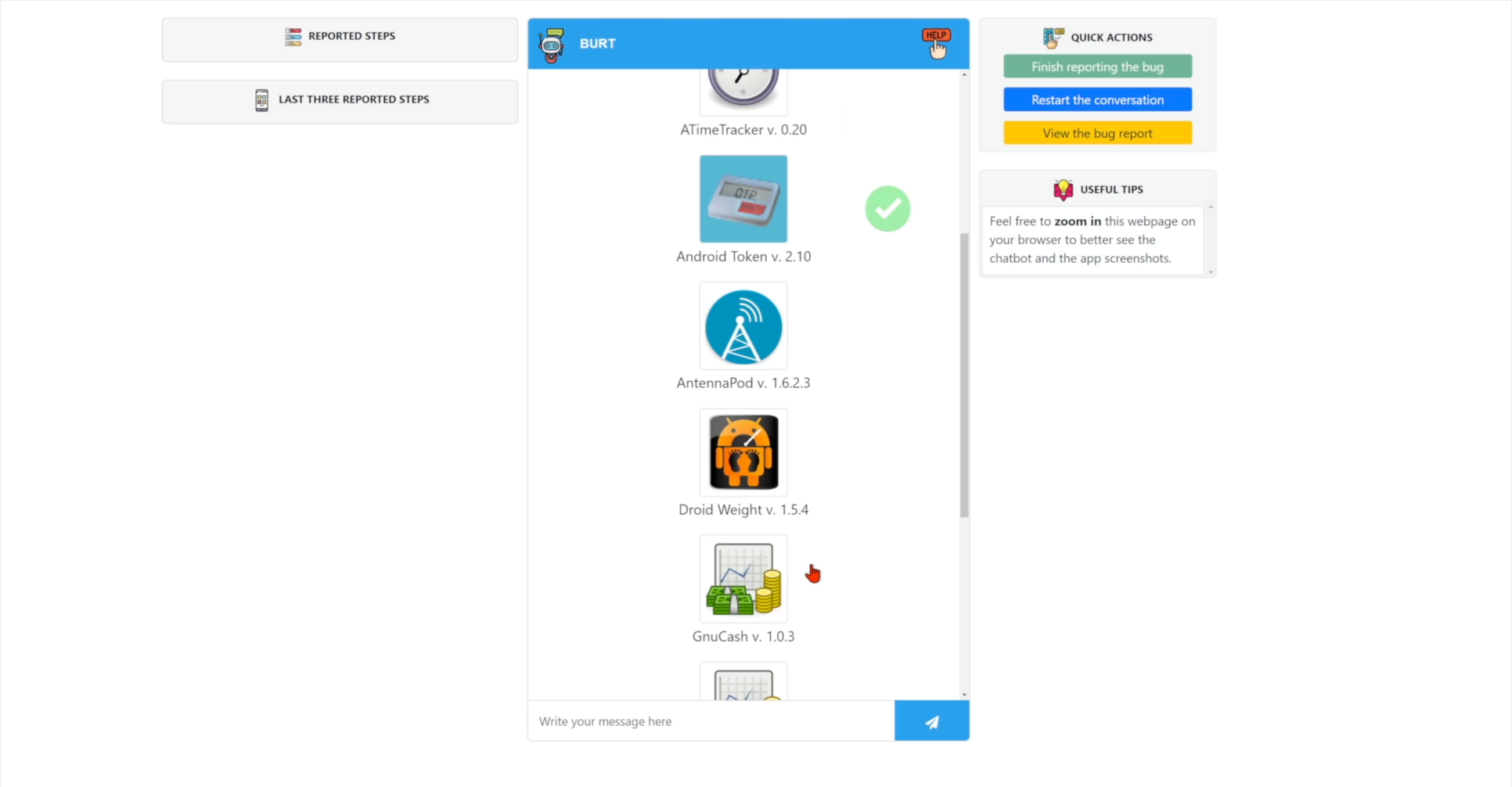
3. CLICK in/on LinearLayout located at Center. The component is in the activity "rg.ebookdroid.ui.viewer.ViewerActivity".
Component image: Screen Image: (Also see thumbnail in corresponding step below)

4. TYPE "2" in/on EditText "1", located at Top. The component is in the activity "rg.ebookdroid.ui.viewer.ViewerActivity".
Component image: Screen Image: (Also see thumbnail in corresponding step below)

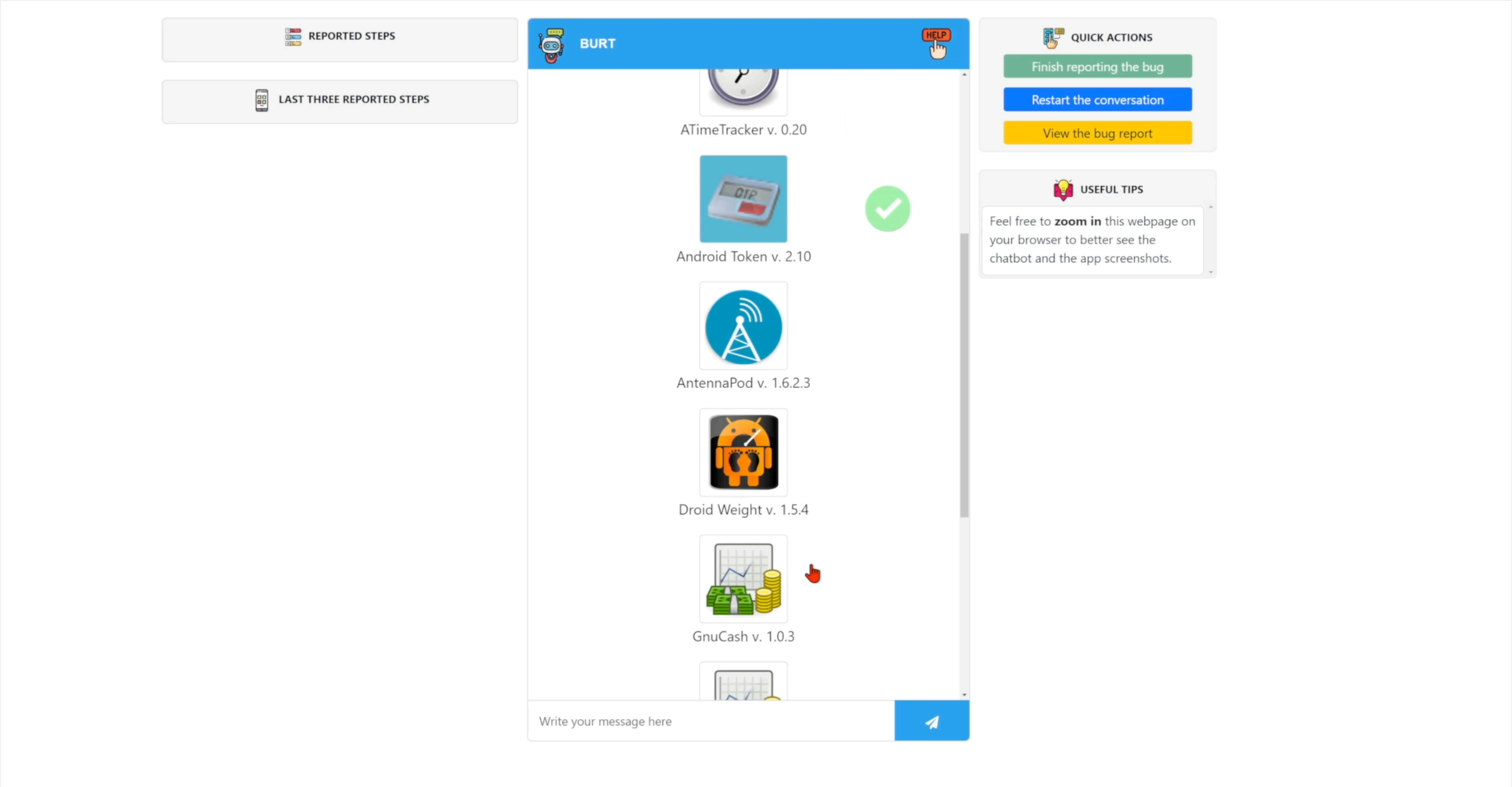
5. CLICK in/on TextView located at Top. The component is in the activity "rg.ebookdroid.ui.viewer.ViewerActivity". Additional info: Prior to clicking again, swipe back up to page one
Component image: Screen Image: (Also see thumbnail in corresponding step below)

UCF

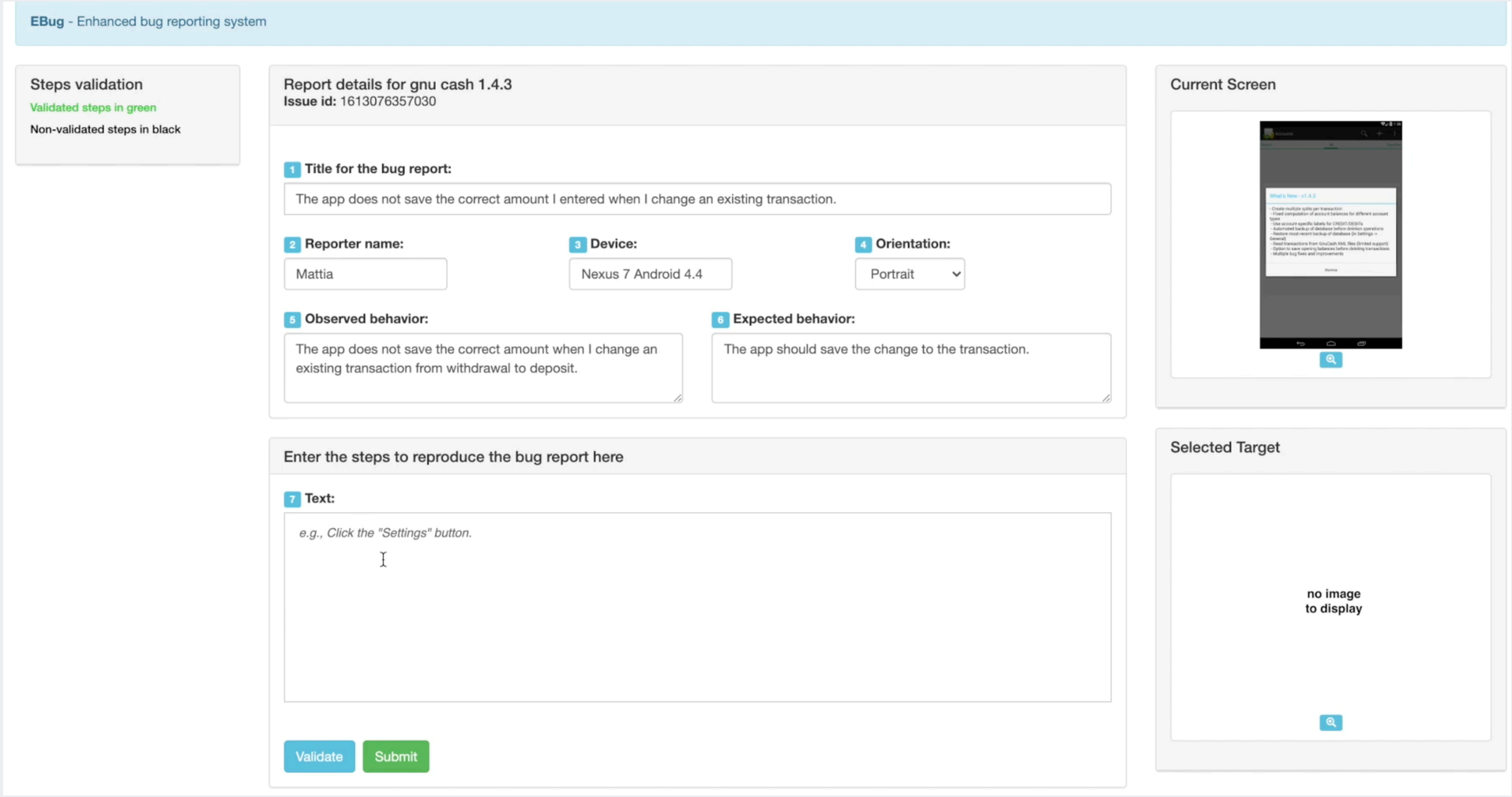
BURT and EBUG: Demo



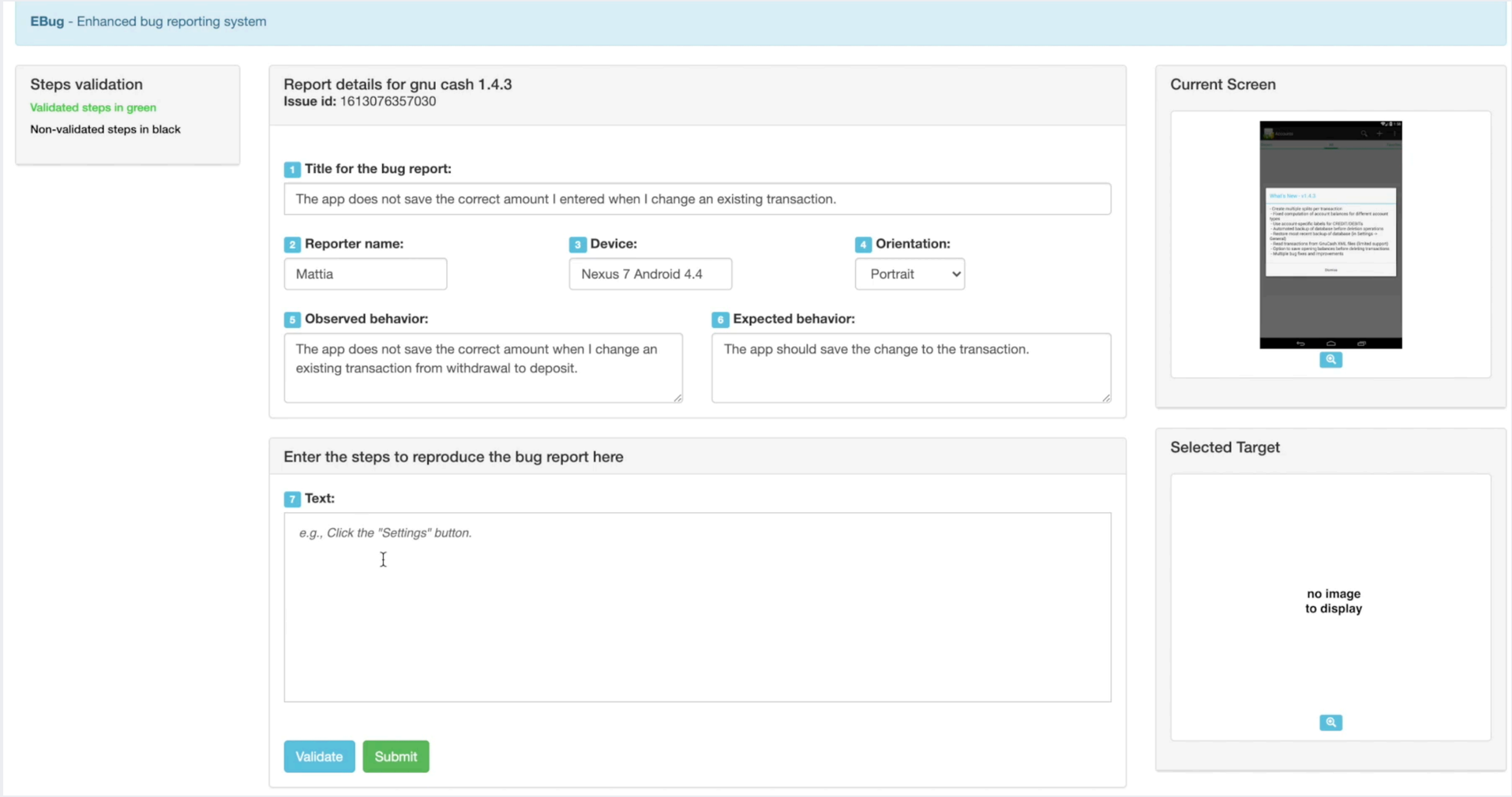
BURT and EBUG: Demo



BURT and EBUG: Demo



BURT and EBUG: Demo



Overview of Research Areas

Automated UI Analysis

Research Title:

Bug Reporting 2.0

- ***Motivation***
  - Increasing Complexity of Software
  - Issue Trackers have not evolved in two decades
  - Empirical Studies illustrate bug reports are often low-quality
  - Tasks related to bug report management dominate Software Maintenance

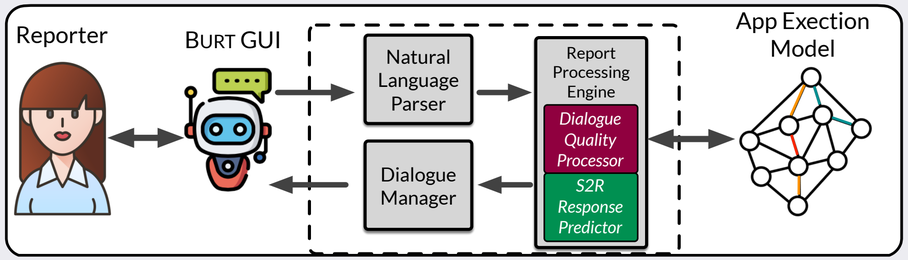


Fig. 1: Overview of the BURT Bug Reporting System

Objectives/Approach

Impact, Outcomes, and Achievements

UCF

Overview of Research Areas

Automated UI Analysis

Research Title: Bug Reporting 2.0

- ***Motivation***
  - Increasing Complexity of Software
  - Issue Trackers have not evolved in two decades
  - Empirical Studies illustrate bug reports are often low-quality
  - Tasks related to bug report management dominate Software Maintenance

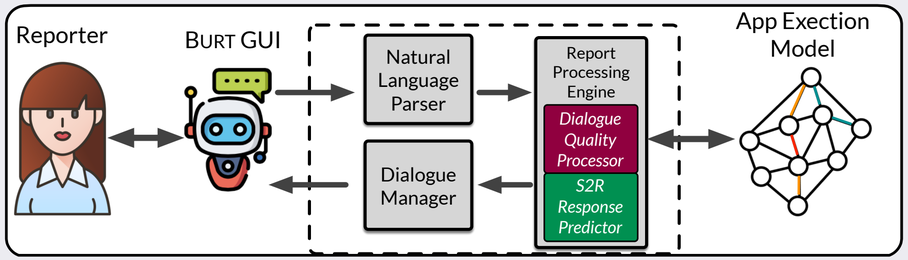


Fig. 1: Overview of the BURT Bug Reporting System

Objectives/Approach

- ***Objectives:***
  - Develop Techniques to *extract information* from software applications, and use this to automate the reporting process
  - Automate tasks related to *bug report management* — including reporting, triaging, localization, and repair
- ***Approach:***
  - Extract information code and UIs using *LLMs* specifically trained for code and *multi-modal computer vision* models
  - Use these models to help *automate* the tasks above

Impact, Outcomes, and Achievements

UCF

Overview of Research Areas

Automated UI Analysis

Research Title: Bug Reporting 2.0

- ***Motivation***
  - Increasing Complexity of Software
  - Issue Trackers have not evolved in two decades
  - Empirical Studies illustrate bug reports are often low-quality
  - Tasks related to bug report management dominate Software Maintenance

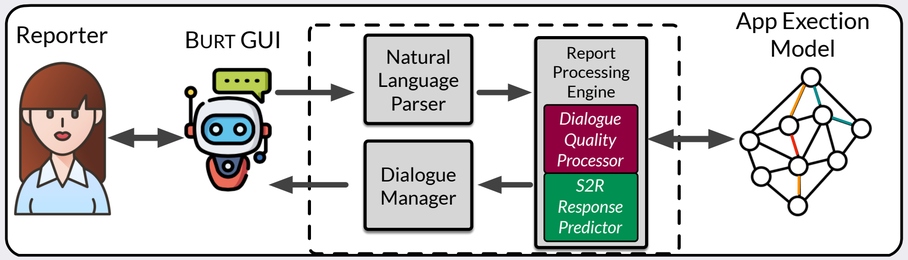


Fig. 1: Overview of the BURT Bug Reporting System

Objectives/Approach

- ***Objectives:***
  - Develop Techniques to *extract information* from software applications, and use this to automate the reporting process
  - Automate tasks related to *bug report management* — including reporting, triaging, localization, and repair
- ***Approach:***
  - Extract information code and UIs using *LLMs* specifically trained for code and *multi-modal computer vision* models
  - Use these models to help *automate* the tasks above

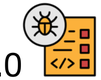
Impact, Outcomes, and Achievements

- ***Academic Achievements:***
  - 16 publications, 1 distinguished paper award
  - 7 Research Prototypes and 1 Dataset developed (open source)
- ***Practical Impact:***
  - Our reporting tools were tested with real developers and were shown to *improve the quality* of reports
  - Our recent work on bug triaging and localization currently represent *state-of-the-art* approaches in the field

UCF

Overview of Research Areas

Automated UI Analysis 

Bug Reporting 2.0 

Research Title: Dev Tools for Security 

- ***Motivation***
  - Increasing Complexity of Software
  - Developers need automated tools to assist in *security assurance*
  - It's not clear how much we can *rely* on existing tools
  - Some domains (e.g., IoT) *do not have mature tools*

Objectives/Approach

Impact, Outcomes, and Achievements

UCF

Overview of Research Areas

Automated UI Analysis

Bug Reporting 2.0

Research Title:

Dev Tools for Security

- ***Motivation***
  - Increasing Complexity of Software
  - Developers need automated tools to assist in *security assurance*
  - It's not clear how much we can *rely* on existing tools
  - Some domains (e.g., IoT) *do not have mature tools*

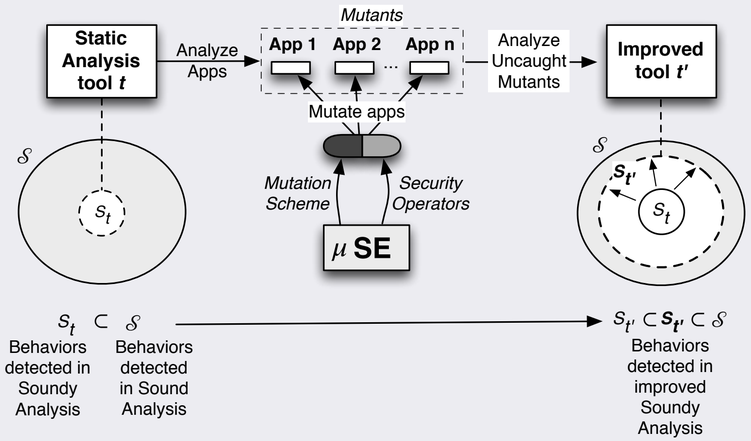


Fig. 1: Overview of the μSE system for improving SASTs

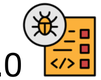
Objectives/Approach

Impact, Outcomes, and Achievements

UCF

Overview of Research Areas

Automated UI Analysis 

Bug Reporting 2.0 

Research Title: Dev Tools for Security 

- ***Motivation***
  - Increasing Complexity of Software
  - Developers need automated tools to assist in *security assurance*
  - It's not clear how much we can *rely* on existing tools
  - Some domains (e.g., IoT) *do not have mature tools*

Objectives/Approach

Impact, Outcomes, and Achievements

UCF

Overview of Research Areas

Automated UI Analysis

Bug Reporting 2.0

Research Title:

Dev Tools for Security

- ***Motivation***
  - Increasing Complexity of Software
  - Developers need automated tools to assist in *security assurance*
  - It's not clear how much we can *rely* on existing tools
  - Some domains (e.g., IoT) *do not have mature tools*

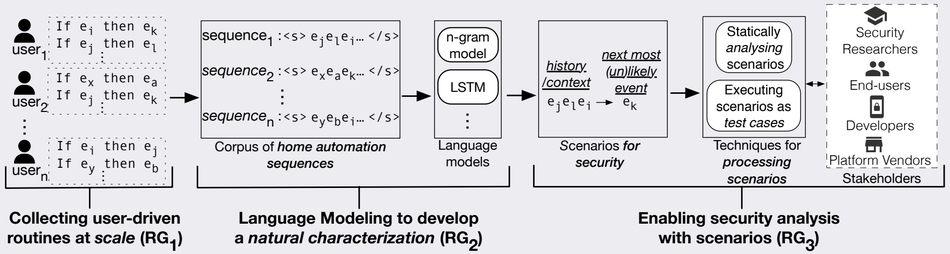


Fig.2: Overview of the Helion System for Improving IoT Security

Objectives/Approach

Impact, Outcomes, and Achievements

UCF

Overview of Research Areas

Automated UI Analysis

Bug Reporting 2.0

Research Title: Dev Tools for Security

- ***Motivation***
  - Increasing Complexity of Software
  - Developers need automated tools to assist in *security assurance*
  - It's not clear how much we can *rely* on existing tools
  - Some domains (e.g., IoT) *do not have mature tools*

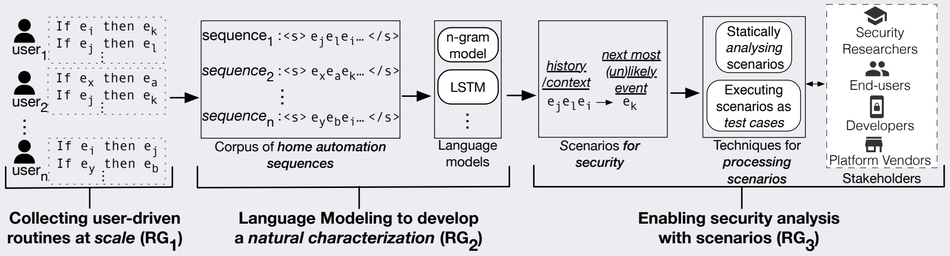


Fig.2: Overview of the Helion System for Improving IoT Security

Objectives/Approach

- ***Objectives:***
  - Develop Techniques to evaluate and improve existing *security-focused static analysis tools* for code
  - Improve and create new *developer-focused security tools* for general software projects and IoT-related projects
- ***Approach:***
  - Adapting techniques from *software testing* (mutation analysis) to measure the soundness of static analysis tools
  - Create *models of IoT product usage* by end users, and use this to develop threat models and new tools for security and privacy

Impact, Outcomes, and Achievements

Overview of Research Areas

Automated UI Analysis

Bug Reporting 2.0

Research Title: Dev Tools for Security

- ***Motivation***
  - Increasing Complexity of Software
  - Developers need automated tools to assist in *security assurance*
  - It's not clear how much we can *rely* on existing tools
  - Some domains (e.g., IoT) *do not have mature tools*

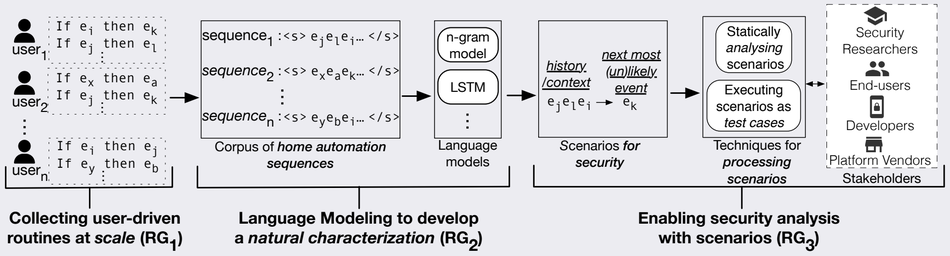


Fig.2: Overview of the Helion System for Improving IoT Security

Objectives/Approach

- ***Objectives:***
  - Develop Techniques to evaluate and improve existing *security-focused static analysis tools* for code
  - Improve and create new *developer-focused security tools* for general software projects and IoT-related projects
- ***Approach:***
  - Adapting techniques from *software testing* (mutation analysis) to measure the soundness of static analysis tools
  - Create *models of IoT product usage* by end users, and use this to develop threat models and new tools for security and privacy

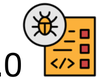
Impact, Outcomes, and Achievements

- ***Academic Achievements:***
  - 9 publications, 1 best paper award
  - 4 Research Prototypes and 1 Dataset developed (open source)
- ***Practical Impact:***
  - Found *3 critical security vulnerabilities* in smart home platforms which have since been patched
  - Identified and fixed numerous flaws/vulnerabilities with *industrial-grade crypto-detectors*
  - Developed a model for *practitioner needs* related to security-focused static analysis tools

UCF

Overview of Research Areas

Automated UI Analysis 

Bug Reporting 2.0 

Dev Tools for Security 

Research Title: Interpretability for Code LLMs 

- ***Motivation***
  - LLMs are increasingly being used in software development
  - As much as ***40%*** of recent production code at Google is ***AI-generated***!!
  - Developers cannot always ***trust*** these AI-powered tools
  - ***Interpretability techniques*** can help developers build trust in these tools

Objectives/Approach

Impact, Outcomes, and Achievements

UCF

Overview of Research Areas

Automated UI Analysis

Bug Reporting 2.0

Dev Tools for Security

Research Title:

Interpretability for Code LLMs

- ***Motivation***
  - LLMs are increasingly being used in software development
  - As much as ***40%*** of recent production code at Google is ***AI-generated***!!
  - Developers cannot always ***trust*** these AI-powered tools
  - ***Interpretability techniques*** can help developers build trust in these tools

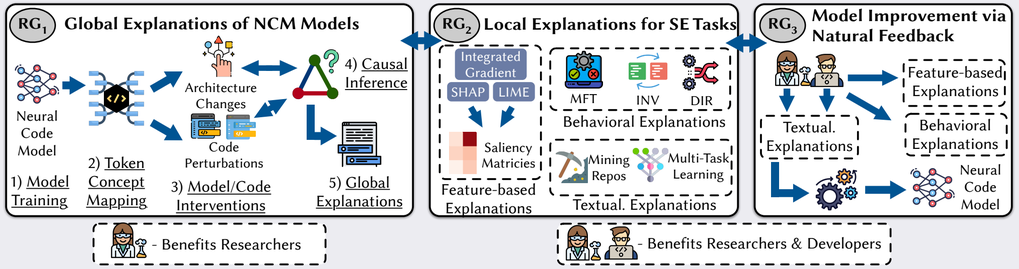


Fig.1: Overview Code LLM Interpretability Work

Objectives/Approach

Impact, Outcomes, and Achievements

UCF

Overview of Research Areas

Automated UI Analysis

Bug Reporting 2.0

Dev Tools for Security

Research Title: Interpretability for Code LLMs

- ***Motivation***
  - LLMs are increasingly being used in software development
  - As much as ***40%*** of recent production code at Google is ***AI-generated***!!
  - Developers cannot always ***trust*** these AI-powered tools
  - ***Interpretability techniques*** can help developers build trust in these tools

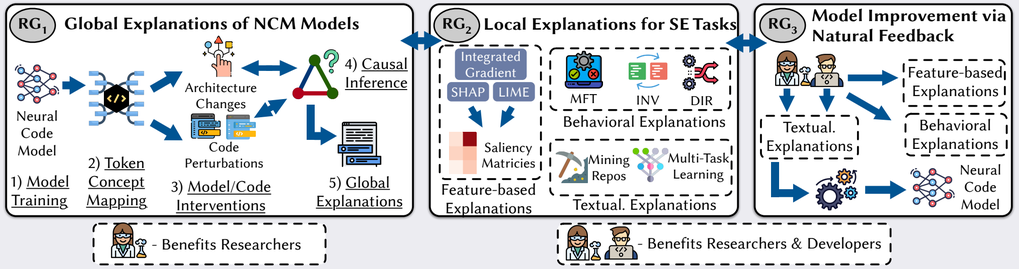


Fig.1: Overview Code LLM Interpretability Work

Objectives/Approach

- ***Objectives:***
  - Develop a *causal inference-based* approach for interpreting Neural Models of Source Code
  - Adapt *explainability techniques* developed in NLP to provide explanations for model predictions in task-specific settings
- ***Approach:***
  - Build an intervention-based causal inference model for *mapping model predictions to code concepts*
  - Adapt techniques such as SHAP, LIME, and integrated gradients and customize them to *explain Code LLM predictions*

Impact, Outcomes, and Achievements

UCF

Overview of Research Areas

Automated UI Analysis

Bug Reporting 2.0

Dev Tools for Security

Research Title: Interpretability for Code LLMs

- ***Motivation***
  - LLMs are increasingly being used in software development
  - As much as ***40%*** of recent production code at Google is ***AI-generated***!!
  - Developers cannot always ***trust*** these AI-powered tools
  - ***Interpretability techniques*** can help developers build trust in these tools

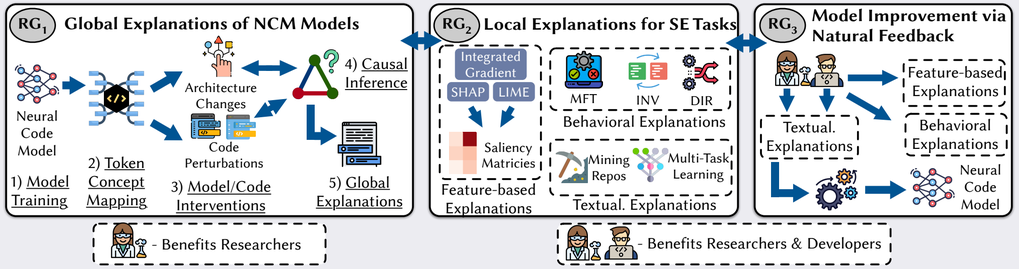


Fig.1: Overview Code LLM Interpretability Work

Objectives/Approach

- ***Objectives:***
  - Develop a *causal inference-based* approach for interpreting Neural Models of Source Code
  - Adapt *explainability techniques* developed in NLP to provide explanations for model predictions in task-specific settings
- ***Approach:***
  - Build an intervention-based causal inference model for *mapping model predictions to code concepts*
  - Adapt techniques such as SHAP, LIME, and integrated gradients and customize them to *explain Code LLM predictions*

Impact, Outcomes, and Achievements

- ***Academic Achievements:***
  - 3 publications under review
- ***Practical Impact:***
  - Illustrated that our proposed interpretability framework can lead to improvements in model accuracy through better understanding

UCF

Overview of Research Areas

Automated UI Analysis

Bug Reporting 2.0

Dev Tools for Security

Interpretability for Code LLMs

Research Title:

- ***Motivation***
  - LLMs are increasingly being used in software development
  - As much as ***40%*** of recent production code at Google is ***AI-generated***!!
  - Developers cannot always ***trust*** these AI-powered tools
  - ***Interpretability techniques*** can help developers build trust in these tools

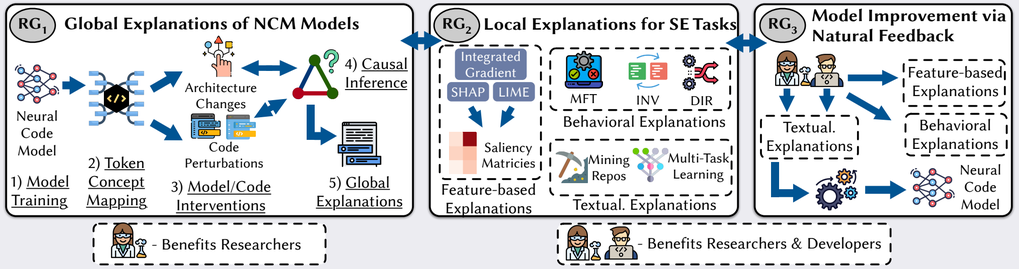


Fig.1: Overview Code LLM Interpretability Work

Objectives/Approach

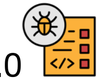
- ***Objectives:***
  - Develop a causal inference-based approach for interpreting Neural Models of Source Code
  - Adapt explainability techniques developed in NLP to provide explanations for model predictions in task-specific settings
- ***Approach:***
  - Build an intervention-based causal inference model for mapping model predictions to code concepts
  - Adapt techniques such as SHAP, LIME, and integrated gradients and customize them to explain Code LLM predictions

Impact, Outcomes, and Achievements

- ***Academic Achievements:***
  - 3 publications under review
- ***Practical Impact:***
  - Illustrated that our proposed interpretability framework can lead to improvements in model accuracy through better understanding

UCF

Future Work and Points of Collaboration

- Automated UI Analysis 
- Bug Reporting 2.0 
- Dev Tools for Security 
- Interpretability for Code LLMs 

Future Work and Points of Collaboration

- ***Future Work (Short Term):***
  - Continue agendas for presented projects
  - Focus on *human studies* with developers
  - Work towards *automating UI accessibility* compliance
- ***Future Work (Long Term):***
  - Use machine learning to aid in software development of *emerging computing platforms* (e.g., AR/VR)
  - Use *interpretability techniques* to build AI-powered tools that developers can seamlessly interact with
  - Develop the next generation of *security-focused software tools* to help developers focus on writing code that is secure by design
- ***Potential Points for Collaboration:***
  - Human studies of programmers
  - Software Accessibility
  - Computer security topics centered around engineering
  - Applications for machine learning and computer vision

Acknowledgements

Junayed Mahmud

Arun Krishnavajjala

SM Hasan Mansur

Safwat Ali Khan

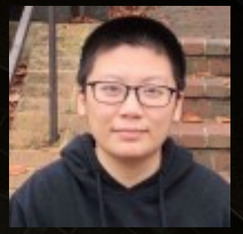
Sabiha Salma

Atish Dipongkor

Acknowledgements

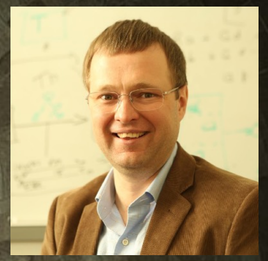
Carlos Bernal Cardenas (Microsoft)

Michele Tufano (Microsoft)

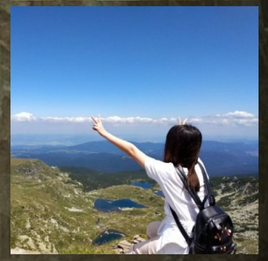


Yang Song (W&M)

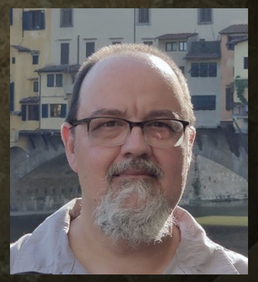
Oscar Chaparro (W&M)



Denys Poshyvanyk (W&M)



Ying Zhou (UT Dallas)



Andi Marcus (UT Dallas)

Mattia Fazzini (Minnesota)

Tyler Wendland (Minnesota)

Alex Orso (GA Tech)

Kevin Moran
University of Central Florida
@kevpmo
kpmoran@ucf.edu
https://www.kpmoran.com

Thank you!

Any Questions?