



運用大型語言模型之軟體錯誤回報自動重現應用

Automatically Reproducing Bug Reports based on LLM

指導教授：李信杰 專題成員：李沛宸

E2-軟體工程、資訊安全、計算機理論、智慧製造



AGENDA

1 專題背景與目的

2 專題簡介

3 系統架構圖

4 成果展示

5 特色與創新之處

6 結論與未來發展

專題背景與目的

Bug Report 資訊經常不完整

2023 年有研究分析了 180 個真實 Bug Reports，發現：

- 環境資訊經常缺失
- Steps to Reproduce 常不完整
- 許多描述不夠具體

➔ 導致 Bug 難以被人工或自動化系統重現

Bug 重現是修復必要前置步驟

Bug Reproduction 是錯誤定位與錯誤修復的必要前置工作，而人工撰寫重現腳本耗時且高度依賴開發者經驗 (FSE 2025)

➔ 若無法穩定重現 Bug，後續除錯與修復工作將難以進行

LLM 具備強大理解與推理能力

- LLM 已廣泛應用於問答、程式碼生成等領域
- 2024-2026 有研究開始探討 AI 自動重現 Bug

➔ AI Agent 成功重現錯誤，仍是目前的重要研究議題

建構 AI Agent 驅動之 Bug 自動重現系統，使 AI 能根據 Bug Report 自動理解、規劃並執行操作流程，降低人工重現成本並提升軟體測試效率



專題簡介

- 結合 **Gemini CLI**與 **Playwright**
- 建構 **AI Agent** 驅動之 **Bug** 自動重現系統
- 採用 **Tool-based Agent Architecture**，並結合頁面狀態擷取機制，根據當前介面資訊動態規劃與執行操作流程
- 透過 **Observe-Reason-Act** 決策循環動態規劃操作流程
- 以 **Plane Issue #8643** 驗證系統可行性



專題簡介

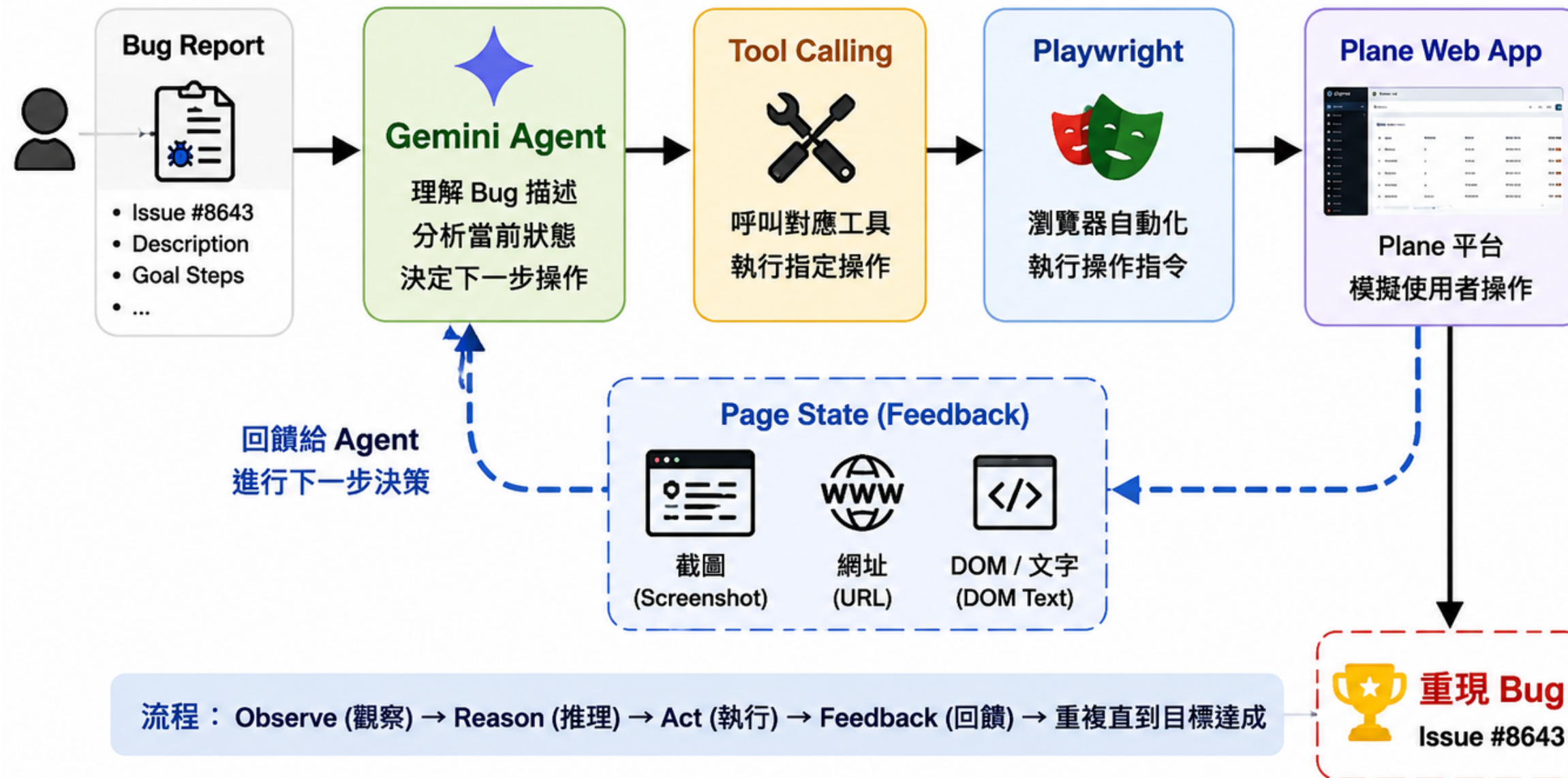
本研究以 Plane 平台中的實際錯誤案例（Issue #8643）作為驗證對象

1. 將一個 Module 指派給某個 Work Item
2. 將該 Module 進行 Archive
3. 前往 Work Items → Group By → Modules
4. 該 Work Item 已經無法在任何欄位中看到
5. 切換至 Group By → Labels
6. 該 Work Item 又重新出現

The screenshot displays the Plane platform's Work Items interface. The top section shows a list of work items grouped by 'None' (2 items). The bottom section shows the same list grouped by 'Labels' (3 items). In the bottom section, the work item 'Bug Test 160911419' is highlighted with a red circle. The 'Group by' dropdown menu is open, and 'Labels' is selected, also highlighted with a red circle. The 'Display Properties' panel on the right shows various filters and sorting options.

Group	Work Item ID	Module
None (2)	BUGREPRODU-127	B
	BUGREPRODU-126	A
Labels (3)	BUGREPRODU-128	Bug Test 160911419
	BUGREPRODU-127	B
	BUGREPRODU-126	A

系統架構圖



- **動態決策式 Agent Loop**：採用 Observe–Reason–Act 架構，根據頁面狀態動態規劃下一步操作
- **Tool-based Action Planning**：將系統操作封裝為高階工具，透過 Tool Calling 完成自然語言到操作流程的轉換。
- **閉環式狀態回饋機制**：持續蒐集 Screenshot、URL 與 DOM 資訊作為決策依據，提升重現流程穩定性



成果展示

A screenshot of the Visual Studio Code (VS Code) editor interface. The top bar shows the search bar with "bug_reproduction" and a file explorer on the left. The Explorer sidebar shows a project named "BUG_REPRODUCTION" with files like "__pycache__", ".venv", "ai_agent_repro.py", "ai.py", "issue_8643_repro (1).py", "issue_8643_repro_final.py", "issue_8643_repro_fixed.py", and "issue_8643_repro.py". The main editor area shows the "ai_agent_repro.py" file with a line number 199 and the code "main()". The bottom panel shows the "TERMINAL" tab with the following output:

```
=== AI Agent End ===  
[AI Suggest] finish  
[AI Explain] finish  
The reproduction steps are complete. The work item was created, assigned to a module, the module was completed and archived, and then the vie  
w was toggled between grouping by module (where the item should disappear) and grouping by labels (where the item should reappear).  
[EXECUTE] finish  
=== AI Agent End ===  
(.venv) (base) lipeichen@lipeichendeMacBook-Air bug_reproduction % python ai_agent_repro.py
```

The status bar at the bottom indicates the current file is "ai_agent_repro.py" at line 199, column 11, using Python 3.14.3 in the ".venv" environment.



特色與創新之處

真實 Web Application 驗證

- 以真實 SaaS平台 Plane 實際作為驗證案例
- 非模擬情境或簡化測試案例
- 驗證 AI Agent 於真實軟體環境中的可行性



整合 LLM 與 Browser Automation

- 結合 Gemini、Tool Calling 與 Playwright
- 建立從 Bug Report 到實際操作的完整流程
- 實現自然語言到系統操作的轉換

複合型 Workflow Bug 重現

- 涉及 Module、Work Item、Archive 等多項功能
- 需經過跨頁面、多步驟操作才能觸發
- 不同於單一步驟 UI 測試案例



結論與未來發展

研究成功整合 Gemini、Tool Calling 與 Playwright，建構 AI Agent 驅動之軟體錯誤回報自動重現系統。未來或許能進一步支援 GitHub Issues、Jira 與 Bugzilla 等不同錯誤回報平台，並強化 Agent 在自主決策與錯誤恢復能力，提升面對複雜真實情境時的錯誤重現效率與穩定性



謝謝聆聽

指導教授：李信杰 專題成員：李沛宸

E2-軟體工程、資訊安全、計算機理論、智慧製造