

AUTOMATED GUI TESTING

Juraj Melo

Master Degree Programme (3), FIT BUT

E-mail: xmeloj01@stud.fit.vutbr.cz

Supervised by: Aleš Smrčka

E-mail: smrcka@fit.vutbr.cz

Abstract: The paper describes a system for automated GUI testing using assistive technologies for accessing and manipulating GUI elements. The only input from the user to automated test system is a description of UI events and activities. For this purpose, a specialized language is proposed. The test system then automatically generates possible sequences of UI events applying a given test criterion. Generated test set is executed by Python interpreter exploiting the Linux Desktop Testing Project (LDTP). Test system proposed in the paper then provides reports for particular test cases and an overall coverage evaluation.

Keywords: automated GUI testing, LDTP, GUI specification

1 ÚVOD

Grafické uživatelské rozhrania, ďalej označované ako GUI, tvoria kľúčovú časť moderných softvérových aplikácií. Podľa [1] tvorí 45 – 60 % zdrojového kódu aplikácie práve implementácia funkcionality GUI. Komplexné testovanie softvérových produktov sa teda nezaobíde bez testovania ich rozhrania.

GUI aplikácie je zložené z hierarchicky usporiadaných, grafických objektov – angl. widget. Interakcia používateľa s GUI generuje udalosti, ktoré môžu byť ošetrené obslužnými metódami. Každá udalosť môže potenciálne zmeniť stav aplikácie, alebo GUI. Testovanie GUI je zložitejšie, ako testovanie konvenčného softvéru, pretože obsahuje veľké množstvo vstupov, ktoré môžu spôsobiť enormný nárast stavového priestoru aplikácie. Dôležité je tiež otestovať rôzne sekvencie udalostí, keďže môžu viesť k odlišným stavom aplikácie. [1]

Zrejme najstaršou metódou testovania GUI aplikácií je record-playback. Jedná sa o zaznamenávanie interakcie testera s GUI, jej uloženie a následné opakované prehrávanie nad testovaným GUI [4]. Ďalší princíp je použiteľný naprieč všetkými aplikáciami zobraziteľnými na obrazovke, a je založený na vyhľadávaní grafických vzorov a popise akcií, ktoré majú byť na nich vykonané [5]. Pri testovaní GUI popísanom v tomto článku, sú použité asistenčné technológie, ktorými je možné získať informácie o štruktúre testovaného GUI, a tiež priamo ovládať jeho jednotlivé časti [3].

2 SYSTÉM PRE AUTOMATIZOVANÉ TESTOVANIE GUI

2.1 POPIS A ŠTRUKTÚRA SYSTÉMU

Systém pre automatizované testovanie GUI využíva Linux Desktop Testing Project [2], ďalej označovaný ako LDTP, ktorý vo forme knižnice pre jazyk Python, dokáže využiť asistenčné technológie pre prístup a ovládanie GUI počas behu testovanej aplikácie.

Automatizovaný testovací systém, navrhnutý v tomto článku, umožňuje testovať aplikácie založené na prostredí GNOME, ktoré majú povolené využívanie asistenčných technológií. Jeho štruktúra je znázornená na obrázku 1. Testovanie začína tým, že tester vytvorí súbor obsahujúci popis testovacej

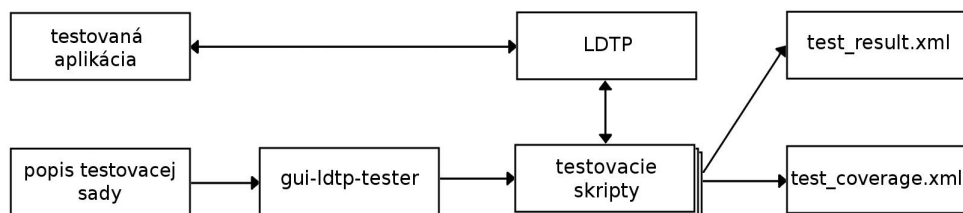
sady. Jazyk vytvorený v rámci tejto práce, umožňuje popísať testovaciu sadu vo forme aktivít. Aktivita zapuzdruje niekoľko činností, ktoré spolu súvisia, a môže byť volaná z ostatných aktivít. Týmto spôsobom je umožnené hierarchicky popísať testovanie GUI.

Aktivita sa skladajú z udalostí, ktoré reprezentujú podmnožinu príkazov knižnice LDTP, ako napríklad: kliknutie, vpísanie textu a iné. Ďalej je možné zisťovať stav a vlastnosti jednotlivých grafických prvkov. Jazyk tiež umožňuje definovanie podmienok, ktorých nesplnenie znamená objavenie chyby v testovanej aplikácii. Vtedy sa nájdená chyba zaznamená a ukončí sa testovanie aktuálneho testovacieho prípadu.

V jazyku pre popis testovacej sady je možné špecifikovať parametre a počet opakovaných spustení aplikácie. Je možné nastaviť premenné prostredia, v ktorom aplikácia pobeží, a parametre s ktorými bude spustená. Každé spustenie aplikácie s vygenerovanou kombináciou parametrov, testované jedným z vygenerovaných testovacích skriptov, tvorí samostatný testovací prípad. Popis testovacej sady spolu so spustiteľným súborom testovanej aplikácie, tvorí vstup pre testovanie GUI. Modul `gui-ldtp-tester` spracuje popis testovacej sady a na jeho základe vygeneruje niekoľko testovacích skriptov zapísaných v jazyku Python, ktoré implementujú samotné testovanie.

Po vygenerovaní testovacej sady ju začne modul `gui-ldtp-tester` vykonávať. Otestovanie aplikácie podľa jedného testovacieho prípadu, začína vygenerovaním parametrov testovanej aplikácie a jej následným spustením. Potom sa vyberie príslušný testovací skript, ktorý je vykonaný interpretom jazyka Python a s využitím knižnice LDTP (modul LDTP) vykoná samotné testovanie. Na záver sa testovaná aplikácia ukončí.

Počas testovania sa vytvoria XML súbory obsahujúce výsledok testovania a vyhodnotenie pokrytia testovania. Výsledok testovania v súbore `test_results.xml` obsahuje informácie o tom, aké mali jednotlivé testovacie prípady parametre, ktoré aktivity boli v rámci testu spustené, ktoré podmienky boli splnené a informácie o prípadných chybách nájdených počas testovania.



Obrázek 1: Štruktúra systému

2.2 AUTOMATIZOVANÉ GENEROVANIE TESTOVACEJ SADY

Súčasťou testovacieho systému je automatické generovanie kombinácií udalostí a aktivít. K tomuto účelu bol navrhnutý jazyk pre popis testovacích sád. Z dôvodu malého priestoru v tomto článku budú popísané len vybrané časti navrhnutého jazyka. Popis testovacej sady môže vyzeráť nasledovne:

```

activity main {
  generate-set random-data-criterion 2 {
    set-environment "LANG" = {"en_US.UTF-8", "cs_CZ.UTF-8"}
    run "/usr/bin/gedit"
  }
  on "*-gedit*"
  generate-set sequence-criterion * {
    do TestMenuOpen
    do TestMenuSaveAs } }

```

Príkaz `generate-set` slúži na generovanie sekvencií udalostí podľa zadaného kritéria. Navrhnutý systém podporuje tri rôzne kritéria. Kritérium `event-criterion` vygeneruje sekvencie udalostí tak, aby každá sekvencia obsahovala práve jednu udalosť zo zoznamu. Kritérium `sequence-criterion` s parametrom `n`, vygeneruje všetky `n`-tice z daných udalostí tak, že každá sekvencia bude obsahovať práve jednu `n`-ticu.

Posledné generovacie kritérium `random-data-criterion` je použiteľné pre vstupné parametre aplikácie. V jeho tele je špecifikovaný zoznam hodnôt, ktoré môže daný parameter nadobúdať. Takto je možné nastaviť premenné prostredia, alebo parametre testovanej aplikácie. Kritérium obsahuje parameter, udávajúci počet opakovaných spustení aplikácie. Hodnoty parametrov sa náhodne vygenerujú, z príslušného zoznamu hodnôt, pred každým spustením aplikácie. Tomuto kritériu je možné zadať parameter `*`. V takom prípade nie je dopredu stanovený počet testovacích prípadov, pretože aplikácia sa bude neustále testovať s náhodnými hodnotami vstupných parametrov, až pokým systém neobdrží signál `SIGHUP`, `SIGINT`, alebo `SIGTERM`.

2.3 VYHODNOTENIE POKRYTIA

V systéme sú definované dve kritériá pokrytia testovania. Prvým je vykonanie aktivít, ktoré je plné v prípade, že boli vykonané všetky aktivity uvedené v popise testovacej sady. Druhým kritériom je otestovanie grafických prvkov. Vtedy je pokrytie plné, ak boli otestované všetky, testovaciemu systému viditeľné, grafické prvky. Vyhodnotenie pokrytia je uložené do súboru `test_coverage.xml`.

3 ZÁVER

Táto práca ukazuje spôsob testovania GUI s využitím asistenčných technológií. Poskytuje možnosti pre automatizované generovanie testovacej sady, jej vykonanie a vyhodnotenie výsledkov a pokrytia testovania podľa definovaných kritérií pokrytia. Testovanie štandardných aplikácií z prostredia GNOME odhalilo drobné nedostatky v implementácií rozhrania pre asistenčné technológie. Prípadné rozšírenie tejto práce by mohlo zahŕňať implementáciu ďalších príkazov, poskytovaných knižnicou LDTP, alebo vytvorenie nových generovacích kritérií.

POĎAKOVANIE

Tento príspevek vznikol za podpory grantu FIT-S-12-1 a výskumného záměru MSM 0021630528.

REFERENCE

- [1] Xie, Q.: Developing Cost-Effective Model-Based Techniques for GUI Testing. In Proceeding ICSE '06: Proceedings of the 28th international conference on Software engineering, New York: ACM, 2006, ISBN 1-59593-375-1, s. 997 – 1000.
- [2] Alagappan, N.: About - Linux Desktop Testing Project. [online], posledná modifikácia: 2. decembra 2012. [cit. 2013-02-27]. URL <http://ldtp.freedesktop.org/wiki/About>
- [3] Alagappan, N.: Linux Desktop Testing Project - LDTP. [online], posledná modifikácia: 2. decembra 2012. [cit. 2013-02-27]. URL <http://download.freedesktop.org/ldtp/doc/ldtp-tutorial.pdf>
- [4] Memon, A.: GUI Testing: Pitfalls and Process. IEEE Computer, 08 2002.
- [5] Sikuli Doc Team: Sikuli Script. [online], [cit. 2013-02-27]. URL <http://www.sikuli.org/>