

Szoftverellenőrzési technikák

Felhasználói felületek tesztelése

Majzik István

Budapesti Műszaki és Gazdaságtudományi Egyetem

Méréstechnika és Információs Rendszerek Tanszék

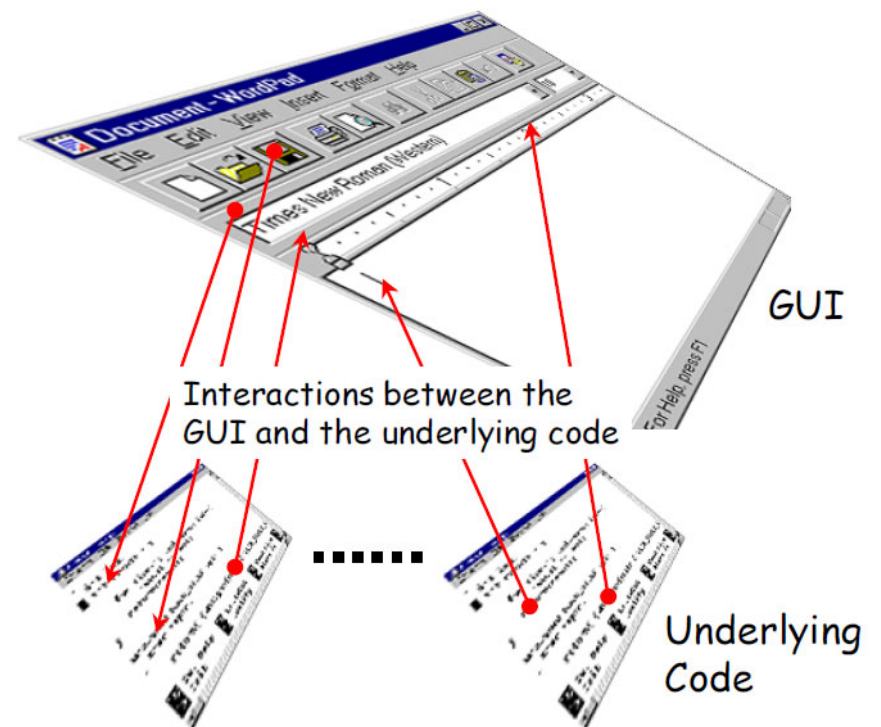
<http://www.inf.mit.bme.hu/>

Tartalomjegyzék

- Motiváció
 - Kihívások a GUI tesztelés során
 - Teszt típusok
- Operátor alapú tesztelés
 - GUI modell
 - Tesztelés scenariók alapján
- Állapotgép alapú tesztelés
 - GUI modell
 - Tesztgenerálás fedettségi kritériumok alapján
- Teszt automatizálás
 - Példák

GUI jellegzetességek

- Felhasználói utasítások fogadása, eredmény megjelenítése
 - Kommunikáció grafikus elemeken keresztül
 - A háttér alkalmazással való interakció
- Eseményvezérelt működésű
 - Manipuláció főként egérrel
- Implementációk
 - GUI toolkitok (Qt, GTK+, Swing, WinForms, ...)
 - Webes GUI



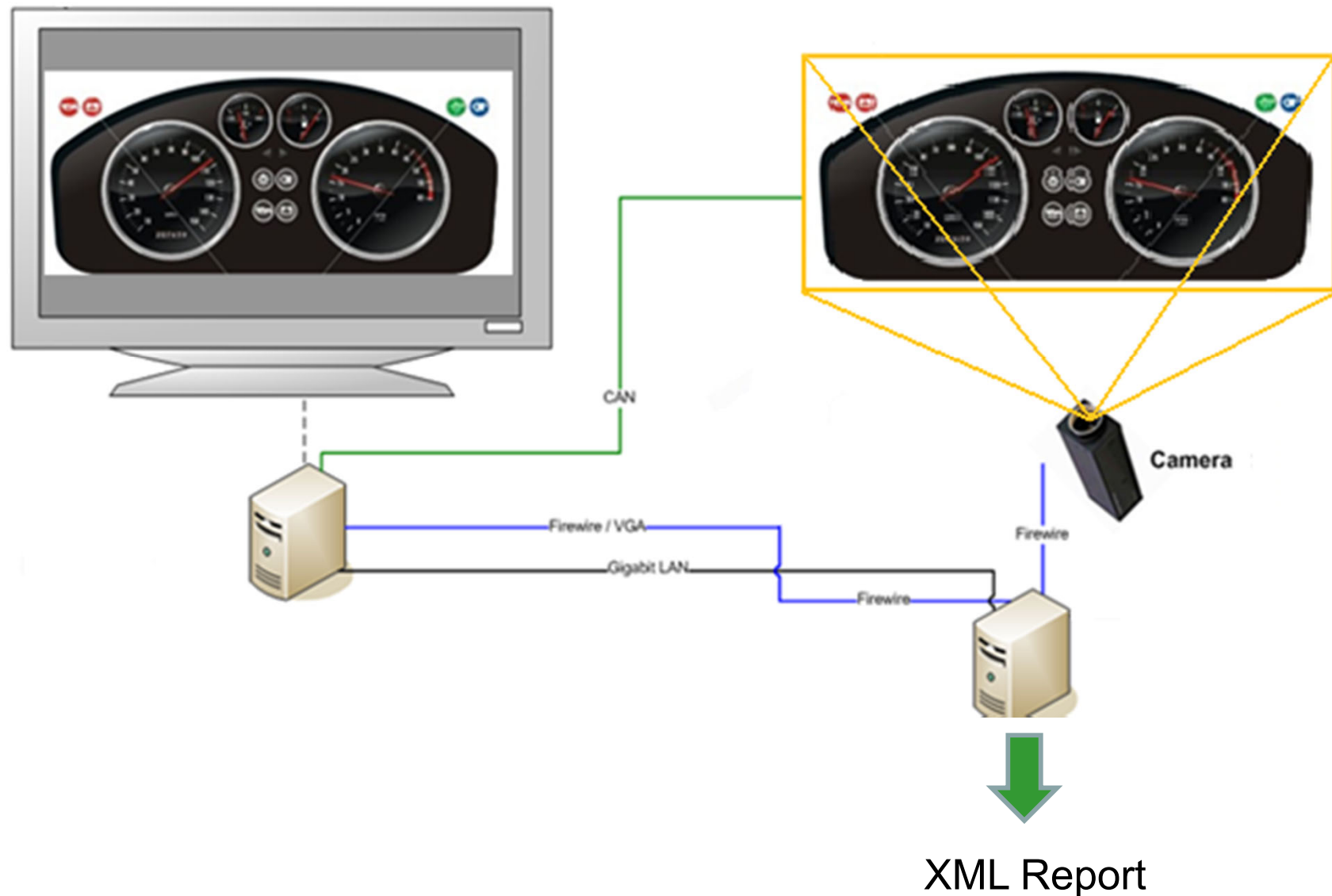
Tesztelési nehézségek

- Informális követelmények
 - Minták, konvenciók (megszokás)
 - Ergonómiai ajánlások
- Nagyszámú bemenet, nagy állapottér
 - Ugyanabban a kontextusban sokféle esemény
 - Ugyanaz az esemény sokféle kontextusban
 - Váratlan események
- Bonyolult GUI funkciók (toolkit mint „fekete doboz”)
 - Rejtett, nem dokumentált funkciók (mi várható el)
- „Interaktív” teszt végrehajtás
 - Felhasználói közreműködés (pl. egérekattintások) reprodukálása
- Nehéz kiértékelés
 - Grafikus felület változásai és háttér működés azonosítása

Példa: Teszt környezet felhasználói felülethez

Prototípus mint specifikáció (PC)

Műszerfal implementáció (beágyazott)



Teszt típusok

- **Használhatósági (ellenőrző lista)**
 - Jól „néz ki” az ablak? Könnyen hozzáférhetőek az információk?
- **Állapot alapú (belső állapotokra)**
 - Ha ezt a gombot megnyomom, akkor elérhetővé válik / tiltott lesz a másik funkció?
- **Integrációs (több modulra)**
 - Ha ezt a gombot megnyomom, megnyílik a link a böngészőben?
- **Kommunikációs (modulok között)**
 - Ha ezt a gombot megnyomom, megtörténik az interakció?
- **Szinkronizációs**
 - Ha ezt a nézetet megváltoztatom, megváltozik a másik is?
- **Terheléses**
 - Milyen gyakran kattinthatok rá?
- **Kompatibilitást vizsgáló**
 - Más alkalmazásokat nem befolyásol?
 - Más platformon is ilyen jól működik?

Szisztematikus teszt tervezés előfeltételei

- GUI modell felvétele

- Előnyök:

- Teszt fedettség definiálható
 - Automatikus tesztgenerálásra is lehetőséget ad

- Két tipikus modell:

- Operátor alapú GUI modell
 - Állapotgép alapú GUI modell

- Teszt módszer rögzítése

- Előnyök:

- GUI modellhez való illeszkedés biztosítható
 - Ad-hoc megoldásoknál jobban kézbe tartható

- Két tipikus módszer:

- Scenario alapú tesztelés: Leggyakoribb használat tesztelése
 - Kombinatorikus tesztelés: Teljes fedésre ad lehetőséget

Tartalomjegyzék

- Motiváció
 - Kihívások a GUI tesztelés során
 - Teszt típusok
- Operátor alapú tesztelés
 - GUI modell
 - Tesztelés scenariók alapján
- Állapotgép alapú tesztelés
 - GUI modell
 - Tesztgenerálás fedettségi kritériumok alapján
- Teszt automatizálás
 - Példák

Operátor alapú GUI modell

- Program objektumok
 - Háttér alkalmazás elemeihez kötött (pl. szövegrészek, fájlok, ...)
- Események
 - Menü események (MM)
 - Műveletek kibontása (pl. File/Save)
 - Fókusz kiterjesztő/kisajátító események (FKE)
 - Munkaablakok (pl. eszköztárak) megjelenítése, új ablak nyitása
 - Rendszerkapcsolati események (RKE)
 - Program objektumok megváltoztatása
- Operátorok
 - Rendszerkapcsolati operátorok: (MM,FKE)*RKE
 - Program objektumok befolyásolása (pl. Edit/Cut és hatása)
 - Felületi menü/fókusz operátorok: MM, FKE kombinálása
 - Új ablak nyitása egy művelet hatására (pl. File/Open... esetén)
 - Összetett (absztrakt) operátorok: Elemi operátorok sorozata
 - Pl. fájl kiírása adott könyvtárba

Scenario alapú tesztelés

1. Teszt cél meghatározás

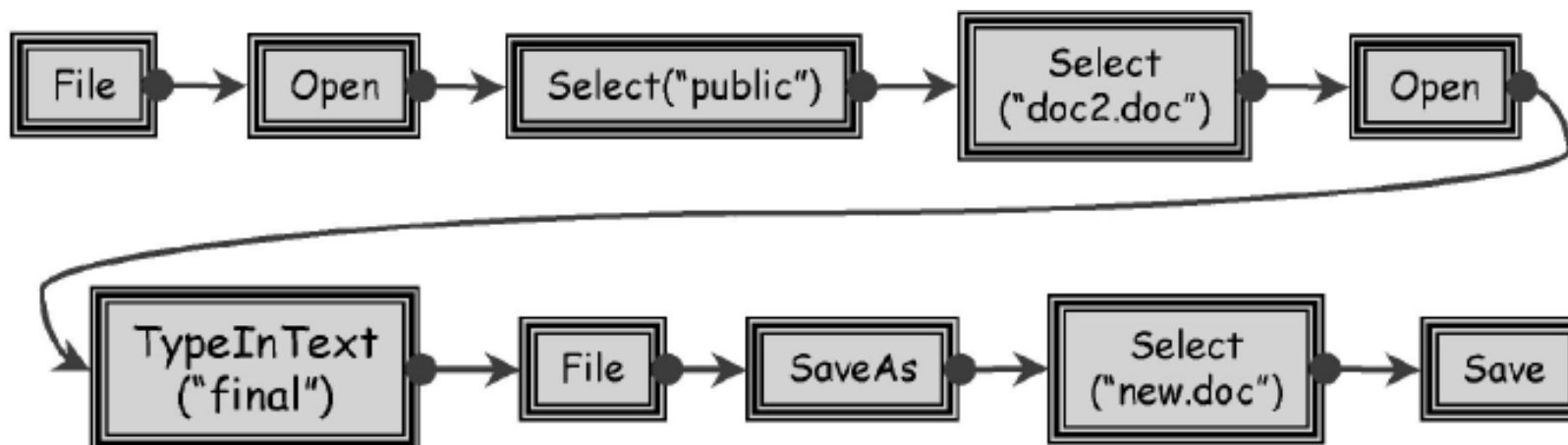
- Operátorok felmérése
- Objektumok felmérése
- Jellegzetes használat (kiindulási állapot, célállapot) meghatározása

2. Operátor szekvencia konstruálása

- Jellegzetes (legvalószínűbb) operátorsorozatok lefedése

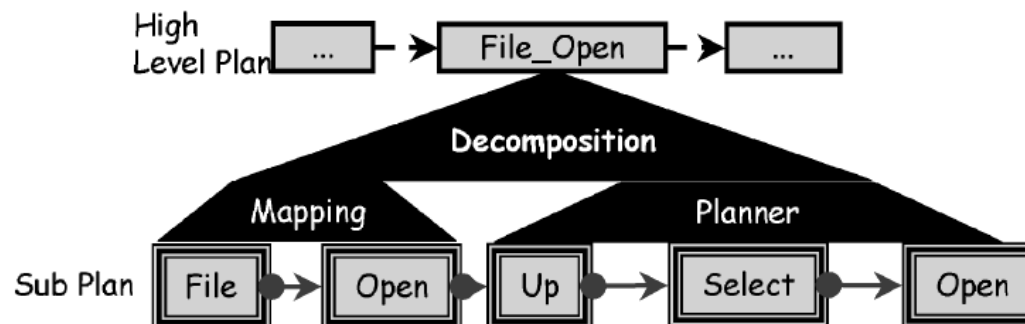
3. Konkrét esemény szekvenciára való leképezés

- Tesztesetek generálása



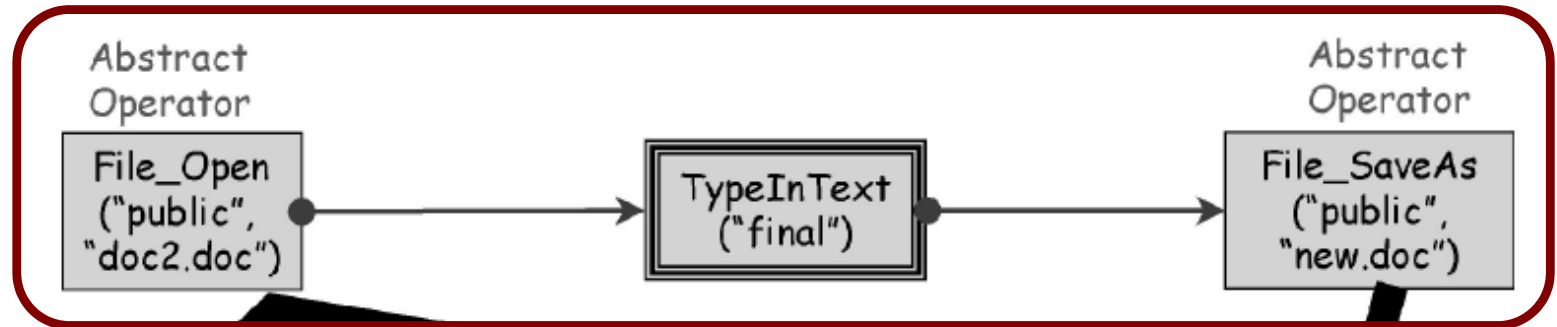
Scenario generálás tervkészítő segítségével

- A tervkészítés probléma elemei a GUI teszt generáláshoz:
 - Kezdőállapot: Kiindulási GUI és rendszerállapot (objektumok állapota)
 - Célállapot: Elérendő GUI és rendszerállapot
 - Operátorok (feltételek és hatások): GUI események alapján
 - Szabad változókat tartalmazhatnak, hierarchikusak lehetnek
 - Objektumok (lehetnek az operátorok változói): Rendszerállapot
- Megoldás: Terv (plan): Célállapot elérése a kezdőállapotból
 - Operátor példányok halmaza
 - Részleges rendezési reláció az operátorok között: sorrendi kötöttség
 - Ok-okozati kapcsolatok az operátorok között: hatások és feltételek kötése
 - Operátorok változóinak behelyettesítése: konkrét objektumok
- A terv teljes sorrendezéssel teszt szekvenciaként használható
 - Linearizálás

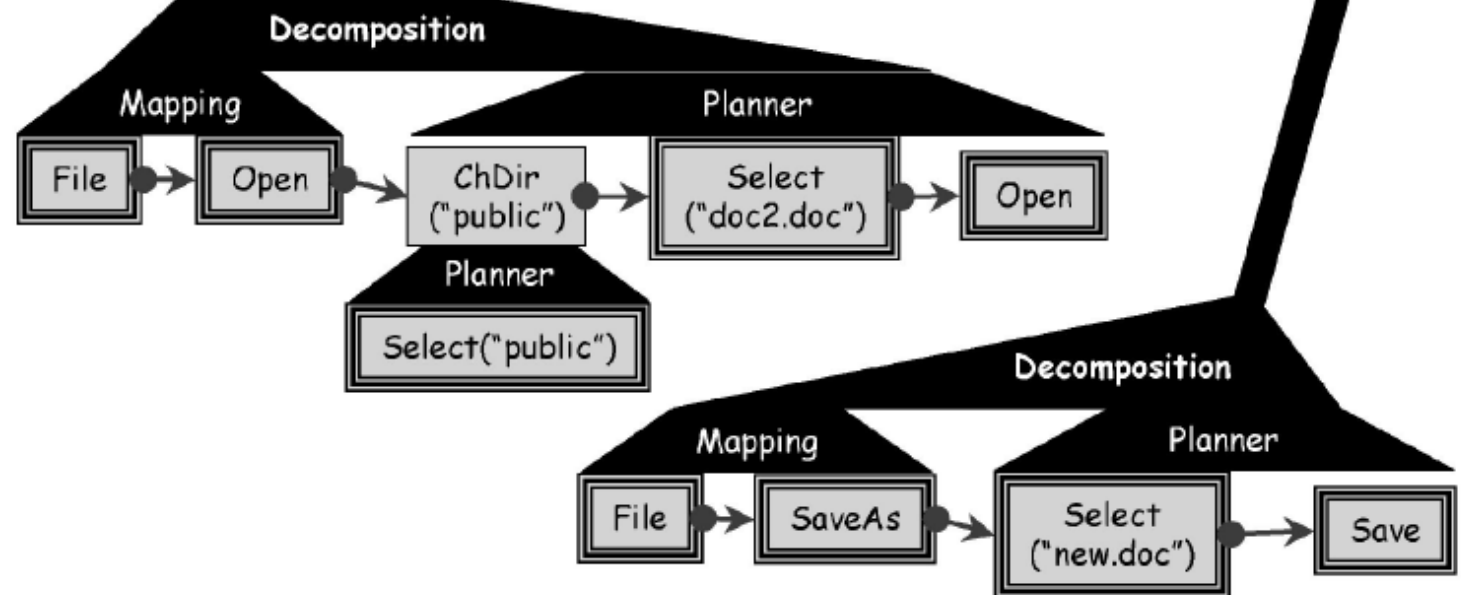


Példa egy összetett operátor leképezésére

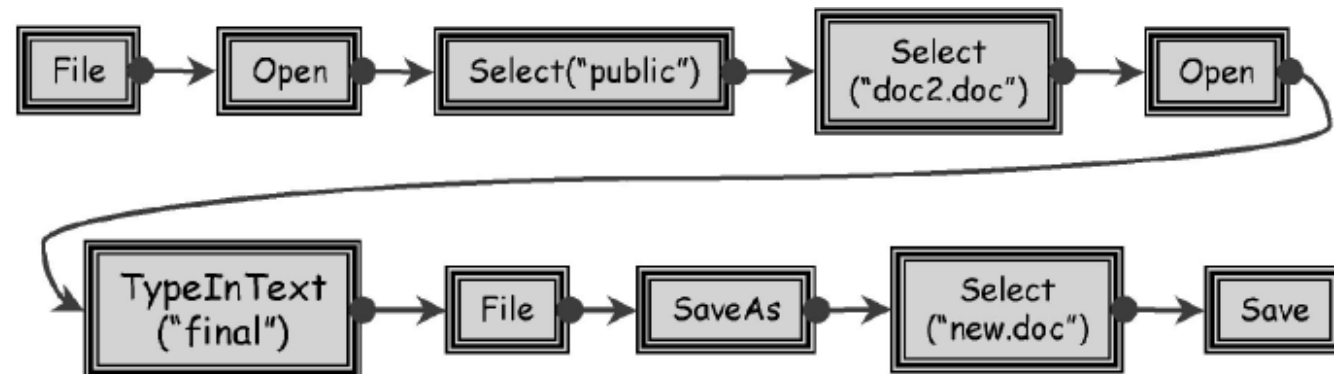
Kiindulás:



Leképezés:

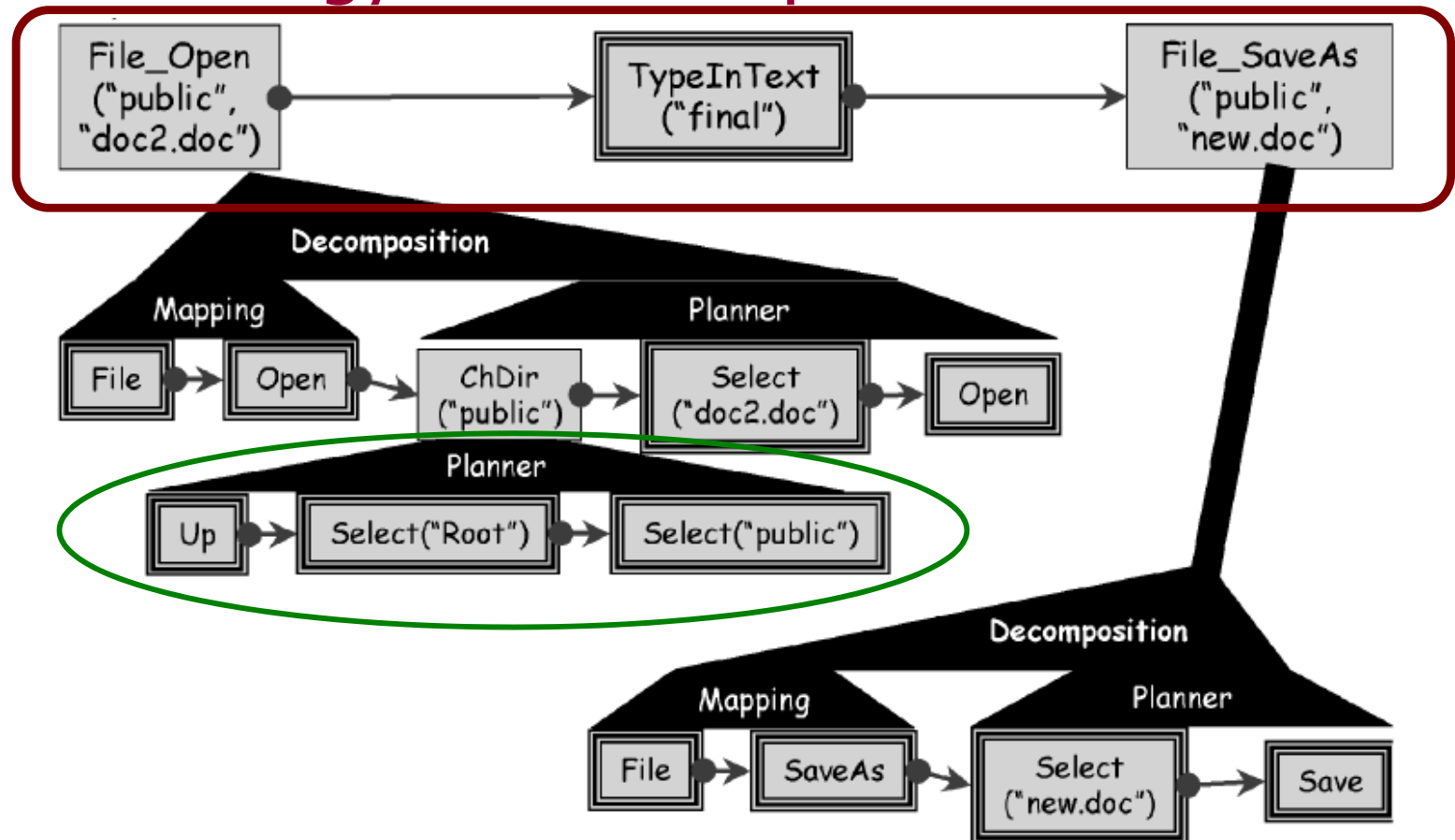


Eredmény:

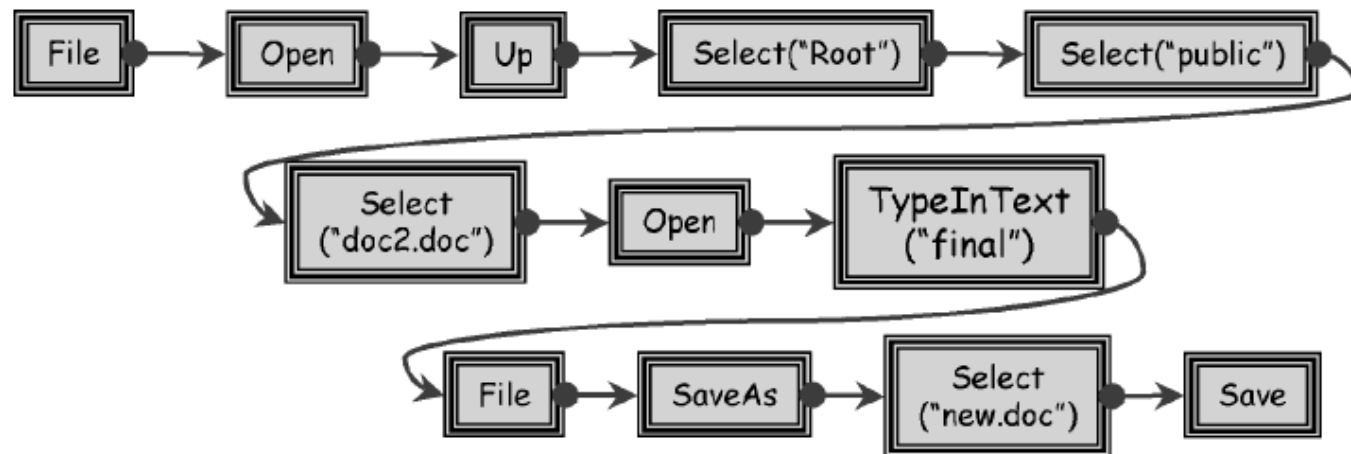


Példa egy másik leképzésre

Leképzés:



Eredmény:

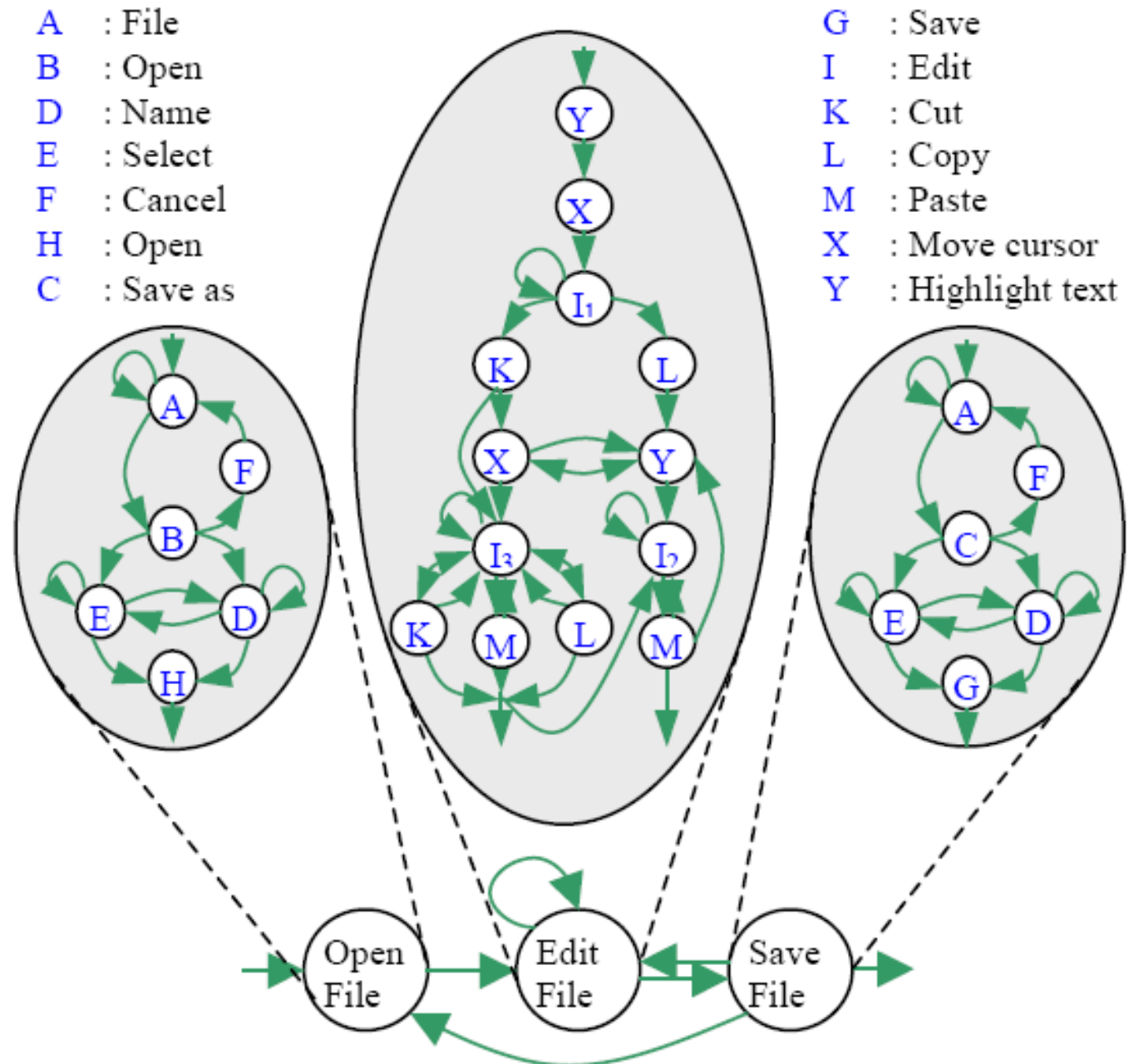


Tartalomjegyzék

- Motiváció
 - Kihívások a GUI tesztelés során
 - Teszt típusok
- Operátor alapú tesztelés
 - GUI modell
 - Tesztelés scenariók alapján
- **Állapotgép alapú tesztelés**
 - GUI modell
 - Tesztgenerálás fedettségi kritériumok alapján
- Teszt automatizálás
 - Példák

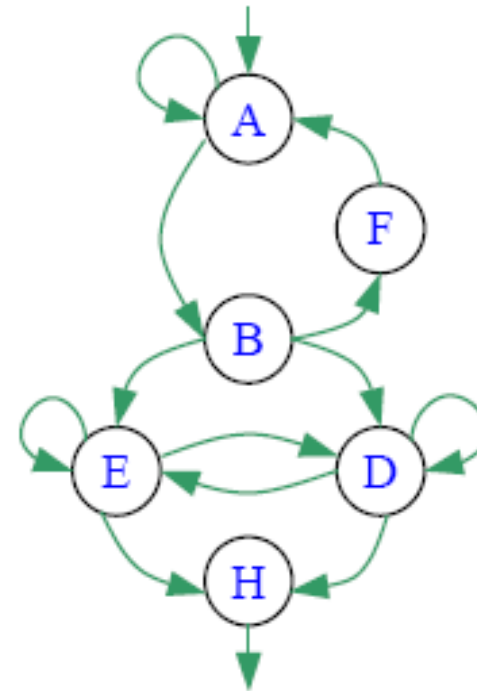
Állapotgép alapú GUI modell

- GUI mint automata
- Eseményfolyam (elemi műveletek)



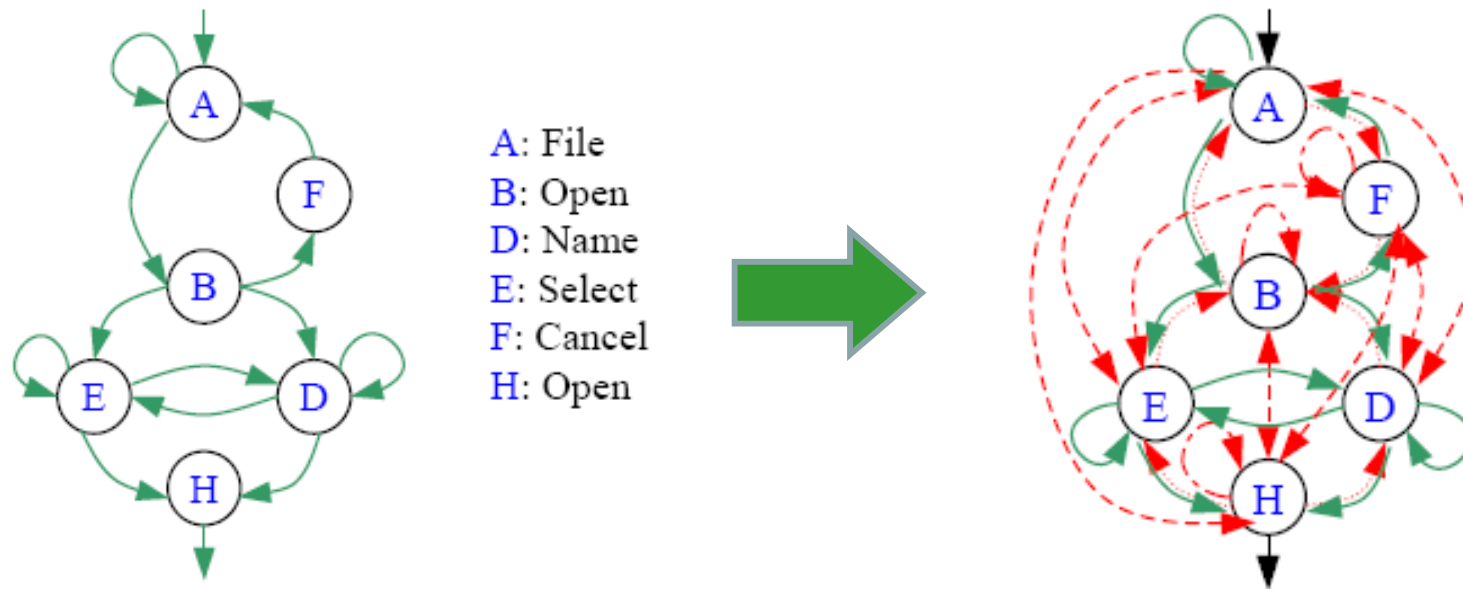
A normál működés tesztelése

- Tesztelés fedettségi kritériumok alapján:
 - Átmenetek tesztelése
 - Átmenet párok tesztelése
 - Átmenet sorozatok tesztelése
 - Részleges bejárás
 - Teljes bejárás
- Valószínűségi tesztelés
 - Legvalószínűbb bejárásokat előre kell venni
 - Markov modell is használható



A: File
B: Open
D: Name
E: Select
F: Cancel
H: Open

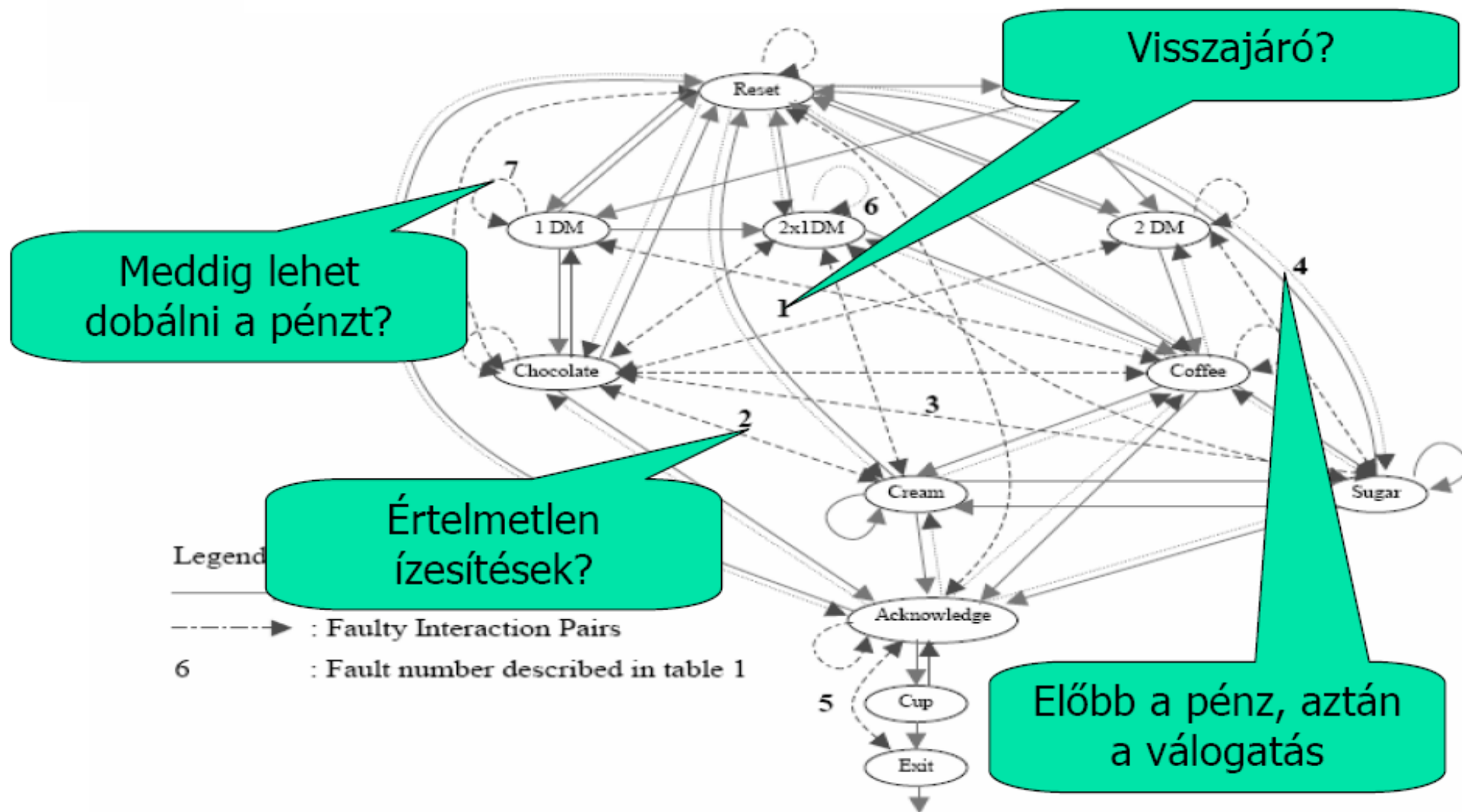
A nem megengedett működés tesztelése



- Az állapotgép kiterjesztése „tiltott” átmenetekkel:
 - Sorrend megfordítása
 - Plusz önhurkok felvétele
 - Új sorrendi kapcsolatok felvétele (teljessé tétel)
- „Tiltott” átmenetek tesztelése
 1. Normál átmenetekkel a tesztelendő tiltott átmenetig
 2. Tiltott átmenet végrehajtásának kísérlete
 3. Elvárt hibajelzés vagy a nem lehetséges végrehajtás ellenőrzése

Állapotgép példa: Kávéautomata

Felderíthető gyakorlati problémák:



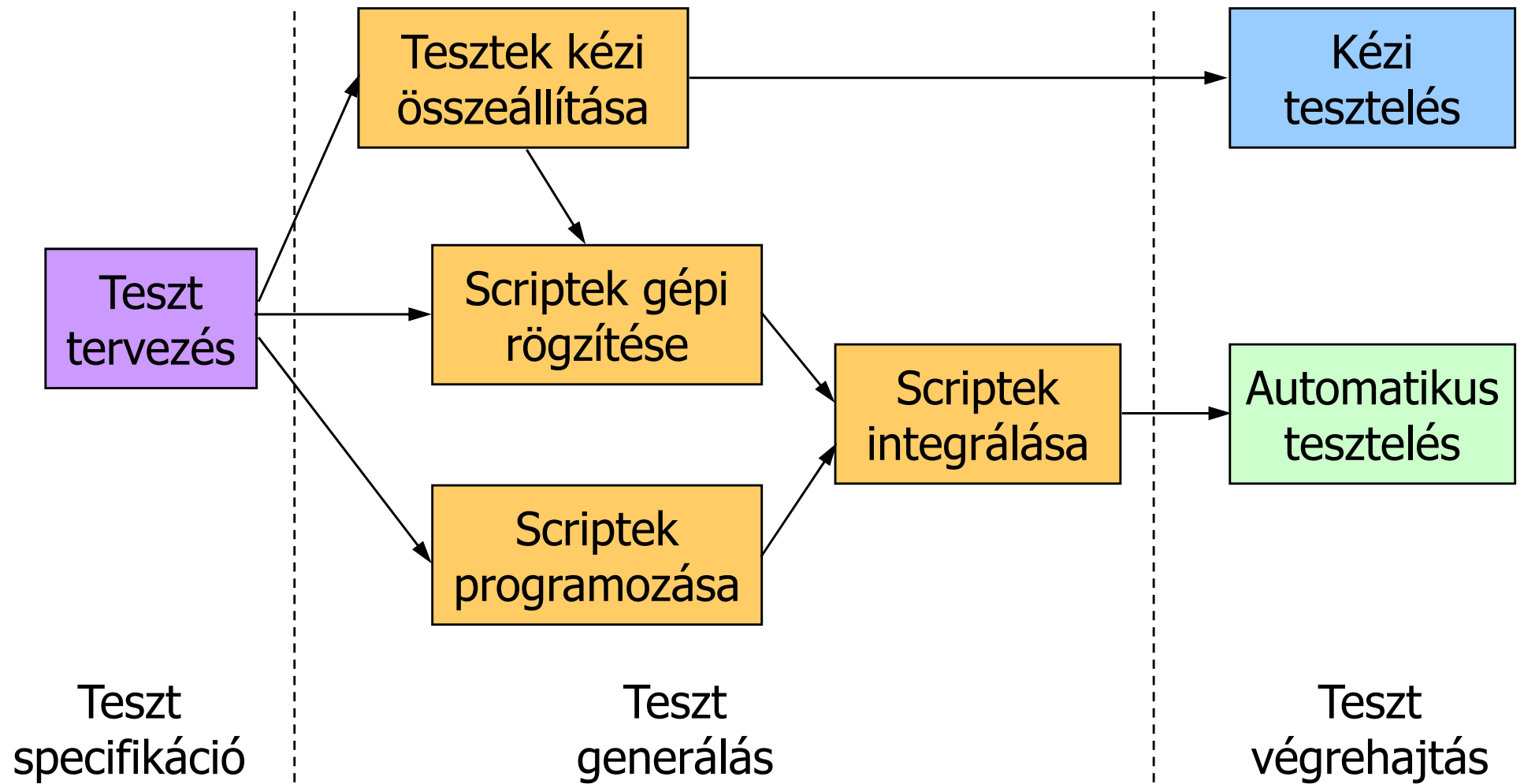
Tartalomjegyzék

- Motiváció
 - Kihívások a GUI tesztelés során
 - Teszt típusok
- Operátor alapú tesztelés
 - GUI modell
 - Tesztelés scenariók alapján
- Állapotgép alapú tesztelés
 - GUI modell
 - Tesztgenerálás fedettségi kritériumok alapján
- **Teszt automatizálás**
 - Példák

Milyen teszt típusok automatizálhatók?

- **Használhatósági** (nehezen automatizálható)
 - Jól „néz ki” az ablak?
- **Állapot alapú** (automatizálható)
 - Ha ezt megnyomom, akkor elérhetővé válik / tiltott lesz a másik funkció?
- **Integrációs** (az egyszerűbb esetekre automatizálható)
 - Ha ezt a gombot megnyomom, megnyílik a link a böngészőben?
- **Kommunikációs** (az egyszerűbb esetekre automatizálható)
 - Ha ezt a gombot megnyomom, megtörténik az interakció?
- **Szinkronizációs** (nehezen automatizálható)
 - Ha ezt a nézetet megváltoztatom, megváltozik a másik is?
- **Terheléses** (automatizálható)
 - Milyen gyakran kattinthatok rá?
- **Kompatibilitást vizsgáló** (az egyszerűbb esetekre automatizálható)
 - Más alkalmazásokat nem befolyásol?
 - Más platformon is ilyen jól működik?

Kézi és automatikus tesztelés



Automatizálási lehetőségek

- **Teszt generálás: Script felvétel („record“)**
 - Felhasználói interakciók felvétele
 - Felvett script módosítása, többszörözése
- **Teszt végrehajtás: Script lejátszás („playback“)**
 - Párhuzamosítható
 - Regressziós teszteléshez alkalmazható
- **Eredmény ellenőrzés:**
 - „Szöveg összevetés”: Csak karakteres felületekhez
 - „Képfeldolgozás”: Grafikus felületekhez
 - Widget alapú
 - Bitmap alapú

Teszt automaták (példák)

Eszköz	Környezet	Licenz
Abbot	Java	CPL
Squish	Java + Web	Commercial (Eval)
SilkTest (Borland)	Multi	Commercial (Eval)
IBM RFT	Multi	Commercial (Eval)
BadBoy	Web	FFNP
NUnit Forms	.Net	BSD
QuickTest (HP)	Java + Windows	Commercial (Eval)
Ranorex	.Net+Web	Mixed
GUIDancer	Java	Commercial (Demo)
GTT	Java	GPL
Jemmy	Java	SPL
JFCUnit	Java	LGPL
Marathon	Java	LGPL
UISpec4J	Java	CPL
QF-Test	Java	Commercial (Eval)
Selenium	Web	Apache 2.0
WET	Web	BSD
Sahi	Web	Apache 2.0

Példa: Rational Robot



- IBM Rational Functional Tester környezet
- Automatizált feladatok GUI komponensekhez:
 - Teszt szekvencia rögzítése („record”)
 - Teszt értékeléshez referencia (verifikációs pont) kijelölése
 - Menü, ablak, régió, clipboard, fájl, szöveg szintű elemek
 - Image mask megadható kép összehasonlításhoz
 - Teszt script mentése (módosítható SQABasic scriptek)
- Kiindulási információ:
 - (Grafikus) felhasználói felület felderítése, objektumok azonosítása
 - Object mapping: Felhasználói objektumokhoz
- Adatkészlet (data pool) megadható teszt sorozatokhoz
- Felhasználás:
 - Rögzített teszt szekvenciák lejátszása
 - Módosított szekvenciák lejátszása
 - Regressziós tesztelés

Példa: Selenium



- Selenium IDE: Böngészőn keresztül történő tesztelés webes felületű alkalmazásokhoz
 - Rögzíti a felhasználói interakciókat
 - Módosítás: Szerkesztés, töréspontok
 - Mentés: Ruby, JavaScript, HTML
 - Kód generálás (Java JUnit)
 - Ezek teszteléshez újra lejátszhatók
 - Teszt bemenet: URL megnyitás, kattintás, szövegbevitel, ...
 - Teszt kimenet (assertion): Widget eltűnés, megjelenés, szöveg megjelenés,...
- Selenium Remote Control:
 - A tesztek több böngészőben futtathatók
 - Szerver komponens: Böngészők indítása, HTTP proxy funkció
 - Kliens könyvtár tesztek írásához: Java, PHP, Perl, Python, Ruby nyelvekhez
- Selenium Grid:
 - A tesztek több szerveren futtathatók a párhuzamos tesztelés érdekében
 - Selenium Hub: Több Remote Controlhoz

The screenshot shows the Selenium IDE 1.0.7 interface. The title bar reads "loginBookstore - Selenium IDE 1.0.7 *". The menu bar includes "Fájl", "Szerkesztés", "Options", and "Súgó". The "Base URL" field is set to "http://localhost:8080/". Below the menu bar is a playback control bar with "Fast" and "Slow" buttons, a play button, a stop button, a refresh button, and a red stop button. The main area is divided into "Table" and "Source" tabs. The "Table" tab displays a table of test steps:

Command	Target	Value
open	/bookstore/Login.jsp	
type	userName	user
type	password	user1
clickAndWait	//input[@value='Login']	
assertTextPresent	Welcome, user!	

Below the table, there is a "Command" dropdown menu set to "assertTextPresent", a "Target" text field containing "Welcome, user!", and a "Value" text field. A "Find" button is located to the right of the "Target" field. At the bottom of the interface is a "Log" panel with tabs for "Log", "Reference", "UI-Element", "Rollup", and "Info". The "Log" tab is active, showing a list of log entries:

```
[info] Executing: |type | userName | user |
[info] Executing: |type | password | user1 |
[info] Executing: |clickAndWait |
//input[@value='Login'] | |
[info] Executing: |assertTextPresent | Welcome, user! |
|
[error] false
```


Összefoglalás

- Motiváció
 - Kihívások a GUI tesztelés során
 - Teszt típusok
- Operátor alapú tesztelés
 - GUI modell
 - Tesztelés scenariók alapján
- Állapotgép alapú tesztelés
 - GUI modell
 - Tesztgenerálás fedettségi kritériumok alapján
- Teszt automatizálás
 - Példák