This article has been accepted for publication in a future issue of a journal/proceedings, but has not been fully edited. Content may change prior to final publication. Citation information: Belli, F., Budnik, Ch. J., Towards Optimization of the Coverage Testing of Interactive Systems, Proc. 28th IEEE International Computer Software and Applications Conference (COMPSAC), pp. 18-19

Towards Optimization of the Coverage Testing of Interactive Systems

Fevzi Belli, Christof J. Budnik
Dept. of Computer Science, Electrical Engineering and Mathematics,
University of Paderborn, Germany
{belli, budnik}@adt.upb.de

1. Introduction

While developing interactive systems, the construction of the *user interactions* (*UI*) has several *desirable* properties, as to user friendliness, reliability or safety. Vulnerabilities of any such property can lead to a *undesirable* situation. The undesirable system features are the sum of the situations, which are complementary to the desirable ones [2].

This paper introduces a model-based approach for minimization of test costs for interactive systems. Results known from state-based conformance testing and graph theory are used and extended to construct algorithms for test case generation and selection to cover the behavioral model of the *system under test* (*SUT*).

The test case selection on the basis of the established model is ruled by an *adequacy criterion,* which provides a measure of how effective a given set of test cases is in terms of its potential to reveal faults [3].

The approach is specification-oriented; i.e., the underlying model represents the system behavior interacting with the user's actions, that are viewed here as *events*.

2. Fault Model and Test Process

The system behavior and, moreover, the facilities from the user's point of view to interact with the system are represented by an *Event Sequence Graphs (ESG).*

Definition 1. An *Event Sequence Graph* ESG=(V, E) is a directed graph with a finite set of *nodes* (*vertices*) $V \neq \emptyset$ and a finite set of *arcs* (*edges*) $E \subseteq V \times V$.

For representing user-system interactions, the nodes of the ESG are interpreted as events. The operations on identifiable components of the UI are controlled/perceived by input/output devices, i.e., elements of windows, buttons, lists, checkboxes, etc. (Fig. 1).

Definition 2. Let V, E be defined as in Def. 1. Then any sequence of nodes $\langle v_0,\ldots,v_k\rangle$ is called an *event sequence (ES)* if $(v_i, v_{i+1}) \in E$, for $i = 0,\ldots,k-1$.

Furthermore, α (*initial*) and ω (*end*) are functions to determine the initial node and end node of an ES, i.e., $\alpha(ES)=v_0$, $\omega(ES)=v_k$. In particular, ES=$\langle v_i\rangle$ is an ES of length 1. An ES=$\langle v_i, v_k\rangle$ of length 2 is called an *event pair (EP).*

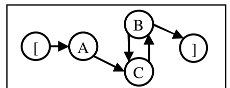

Fig. 1: An ESG with [as entry and] as exit

The assumption is made that there is an ES from the single node *s* to all other nodes, and from all nodes there is an ES to the single node γ (s, $\gamma \notin V$). *s* is called the *entry* and γ is called the *exit* of the ESG.

Definition 3. An ES is called a *complete ES* (*Complete Event Sequence*, *CES*), if $\alpha(ES)=s$ is the entry and $\omega(ES)=\gamma$ is the exit.

A CES represents a *walk* through the ESG.

Definition 4. The *complementary ESG* is defined as $\overline{ESG} = (V, \overline{E})$ with $\overline{E} = \widehat{E} \setminus E$ (\: set difference operation) and $\widehat{E} = V \times V$.

Any EP of the $\overline{ESG}$ is then a *faulty event pair* (*FEP*) for ESG (Fig. 2).

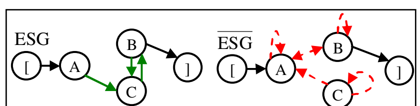

Fig. 2: ESG of Fig. 1 and its inversion $\overline{ESG}$

Definition 5. Let ES=$\langle v_0,\ldots,v_k\rangle$ be an event sequence of length k+1 of an ESG and FEP=$\langle v_k, v_m\rangle$ a faulty event pair of the according $\overline{ESG}$. The concatenation of the ES and FEP forms then a *faulty event sequence* *FES* = $\langle v_0,\ldots,v_k, v_m\rangle$.

Definition 6. An *FES* will be called *complete* (*Faulty Complete Event Sequence*, *FCES*) if $\alpha(ES)=s$ is the entry. The ES as part of a FCES is called a *starter*.

The test process introduced in this paper applies CES, and FCES, as test inputs. If the input is a CES, the SUT is supposed to proceed it and thus, to *succeed* the test. Accordingly, if a FCES is used as a test input, a *failure* is expected to occur.

The problem to determine the point in time in which to stop testing (*test termination*) is converted into the *coverage of the ES and FES of length k of the* ESG subject to keeping the total length minimal.

Depending on the length of ESs and FESs, the test costs can be scalable and stepwise increased in accordance with the quality goal and test budget.

3. Minimizing the Spanning Set

The union of the sets [4] of CESs of minimal total length to cover the ESs of a required length is called *Minimal Spanning Set of Complete Event Sequences (MSCES)*. A similar problem to the determination of MSCESs is the *Directed Chinese Postman Problem (DCPP)* [1].

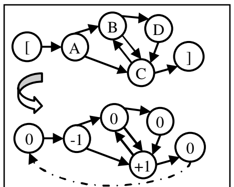

Fig. 3: Transferring walks into tours

The object is to determine a set of walks with the minimal total length to cover all EPs and requires that this graph be *strongly connected*. This can be done through an additional arc from the exit to the entry (Fig. 3). The figures within the nodes indicate the calculated difference of incoming and outgoing degree of these nodes. These balance values determine the number of additional EPs that will be identified by searching the *all-shortest-path* and solving the optimization problem [5]. The problem can then be transferred to the construction of the Euler tour for this graph. The MSCES can be solved in O(|V|^3) time.

The union of the sets of FCESs of the minimal total length to cover the FESs of a required length is called *Minimal Spanning Set of Faulty Complete Event Sequences (MSFCES)*. In comparison to the interpretation of the CESs as legal walks, *illegal walks* are realized by FCESs that never reach the exit. An illegal walk is *minimal* if its starter cannot be shortened.

A phenomenon in testing interactive systems is that faults can be frequently detected and reproduced only in some context. This makes test sequences of length > 2 necessary since repetitive occurrences of some subsequences are needed to cause an error to occur/re-occur.

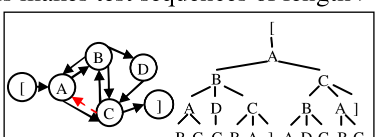

Fig. 4: Considering intrinsic faults and the reachability tree (not complete)

Consider following scenario, based on the ESG given in Fig. 4: The tester observes that the EP given by *BC* always reveals a fault, no matter if executed within *[ABC]*, or *[ABABC]*; i.e., the test cases containing *BC* always detect the fault in any context. The same scenario (so the assumption) demonstrates that the FEP *CA* reveals another fault, however only in the context of *[ABCBCA*, and never within *[ABCA*, or *[ABDCA*, etc. Such observations clearly indicate that the test process must be applied to longer ESs than 2 (EPs).

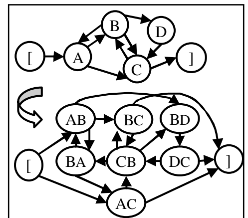

Fig. 5: Extending the ESG for covering ETr

To solve this problem, the given ESG is transformed into a graph in which the nodes are used to generate test cases of length > 2, in the same way that the nodes of the original ESG are used to generate EPs and to determine the appropriate MSCES. For this purpose, the reachability tree (Fig. 4) of the nodes is traversed to determine the sequences of adjacent nodes of length *n-1*, if any. Following this step, a node *u* is connected with a node *v* if the last *n-1* events that are used in the identifier of *u* are the same as the first *n-1* events that are included in the identifier of *v*. The nodes *[*, *]* are connected with all the extensions of the nodes with which they were connected before the extension. Fig. 5 illustrates the generation of ESs of length=3.

4. Conclusion and Future Work

The objective of the approach is the construction of a set of CESs of minimal total length that covers all ESs of a required length. A similar optimization problem arises for the validation of the SUT under exceptional, undesirable situations. To model the latter problem, FESs and FCESs are considered. These optimizing problems have been called determination of Minimal Spanning Sets of CESs and FCESs, respectively. The research has shown that their complexity is expectedly less than the complexity of similar problems, since the vertices of ESGs are equidistant and its edges have no attributes and weights.

The next step planned is to exploit the structure of the SUT to enable the elimination of unnecessary and/or infeasible test cases for further reduction of the test costs.

Literature

[1] A. V. Aho, A. T. Dahbura, D. Lee, M. Ü. Uyar, "An Optimization Technique for Protocol Conformance Test Generation Based on UIO Sequences and Rural Chinese Postman Tours", *IEEE Trans. Com.* 39, pp. 1604-1615, 1991

[2] F. Belli, "Finite-State Testing and Analysis of Graphical User Interfaces", Proc. 12th ISSRE, pp. 34-43, 2001

[3] Hong Zhu, P.A.V. Hall, J.H.R. May, "Software Unit Test Coverage and Adequacy", ACM Comp. Surveys, 29/4, 1997

[4] M. Marré, A. Bertolino, "Using Spanning Sets for Coverage Testing", *IEEE Trans. on Softw. Eng.* 29/11, pp. 974-984, 2003

[5] H. Thimbleby "The directed Chinese Postman Problem", Softw., Pract. Exper. 33(11): 1081-1096, 2003