PTO/SB/16 (10-17)
Approved for use through 11/30/2020. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PROVISIONAL APPLICATION FOR PATENT COVER SHEET – Page 1 of 1

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53(c).

Priority Mail Express® Label No. _____

INVENTOR(S)

Given Name (first and middle [if any])	Family Name or Surname	Residence (City and either State or Foreign Country)
Michael	SHALYT	Haifa, Israel
Nataly	BENDERSKY	Tel Aviv, Israel

Additional inventors are being named on the _____ separately numbered sheets attached hereto.

TITLE OF THE INVENTION (500 characters max):

SYSTEM AND METHOD FOR AUTOMATED TEST GENERATION AND TESTING AN APPLICATIONS

CORRESPONDENCE ADDRESS

Direct all correspondence to:

☒ *The address corresponding to* Customer Number: **49443**

OR

☒ Firm *or* Individual Name **Pearl Cohen Zedek Latzer Baratz LLP**

Address	**1500 Broadway, 12th Floor**				
City	**New York**	State	**New York**	Zip	**10036**
Country	**USA**	Telephone	**(646) 878-0800**	Email	**USPTO@PearlCohen.com**

ENCLOSED APPLICATION PARTS *(check all that apply)*

☒ Application Data Sheet. See 37 CFR 1.76 ☐ CD(s), Number of CDs _____

☒ Drawing(s) *Number of Sheets* 4 ☐ Other (specify) _____

☒ Specification (e.g. description of the invention) *Number of Pages* 20

Fees Due: Filing Fee of $280 ($140 for small entity). If the specification and drawings exceed 100 sheets of paper, an application size fee is also due, which is $400 ($200 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).

METHOD OF PAYMENT OF THE FILING FEE AND APPLICATION SIZE FEE FOR THIS PROVISIONAL APPLICATION FOR PATENT

☒ Applicant asserts small entity status. See 37 CFR 1.27.

☐ A check or money order made payable to the *Director of the United States Patent and Trademark Office* is enclosed to cover the filing fee and application size fee (if applicable).

☐ Payment by credit card. Form PTO-2038 is attached.

140

TOTAL FEE AMOUNT ($)

☒ To the extent that there are any fees associated with this filing that are due but are not paid or are not able to be paid by a credit card mentioned elsewhere in this document, Applicants authorize the U.S. Patent and Trademark Office to charge any such fees to Deposit Account Number: **60-2041**

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.

☒ No.

☐ Yes, the invention was made by an agency of the U.S. Government. The U.S. Government agency name is: _____

☐ Yes, the invention was made under a contract with an agency of the U.S. Government. The name of the U.S. Government agency and Government contract number are: _____

WARNING:

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to the USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.

SIGNATURE /Devorah Livneh, Reg. No. 71,347/ Date **18 March 2019**

TYPED or PRINTED NAME **Devorah N. Livneh** REGISTRATION NO. (if appropriate) **71,347**

TELEPHONE **646-878-0800** Docket Number: **P-582424-USP**

USE ONLY FOR FILING A PROVISIONAL APPLICATION FOR PATENT

This collection of information is required by 37 CFR 1.51. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 8 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

SYSTEM AND METHOD FOR AUTOMATED TEST GENERATION AND TESTING AN APPLICATIONS

FIELD OF THE INVENTION

[0001] The present invention relates generally to testing applications. More specifically, the present invention relates to automatically and autonomously defining and performing test procedures for computer applications.

BACKGROUND OF THE INVENTION

[0002] Known systems and methods for testing computer applications are costly and typically involve a human tester who is required to repeat operations and/or interactions with a tested application. Accordingly, not only are known systems and methods costly both time-wise and budget-wise, since a human tester is error-prone, the results cannot be trusted to indicate the true state or quality of a tested application. Moreover, known systems and methods do not scale with complexity and scope of applications to be tested. Accordingly, there is a need in the art for an automated system and method that can automatically and autonomously test an application.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003] Non-limiting examples of embodiments of the disclosure are described below with reference to figures attached hereto that are listed following this paragraph. Identical features that appear in more than one figure are generally labeled with a same label in all the figures in which they appear. A label labeling an icon representing a given feature of an embodiment of the disclosure in a figure may be used to reference the given feature. Dimensions of features shown in the figures are chosen for convenience and clarity of presentation and are not necessarily shown to scale. For example, the dimensions of some of the elements may be exaggerated relative to other elements for clarity, or several physical components may be included in one functional block or element. Further, where considered appropriate, reference numerals may be repeated among the figures to indicate corresponding or analogous elements.

[0004] The invention, both as to organization and method of operation, together with objects, features and advantages thereof, may best be understood by reference to the following detailed description when read with the accompanied drawings. Embodiments of the invention are illustrated by way of example and not limitation in the figures of the accompanying drawings, in

which like reference numerals indicate corresponding, analogous or similar elements, and in which:

[0005] Fig. 1 shows a block diagram of a computing device according to illustrative embodiments of the present invention;

[0006] Fig. 2 shows a system and flows according to illustrative embodiments of the present invention;

[0007] Fig. 3 graphically shows paths that include states and interactions according to illustrative embodiments of the present invention; and

[0008] Fig. 4 shows a flow according to illustrative embodiments of the present invention.

DETAILED DESCRIPTION

[0009] In the following detailed description, numerous specific details are set forth in order to provide a thorough understanding of the invention. However, it will be understood by those skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known methods, procedures, and components, modules, units and/or circuits have not been described in detail so as not to obscure the invention. Some features or elements described with respect to one embodiment may be combined with features or elements described with respect to other embodiments. For the sake of clarity, discussion of same or similar features or elements may not be repeated.

[0010] Although embodiments of the invention are not limited in this regard, discussions utilizing terms such as, for example, "processing," "computing," "calculating," "determining," "establishing", "analyzing", "checking", or the like, may refer to operation(s) and/or process(es) of a computer, a computing platform, a computing system, or other electronic computing device, that manipulates and/or transforms data represented as physical (e.g., electronic) quantities within the computer's registers and/or memories into other data similarly represented as physical quantities within the computer's registers and/or memories or other information non-transitory storage medium that may store instructions to perform operations and/or processes. Although embodiments of the invention are not limited in this regard, the terms "plurality" and "a plurality" as used herein may include, for example, "multiple" or "two or more". The terms "plurality" or "a plurality" may be used throughout the specification to describe two or more components, devices,

elements, units, parameters, or the like. The term set when used herein may include one or more items.

[0011] Unless explicitly stated, the method embodiments described herein are not constrained to a particular order in time or to a chronological sequence. Additionally, some of the described method elements can occur, or be performed, simultaneously, at the same point in time, or concurrently. Some of the described method elements may be skipped, or they may be repeated, during a sequence of operations of a method.

[0012] Reference is made to Fig. 1, showing a non-limiting, block diagram of a computing device or system 100 that may be used to autonomously and automatically define tests for an application and/or test an application according to some embodiments of the present invention. Computing device 100 may include a controller 105 that may be a hardware controller. For example, computer hardware processor or hardware controller 105 may be, or may include, a central processing unit processor (CPU), a chip or any suitable computing or computational device. Computing system 100 may include a memory 120, executable code 125, a storage system 130 and input/output (I/O) components 135. Controller 105 (or one or more controllers or processors, possibly across multiple units or devices) may be configured (e.g., by executing software or code) to carry out methods described herein, and/or to execute or act as the various modules, units, etc., for example by executing software or by using dedicated circuitry. More than one computing devices 100 may be included in, and one or more computing devices 100 may be, or act as the components of, a system according to some embodiments of the invention.

[0013] Memory 120 may be a hardware memory. For example, memory 120 may be, or may include machine-readable media for storing software e.g., a Random-Access Memory (RAM), a read only memory (ROM), a memory chip, a Flash memory, a volatile and/or non-volatile memory or other suitable memory units or storage units. Memory 120 may be or may include a plurality of, possibly different memory units. Memory 120 may be a computer or processor non-transitory readable medium, or a computer non-transitory storage medium, e.g., a RAM. Some embodiments may include a non-transitory storage medium having stored thereon instructions which when executed cause the processor to carry out methods disclosed herein.

[0014] Executable code 125 may be an application, a program, a process, task or script. A program, application or software as referred to herein may be any type of instructions, e.g., firmware,

middleware, microcode, compiled machine code, hardware description language etc. that, when executed by one or more hardware processors or controllers 105, cause a processing system or device (e.g., system 100) to perform the various functions described herein.

[0015] Executable code 125 may be executed by controller 105 possibly under control of an operating system. For example, executable code 125 may be an application that autonomously and automatically defines tests for an application and/or tests an application as further described herein. Although, for the sake of clarity, a single item of executable code 125 is shown in Fig. 1, a system according to some embodiments of the invention may include a plurality of executable code segments similar to executable code 125 that may be loaded into memory 120 and cause controller 105 to carry out methods described herein.

[0016] Storage system 130 may be or may include, for example, a hard disk drive, a universal serial bus (USB) device or other suitable removable and/or fixed storage unit. As shown, storage system 130 may include states 131 and paths 132 (collectively referred to hereinafter as states 131 and/or paths 132 or individually as state 131 and/or path 132, merely for simplicity purposes).

[0017] States 131 and paths 132 as referred to herein may be any suitable digital data structure or construct or computer data objects that enables storing, retrieving and modifying values. For example, states 131 and paths 132 may be files, tables or lists in a database in storage system 130, and may include a number of fields that can be set or cleared, a plurality of parameters for which values can be set, a plurality of entries that may be modified and so on. For example, states, screens or other attributes of a tested application may be set, cleared or modified in states 131 that may be associated with one or more paths 132.

[0018] Content or data may be loaded from storage system 130 into memory 120 where it may be processed by controller 105. For example, paths 132 and states 131 may be loaded into memory 120 and used for testing an application as further described herein.

[0019] In some embodiments, some of the components shown in Fig. 1 may be omitted. For example, memory 120 may be a non-volatile memory having the storage capacity of storage system 130. Accordingly, although shown as a separate component, storage system 130 may be embedded or included in system 100, e.g., in memory 120.

[0020] I/O components 135 may be, may be used for connecting (e.g., via included ports) or they may include: a mouse; a keyboard; a touch screen or pad or any suitable input device. I/O components may include one or more screens, touchscreens, displays or monitors, speakers and/or any other suitable output devices. Any applicable I/O components may be connected to computing device 100 as shown by I/O components 135, for example, a wired or wireless network interface card (NIC), a universal serial bus (USB) device or an external hard drive may be included in I/O components 135.

[0021] A system according to some embodiments of the invention may include components such as, but not limited to, a plurality of central processing units (CPU) or any other suitable multi-purpose or specific processors, controllers, microprocessors, microcontrollers, field programmable gate arrays (FPGAs), programmable logic devices (PLDs) or application-specific integrated circuits (ASIC). A system according to some embodiments of the invention may include a plurality of input units, a plurality of output units, a plurality of memory units, and a plurality of storage units. A system may additionally include other suitable hardware components and/or software components. In some embodiments, a system may include or may be, for example, a personal computer, a desktop computer, a laptop computer, a workstation, a server computer, a network device, or any other suitable computing device.

[0022] Reference is made to Fig. 2, showing a system and flows according to illustrative embodiments of the present invention. As shown a system may include, or may be operatively connected to, a system under test (SUT) 215. SUT 215 may be any system, device or application that is being tested for correct operation. SUT 215 may be, may include or may act as components of, computing device 100. SUT 215 may be or may include software (e.g., executable code 125) and/or firmware application that may be executed on a computer, smartphone or any suitable computing device 100. As further shown, a system may include an autotester (AT) unit 210 that may automatically and autonomously test and/or model SUT 215. For example, AT 210 may be a bot (e.g. a computer program that presents output and reacts to input as if a human would) that mimics or simulates a human, e.g., AT 210 may be a bot that interacts with SUT 215 the same way a human tester does. It will be noted that although (for the sake of clarity), AT 210 and SUT 215 are shown as separate units, various configurations may be contemplated, e.g., AT 210 and SUT 215 may be included in the same, single computing device 100 or they may reside on two separate computing devices. As shown by block 220, AT 210 may interact with SUT 215. For example, AT

210 may simulate a mouse click on a graphical user interface (GUI) button displayed by SUT 215. Generally, output or interaction 220 performed by AT 210 may be any interaction with SUT 215 that may be done by a user or by an application. For example, interaction 220 may be, or may include, input to a GUI or other interface, mouse clicks as described, a touch on a touch screen, sending an email or other message to SUT 215 and so on. Generally, interaction 220 may include any data, event or message provided to SUT 215 which simulates or mimics what occurs when a human or application interacts with SUT 215, e.g., filling of input text boxes, clicking buttons, providing credentials etc.

[0023] As shown by block 225, a response of SUT 215 to interaction 220 may be received by AT 210. For example, AT 210 may analyze the screen on which SUT 215 displays output to thus determine response 225, AT 210 may use a microphone to capture audio response or output of SUT 215 and/or AT 210 may monitor any aspect of a computing device on which SUT 215 is executed in order to identify a response of SUT 215 to an interaction 220. In some embodiments, AT 210 may be connected to, or even integrated in, SUT 215, such that any internal state change of SUT 215 may be detected and/or provided as response 225. Accordingly, any response of SUT 215 may be detected by AT 210 and processed as further described.

[0024] As shown by block 230, based on response 225, AT 210 may determine and/or evaluate the state of SUT 210. Generally, a state 131 of SUT 210 as referred to herein may include any value, parameter, attribute or aspect of SUT 215. For example, a state determined as described may include a layout of a display screen or monitor including all GUI objects and their states (e.g., dimmed, active and so on). A state 131 of SUT 215 may include a mode, e.g., SUT 215 is waiting for input, waiting for a timer, shutting down and so on.

[0025] As shown by block 245, AT 210 may record state information. For example, following an interaction 220, an output or response 225 and a determination of a state as shown by block 230, the state of SUT 215 may be recorded, e.g., in a state 131 object on storage system 130.

[0026] As shown by block 235, based on the state of SUT 215, based on path information and further based on additional considerations as described, an (new or additional) interaction with SUT 215 may be selected and the selected interaction may be performed as shown by block 220. As further shown, the selected interaction may be recorded and used, for example, in defining a path as further described herein.

[0027] As shown by block 250, using at least state information and interaction data, AT 210 may determine and record path information. For example, path information may be recorded in a path 132 object in storage system 130. Generally, path information may include data that represents both interactions and states, e.g., as further described with reference to Fig. 3.

[0028] Reference is made to Fig. 3 which graphically shows paths that include states and interactions according to illustrative embodiments of the present invention. For example, SUT 215 may be in an initial state S1, e.g., S1 may be the state of SUT 215 when first launched. Following an interaction I1 (e.g., a click on a GUI button displayed by SUT 215), SUT 215 may move to (or assume) state S2, then, following interaction I2, SUT 215 may move to state S3 and finally, following interaction I3, SUT 215 may reach state S4 that may be, for example, a desired final state. As illustrated, state S4 can be reached via the sequence of interactions and states (path) S1, I4, S5, I6 S6 and I7. Alternatively, state S4 can be reached via the path of S1, I4, S5, I5, S3 and I3. A path or state-space may include a set of one or more forbidden states. For example, states S7 and S8 shown in Fig. 3 may be states that SUT 215 must never assume or be in, e.g., during an automated test, if AT 210 detects or determines that SUT 215 is in one of states S7 or S8 then AT 210 determines the test has failed.

[0029] In some embodiments, reaching a forbidden state may be used for adjusting, modifying or updating AT 210. For example, if, AT 210, causes SUT 215 to actually transfer funds to a seller of a product, e.g., when SUT 215 is in forbidden state S7, then logic of AT 210 may be revised so that real purchases never occur when AT 210 is running a test. Accordingly, embodiments of the invention enable improving an automatic tester based on results of a test.

[0030] Paths 132 stored in storage 130 may include information that describes the paths illustrated in Fig. 3. For example, a path 132 may include states, interactions and their relations, e.g., which interaction carries SUT 215 from a first state 131 to a second state 131.

[0031] In some embodiment, AT 210 may automatically and autonomously map, identify, discover or determine some, or even all paths that carry SUT 215 from state S1 to state S4. Otherwise described, the path-space of SUT 215 may be automatically identified or discovered by AT 210. Conceptually, e.g., given enough time and processing power, AT 210 may automatically and autonomously map, identify, discover or determine the entire path-space of SUT 215, e.g., paths 132 may include information that completely describes the behavior of SUT 215, e.g., for

each of all possible states of SUT 215, paths 132 may include all states that SUT 215 can assume or move to, as well as the relevant interactions or events that cause the state transitions.

[0032] In some embodiments, AT 210 is provided with a path, e.g., a path that includes an initial state of SUT 215, a set of intermediate states and a final (goal) state and the interactions that cause SUT 215 to move through the intermediate states from the initial state to the final, goal or desired state. Using a path provided as described, AT 210 may automatically and autonomously test SUT 215, e.g., by using a path provided to generate interactions 220 and verifying (e.g., as shown by block 230) that SUT 215 indeed behaves as expected, e.g., moves along, or according to, the path.

[0033] As further described below, AT 210 can receive a path and then, for example, as SUT 215 evolves (e.g., in subsequent versions or updates), AT 210 can change a received or original path such that new paths are discovered. Accordingly, unlike scripted tests or other known methods, AT 210 may autonomously and automatically create new tests as a tested system evolves, autonomously and/or automatically finds tests for a given tested system as further described herein. In yet other cases and as described, AT 210 autonomously and/or automatically creates or generates an initial or original path, e.g., by autonomously and/or automatically discovering a path from an initial state of SUT 215 to a desired (goal) state of SUT 215. Accordingly, unlike known systems or methods, embodiments of the invention may improve on prior technology by fully or completely test a system, without intervention of a human user, that is, given access or interface to any tested system SUT 215, AT 210 can autonomously identify and test SUT 215, e.g., based on screens displayed AT 210 can automatically identify the system type or OS (e.g., determine whether SUT 215 is Android based or iOS), and, as described, define and execute tests.

[0034] In some embodiments, a test intent is translated into "bot readable" language. For example, a unit adapted to perform text analysis (or other methods known in the art) may be used to analyze information in a file created by a user, e.g., a file that describes test scenarios, a flow from an initial screen or state of SUT 215 to a final (goal) state or screen and the like and the unit may generate, based on the information in the file, a bot readable object (that may itself be a file). Generally, any system or method may be used for converting or translating human input (test intent) into a language or format that is understood and usable by AT 210. A bot readable object may be a text file or other file type that includes values, parameters or any construct or digital object that represent an intention or purpose of a test in a way readable and/or usable by AT 210. Typically,

a test has an intention (test intent) that may be or may include a certain feature or user flow that the test creator (e.g., QA professional, product manager, game designer etc.) wanted tested. The test intent is usually recorded in written or human-readable form (whether on a dedicated system or simply on an Excel sheet or any other human-readable medium or format), sometimes detailing specific steps to perform the test (e.g., initial state -like "open app" and following actions - like "tap here, then swipe left, then input this text" etc.). Any unit or method (e.g., an application that receives an Excel spreadsheet and produces machine readable code) may be used for converting a written form of a test into "bot readable" language, e.g., an Excel sheet listing features to be tested may be automatically converted into an object readable and usable by AT 210 for testing features in the Excel sheet. Any system or method known in the art for converting or translating a document created by a human (e.g., an Excel spreadsheet, a table in a text document etc.) into commands and parameters usable by a machine may be used, e.g., as known in the art with respect to generating scripted tests.

[0035] A test intent may also include or define an expected behavior, e.g., in the form of the final state that the application will be in once all the steps are done and, sometimes, also how/what should the intermediate states look like or be (e.g., "we should get to the 'settings' window" or "we should now have a new dress in the shopping cart"). With current systems and methods, the test intent is not defined explicitly. Instead, current systems and methods resort to comparison of a behavior of an application to the tester's previous experience with the application or specific feature and trusting a human tester to validate that behavior did not change, e.g., the tester is instructed to "make sure it looks fine". Existing scripted tests (used by known systems and methods) have the opposite problem – they are too strict and "literal" in the actions they perform and the states they expect to see/achieve. They are hard to code in the first place and often "break" when the SUT changes – creating a discrepancy between the state the programmer assumed the SUT will be in after an interaction and the real state of the system. Without adaptive awareness to the SUT state this will always be the case.

[0036] In some embodiments, a test intent or definition is translated or converted into a "bot readable" definition that includes an "initial state", an "action goal" or "goal state" and (optionally) additional states, events, interactions or actions that must, or must not be observed, occur or assumed, for example, a bot readable definition may describe the states and interactions or events as shown in Fig. 3. For example, to pass a test or for a flow to be verified, the user (or application)

may be assumed or expected to take several steps to achieve a goal. A "bot readable" definition can include states that must not be reached or assumed by a tested application as well as actions that must never be taken during testing, e.g., actions that can initiate or cause real purchases or money transactions. For example, a bot readable definition can be digital information in a file or memory 120 that enables AT 210 to select one or more actions based on output and/or state of SUT 215. It is noted that AT 210 can operate based on many bot readable definitions, e.g., a first bot readable definition object may be created for, and used for testing, a first SUT 215 (e.g., an android application) and a second, different bot readable definition object may be created, and used for a second SUT 215, e.g., a web-based application.

[0037] In some embodiments, reaching the final or goal state definition is construed as a validation of the test, e.g., the test succeeds when a bot (e.g., AT 210) causes a tested application (e.g., SUT 215) to reach the final state as defined in the bot readable definition, and the test fails if AT 210 cannot cause SUT 215 to reach the final state.

[0038] Various methods for translating or converting the essence or intent of a test into a bot readable definition may be used. For example, a GUI tool usable for defining tests using object/parameter names may be provided. For example, an initial state in a bot readable definition can be "start application" and a final state therein can be "the 'profile' screen", where 'profile' is the name of the screen used within the code. In another case, the bot readable definition includes "load level 2" and then "kill any Orc" in an action game. In some embodiments, AT 210 is integrated into a product or application and can thus retrieve or obtain names and parameters used by SUT 210 (possibly internally), accordingly, AT 210 can, during testing, determine a state, a screen layout and the like. It is noted that AT 210 may be integrated into SUT 215 during testing and removed from a final, shipped version of SUT 215.

[0039] In some embodiments, internal parameters, values or objects of SUT 215 are identified by AT 210 and are then presented (exposed) to a human test creator, e.g., while he/she is using or interacting with SUT 215, e.g., using an overlay in real time, thus making it easy for the tester to understand what exactly are the relevant parameters to notice in each test. Using a GUI tool, the user can then mark the relevant objects, states or parameters for a specific test thus enabling AT 210 to learn which elements are relevant and which are to be ignored during testing.

[0040] In some embodiments, a system and/or method can be used as a debugging tool. For example, during interaction with TA 210, a test creator using the above-mentioned tool has visibility into state parameters and components of the SUT 215 that are not normally visible (like internal variables, memory structures etc.). This visibility is often very useful when trying to understand the root cause of some unwanted and not understood behavior, as it shows the underlying system state that leads to the visible behavior – hence speeding the process of locating the internal function/component that produces the unexpected pattern (debugging).

[0041] Some embodiments include "definition through example" that may be or may include showing instead of telling. For example, a test creator can demonstrate the desired test steps that achieve a legitimate or desired goal state several times, and AT 210 records the steps, actions, interactions, states or other information related to the path which the tester followed including of course, the final state that was achieved. Then AT 210 "reverse engineers" the user's intent and uses that to generate tests as described herein. Accordingly, an embodiment can translate a demonstration into a "bot readable" definition automatically and/or autonomously.

[0042] Some embodiments include generating test scenarios with no user input. AT 210 can use various methods to automatically and/or autonomously create, define or "invent" tests or test scenarios without any involvement of a human tester.

[0043] In some embodiments, AT 210 autonomously explores the state space of SUT 215 by performing, or trying, new interactions or actions thus autonomously and automatically discovering paths 132. Accordingly, AT 210 can achieve complete coverage of a path-space or state-space of SUT 215. The terms "new actions" or "new interactions" merit a discussion. For example, different paths can lead to similar or identical sequences of states (e.g., clicking the same button in different pixels, or filling the username then the password vs. password then username in a login screen can lead to identical results). In some cases, identical actions can lead to different states (for example, if there are random elements in a game and the seed wasn't fixed, or the functionality depends on external input like the weather or today's news). If a sufficiently large number of action flows is generated, (e.g., AT 210 attempts a large number of interactions with a large number of states of SUT 215) then AT 210 can mitigate or overcome the problems of "identical actions leading to different states" or "different interactions can lead to identical states", something scripted tests or other known systems and methods cannot do. In some embodiments,

AT 210 can accelerate the path-space discovery process, for example, AT 210 can be configured to be "curious", e.g., using an element of predictability in a cost function.

[0044] In addition to the validation that SUT 215 does not crash or performs some clearly wrong action (e.g., deleting its own database), AT 210 maps, or identifies paths and relationships between user actions and expected results. AT 210 can alert of any substantial changes in the functionality of SUT 210 due to changes in the code/introduction of a new feature etc. Identifying substantial changes can be achieved via human supervision, e.g., AT 210 shows the user differences that are assumed important and asks "Is this really different than last version?" or "Was this change intentional or accidental?", based on user input, AT 210 can update the functionality expectations (e.g., in a path 132). Accordingly, AT 210 can learn, with time, what the user cares about and what he/she doesn't deem important.

[0045] In the modern age there is an increasing convergence into common design languages on every platform. For example, if one is using an Android app one already knows that the three vertical-points icon means "additional options". In some embodiments, AT 210 automatically and/or autonomously collects and records information about applications' screens, states, structure, and actions, e.g., AT 210 is provided with definitions of GUI or other elements, (e.g. the hamburger icon in Android screens) and is then allowed to freely interact with SUT 215. AT 210 can record paths 132 and/or states 131 it discovers in an unsupervised session. Based on an unsupervised interaction with SUT 215, AT 210 can suggest initial "packages" of commonly used basic functionality tests, user flows, paths 132 and or states 131. For example, since a login flow looks very similar in most Android applications, having interacted with a number of Android applications, AT 210 can record elements frequently seen (and interacted with), and use such elements and their attributes to suggest (or create) a test for an Android login flow.

[0046] Provided with a well-defined test intent (or 'mission'), e.g., in the form of bot readable definition as described, AT 210 can perform the task of finding a sequence of actions (e.g., mouse clicks, taps, swipes, text input, keyboard keys etc.) that carry SUT 215 from an initial state to the goal. The exact scope of what actions AT 210 performs, and what is "visible" to AT 210 as the state of SUT 215 may depend on the type of integration used, the platform that is tested and definitions and limitation provided by a user.

[0047] A few examples of the data and information used for defining paths and states (or state parametrization) are described below. In some embodiments AT 210 is provided with visual information, in the form of pixel data, that represents or reflects screens as seen by a human tester. It is noted that this method requires no (or light or minimal) integration of AT 210 with SUT 215. For example, if tests are run on virtual machines where screens can be captured and analyzed on a computer then AT 210 can be provided with screens presented by SUT 215 without requiring any integration of AT 210 into, or with, SUT 215.

[0048] In some embodiments AT 210 is provided with object locations, speed, texture and other metadata that are mostly relevant to apps/games written on top of graphical engines like Unity and Unreal (as they have a clear data structure that can be easily understood and extracted). In some embodiments AT 210 is provided with Document Object Model (DOM) hierarchy object structure, e.g., for web apps. For example, DOM information can include, or be relevant to, screens/surfaces/URLs that user flows go through. Any other method for determining an output or state of SUT 215 based on information displayed on a screen may be used, e.g., pixel data may be used to identify objects or elements displayed by SUT 215.

[0049] Accordingly, using DOM information, AT 210 can determine or identify the output of SUT 215, e.g., AT 210 cay determine where, on a screen, GUI objects are located, what these GUI objects are (e.g., a Yes/No button, a text box etc.), their states and so on. Accordingly, AT 210 can determine whether or not the output of SUT 215 is as expected or correct, as described, output of SUT 215 may be used for determining a state of SUT 215.

[0050] In some embodiments AT 210 is provided with any information or data related to outbound and/or inbound communications of SUT 215 with other devices, for example, Google analytics events that are sent out when certain events are triggered provide AT 210 with visibility into internal code flows without requiring AT 210 to be integrated with, or into SUT 215. In some embodiments AT 210 is provided with device information like GPS position, system language, geo-ip etc. For example, such information may be relevant for localization testing. In some embodiments AT 210 can access SUT 215's RAM, accordingly, AT 210 can monitor any aspect of SUT 215, e.g., any which system calls, libraries and/or APIs are used etc.

[0051] In some embodiments AT 210 interacts with the digital environment of SUT 215 and/or the underlying platform. For example, AT 210 may have direct access to the computing device

(e.g., PC or smartphone) on which SUT 215 is executed, accordingly, AT 210 is provided with events such as mouse clicks, key presses, swipes and the like and can thus record a complete session including all interactions with SUT 215 as well as any response or output of SUT 215.

[0052] Accordingly, AT 210 may define or create a definition of the state space (S) of SUT 215, the actions that are available at each state (AS), the starting conditions (Si) and the goal state (Sf). It is noted the Sf can be a family of states since the validated conditions are only part of the full state parametrization). In some embodiments, the challenge of generating tests is met using reinforcement learning, e.g., navigating from Si to Sf via a series of actions a1, a2, and so on.

[0053] There are various approaches that can solve "navigation" problems, the simplest one is fuzzing. Using fuzzing, at each state Si a random (possible) action is chosen thus generating a random walk in the state space. Given enough steps and repetitions (restarting the tested app back to si), some paths will lead to the coveted Sf. These paths will be marked as successful and stored, these can be the regression tests that will run in the future. This naive approach may be inefficient (brakes down) once the relevant state space is large (or, more accurately, when the (branching factor) * (number of actions needed to get from Si to Sf)/|sf| is large) because it would take an unreasonably long time to randomly stumble into a final state.

[0054] Some embodiments may use a "navigation" algorithm (e.g., Q-learning) that learns, records and/or maps states and paths it sees when attempting to reach the final goal. On average, any learning method produces successful series of actions faster than random walks, but for complex tests there is still the problem of sparse rewards (which means that AT 210 doesn't know if it's in the right direction until actually getting to a final state, so it mostly wanders around aimlessly like a random walk). There are many methods to combat this problem, approaches used by embodiments may be, for example, a curiosity-driven state exploration. Generally, using a curiosity-driven AT 210 is penalized for getting into states it already visited, and performing actions for which the result is already known, accordingly, a curiosity-driven approach pushes AT 210 to cover more possibilities it didn't see before thus speeding up the discovery of a final, desired state.

[0055] In some embodiments, a test creator can be asked to demonstrate a successful run which may be recorded and used as an initial reference based on a training may be performed. Yet another approach used by some embodiments may be reward shaping. Generally, reward shaping is an

approach that translates sparse, rare and discreet (success/failure) goal functions into continuous, omnipresent reward functions that helps guiding the algorithm to the final state. Reward shaping can be viewed as applying a gentle slope to the state space that causes AT 210 to be nudged in the right direction. This can sometimes be generated from the test definition itself (for example, a step that says "player must get close to an orc" translates into a reward function that minimizes the distance to the closest orc in the level/screen).

[0056] The slope in reward shaping can be inferred from past experience with a tested application (for example, if you must first login before you're able to perform most actions in an app then this pattern can in principal be generalized from the many other tests that already figured out the login flow).

[0057] If the AT 210 fails to generate action flows that lead to desired final states for a certain test, the test creator can be asked to break the test into several smaller steps each ending in a sub-goal or checkpoint. This method effectively shortens the number of actions AT 210 has to find autonomously at each step making the test generation process easier and/or substantially shorter.

[0058] In some embodiments, AT 210 starts with several successful (original, possibly user defined) paths (e.g., paths that end at a desired state) and then slightly changes the paths, e.g., adding interactions to a path thus finding changes/permutations that still lead to a desired final state. The original (or known to reach a desired state) paths can be user generated, lucky random flows or other initial conditions. This approach of path enrichment can automatically and autonomously generate a multitude of action flows or paths that take SUT 215 from a state Si to a state in Sf. A multitude of tests or paths created as described can then be used for regression, e.g., used for tests for future code changes.

[0059] The flows, paths and states defined or discovered as described may be different in a real sense, e.g., not just pressing the same button shifted by one pixel to the right, and still get to the same final, desired state. For example, using the method described, AT 210 can discover flows, paths or states that are different from the original paths, that is, AT 210 discovers flows or paths in which, with respect to known or provided or original paths, different states are visited, different interactions are used for same screens, however, the goal or desired state is still reached and the paths discovered can be viewed as successful or good paths.

[0060] Flows (or paths) may be sorted or grouped, e.g., according to their length, and different tests or coverage of the functionality of SUT 215 may be generated based on a group or class of flows or paths.

[0061] Yet another approach used by some embodiments includes skipping the concept of action flows that are generated and later performed. Instead, the paradigm can be flipped and a system can treat a trained bot (e.g., AT 210) as the test itself. A well-trained AT 210 with the right model architecture should be able to achieve the desired final state even if there were small changes in the path (location of the button, the initial position of the orc etc.) - thus being inherently robust to irrelevant changes. Instead of 40 action sequences, the system would just let the bot run and see if it reaches its goal.

[0062] Test implementations currently known are hard-coded sets of actions (meant to achieve a certain intent) and further include a validation step on the final state (making sure said intent has been achieved). Taking a different approach to improve on such technology, embodiments of the invention may take as input a bot readable definition that is designed to achieve an intent using a flow or path of interactions, events and/or actions, and then automatically and/or autonomously, improve the test, e.g., a bot readable definition can be updated or modified, by AT 210, based on results of a test run. In addition, new tests, flows or paths may be generated, and the new paths can replace previous ones, e.g., ones that broke due to changes made to SUT 215.

[0063] In some embodiments, AT 210 is an "intention driven bot". When operating in intention-driven mode, AT 210 no longer needs to find and record or remember specific flows of the SUT 215, instead, AT 210 automatically and/or autonomously learns to achieve a goal or state. Any knowledge and understanding related to SUT 215 can be stored, e.g., in a neural network system or object. It will be appreciated by a person of skill in the art that a bot that knows an SUT as described is far superior to any system or method that requires to know sets of action states in order to function. For example, AT 210 may be given nothing but the goal of "send an email" and, using its knowledge of an application (e.g., Gmail) and/or a platform (e.g., Android) AT 210 may find many (or even all) paths, interactions and states that will carry a user from any screen to the mail application and cause the mail application to send an email.

[0064] In order to accelerate the process of path mapping, intent completion and later regression testing and validation, AT 210 may employ various techniques to find short and efficient paths to

the given goal. For example, paths that are generated during the exploratory phase and contain loops (series of actions that lead from a certain state back to the same exact state) are shortened by removing the loops from the action-path. Another algorithm may be "action unification" – namely identifying automatically that certain actions that seem different actually lead to the exact same result in certain states and unifying them into a single action – sometimes without any a-priory knowledge about the product or input from the user (for example – in the above Gmail example clicking the "send" button in the middle pixel or 3 pixels to the right will result in the exact same result every time, so the actions "click middle pixel" and "click slightly right pixel" are learned to be identical in the current state – the "send email" screen).

[0065] AT 210's ability to interact with a SUT 215 as described enables AT 210 to achieve a goal or test intent even in the face of some changes in the structure of SUT 215. For example, in the above email example, if after an update of SUT 215, the path or way for sending an email using SUT 215 changes, AT 210 operating in "intent driven bot" mode can reach a goal of sending an email via a different set of interactions – without retraining or specific new path discovery.

[0066] Using its ability to discover paths as described, AT 210 can automatically identify, in a set of paths, an optimal path. For example, AT 210 can discover or identify a number of ways to cause SUT 215 to send an email. Based on rules or criteria, e.g., the number of steps or interactions required to send an email may be a criterion and AT 210 may select the path with the smallest number of steps as the optimal path, in another example, response time of SUT 215 may be a criterion and thus the optimal path may be the response-wise fastest path.

[0067] AT 210 may automatically mark paths as failed or redundant. For example, AT 210 may record, for a number of SUT 215, paths to goals. For example, AT 210 records all paths it found that start at a home screen of an Android device and end at sending an email. AT 210 can further mark the optimal paths as described. When starting a new test (e.g., for a new version of SUT 215), AT 210 can first attempt to send an email using the recorded paths. If a recorded or known path fails to reach a target or goal then AT 210 may mark the path as failed. Accordingly, AT 210 can provide valuable information to a user, e.g., report a way (path) for sending an email that worked in a previous version of SUT 215 no longer works.

[0068] Typically, if there are no changes in SUT 215 functionality, the generated flows from a previous run or iteration will all succeed in reaching the expected final state, e.g., returning a "pass"

for each test case. However, if following a change of SUT 215, e.g., when a new version is released, some or even all tests might fail. In such case, AT 210 first tries to run the same training methods to generate alternative paths to reach the same goal for the new version of SUT 215. In case that doesn't work (e.g., AT 210 determines, to a certain degree of confidence, that the desired state cannot be reached) then AT 210 marks the test as "failed" and reports to a user. As described, AT 210 may then automatically and/or autonomously discover ways (paths) to reach a goal or desired state, e.g., if AT 210 fails to cause SUT 215 to send an email using recorded or known interactions or paths, AT 210 may proceed to looking for new ways, interactions or paths that will cause SUT 215 to send an email.

[0069] In some embodiments, a portion of a path (or sequence of actions) may fail to get to the same goal as before, but another portion of the path does succeed. In such case AT 210 generates new flows (e.g., as described herein) to replace the previous broken ones and marks the test scenario as "pass*" thus reporting that some functionalities leading to the goal in testing have changed since the tests were last ran, but the feature seems to be operating correctly.

[0070] In case the test was marked as "fail", one of the failed paths is chosen, its actions performed again and the activity can be recorded, e.g., as a video, to be shown to a user as part of the fail report. A possible improvement to this feature can be using the multitude of paths to pinpoint as accurately as possible the stage where they deviate from the expected path, e.g., recording which action now leads to a state that is different from the state it led to when a previous version of SUT 215 was tested. This can be done by a pattern matching algorithm if the (state, action) sequences are fully stored for all test flows, e.g., in paths 132. Accordingly, AT 210 can show a user the specific functions of SUT 215 that now behave differently with respect to a previous or other version of SUT 215.

[0071] Reference is made to Fig. 4, a flow of a method according to some embodiments of the invention. As shown by block 410, an auto testing unit may be provided with an interface with a system under test. For example, AT 210 may be provided access to a screen used by SUT 215 to provide interface to a user and AT 210 may use the screen to interact with SUT 215, e.g., simulate or cause button presses, swipe actions and the like. Generally, AT 210 may use an interface provided as described to perform any actions that are normally performed by a user or application when interacting with SUT 215.

[0072] As shown by block 420, one or more goals may be provided to an auto tester unit. For example, a set of goals such as send an email, take a picture, send a text message and so on may be given to AT 210 as described. As shown by block 430, an auto testing unit may autonomously discover one or more ways to cause a system under test to complete the goals. For example, AT 210 may discover ways (paths) to cause SUT 215 to send an email as described.

[0073] In the description of the present application, each of the verbs, "comprise" "include" and "have", and conjugates thereof, are used to indicate that the object or objects of the verb are not necessarily a complete listing of components, elements or parts of the subject or subjects of the verb. Unless otherwise stated, adjectives such as "substantially" and "about" modifying a condition or relationship characteristic of a feature or features of an embodiment of the disclosure, are understood to mean that the condition or characteristic is defined to within tolerances that are acceptable for operation of an embodiment as described. In addition, the word "or" is considered to be the inclusive "or" rather than the exclusive or, and indicates at least one of, or any combination of items it conjoins.

[0074] Descriptions of embodiments of the invention in the present application are provided by way of example and are not intended to limit the scope of the invention. The described embodiments comprise different features, not all of which are required in all embodiments. Some embodiments utilize only some of the features or possible combinations of the features. Variations of embodiments of the invention that are described, and embodiments comprising different combinations of features noted in the described embodiments, will occur to a person having ordinary skill in the art.

[0075] While certain features of the invention have been illustrated and described herein, many modifications, substitutions, changes, and equivalents may occur to those skilled in the art. Various embodiments have been presented. Each of these embodiments may of course include features from other embodiments presented, and embodiments not specifically described may include various features described herein.

CLAIMS

1. A system substantially as described in the specification and drawings.

3. A method as described in the specification and drawings.

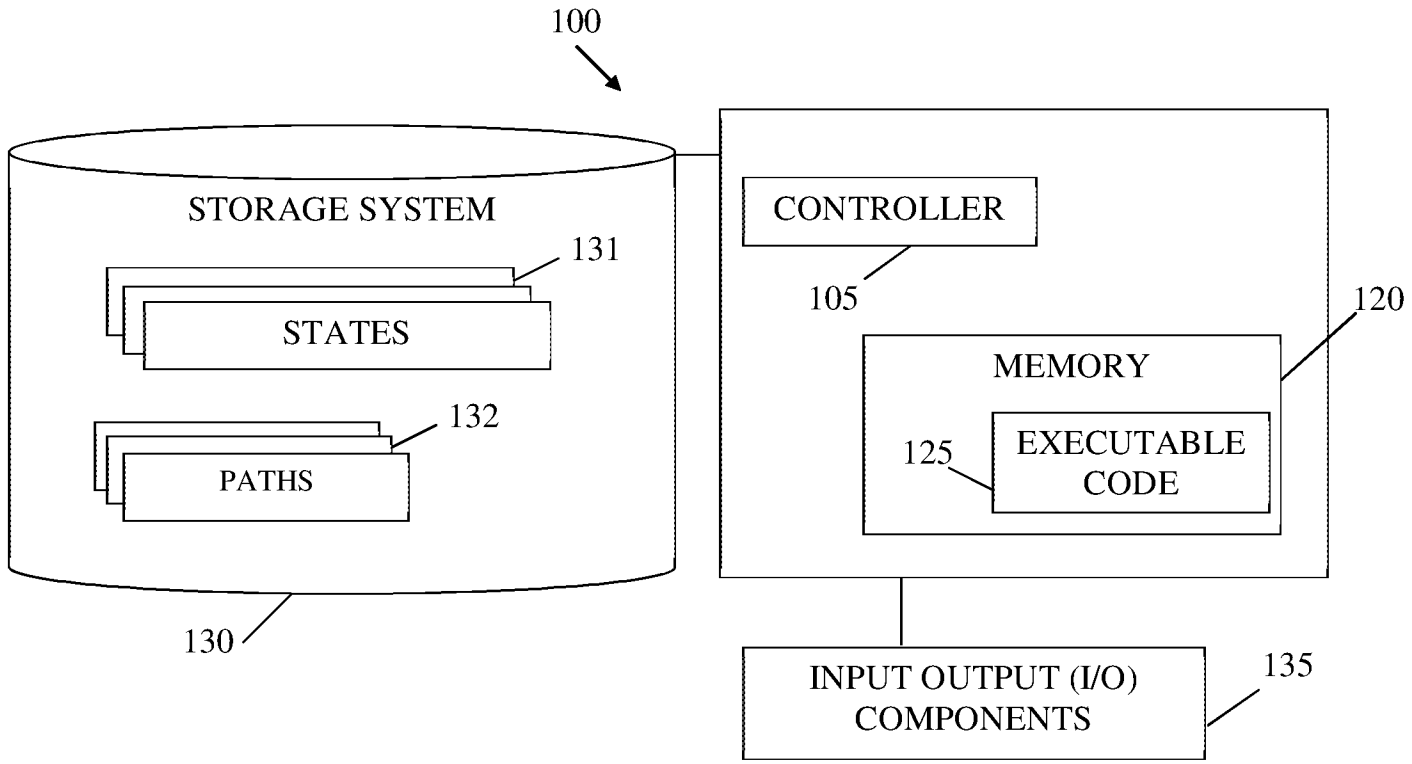
100
STORAGE SYSTEM
131
STATES
132
PATHS
130
CONTROLLER
105
120
MEMORY
125
EXECUTABLE CODE
INPUT OUTPUT (I/O) COMPONENTS
135

Fig. 1

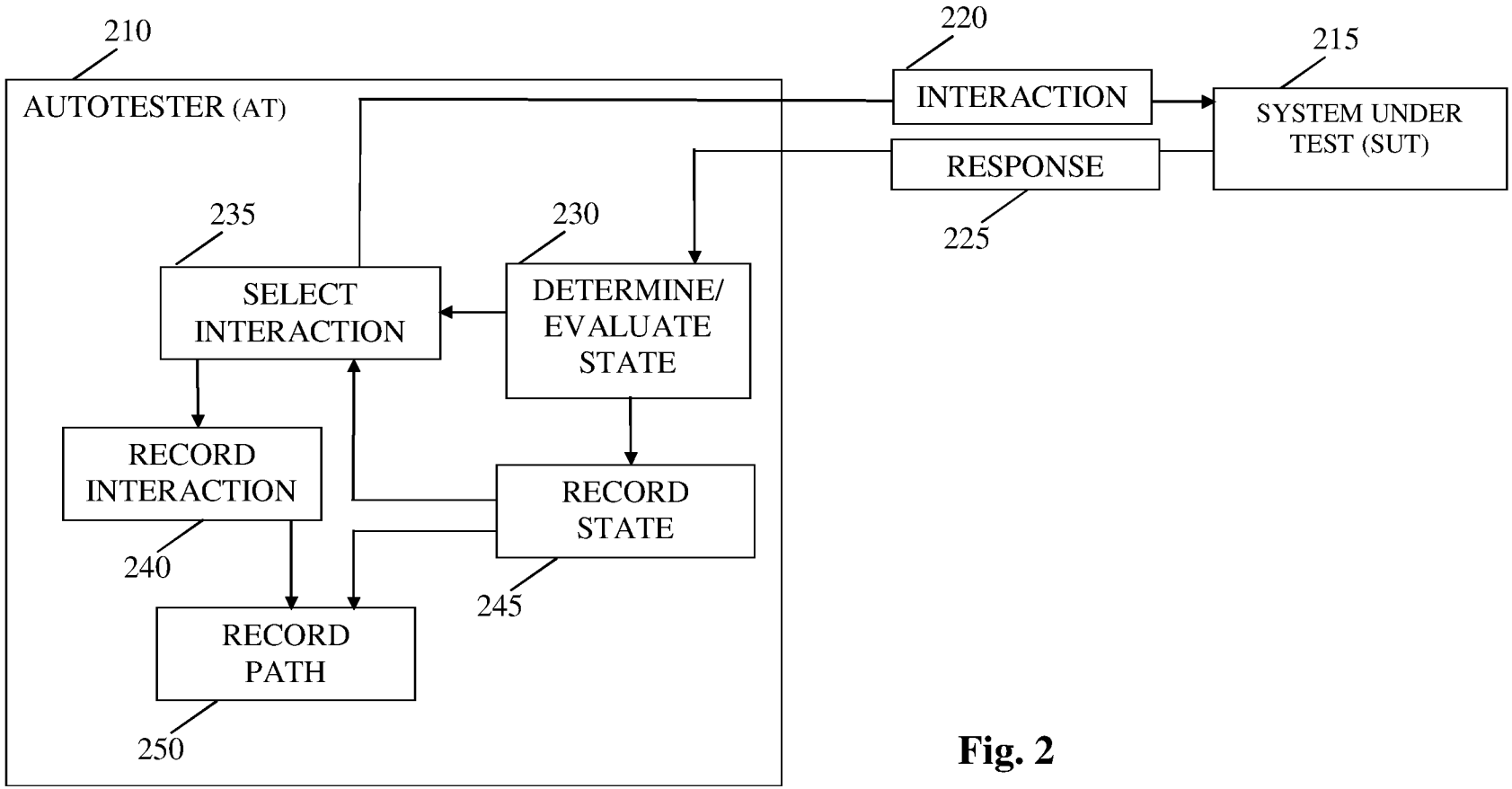
210
AUTOTESTER (AT)
220
INTERACTION
215
SYSTEM UNDER TEST (SUT)
RESPONSE
225
235
SELECT INTERACTION
230
DETERMINE/ EVALUATE STATE
RECORD INTERACTION
240
RECORD STATE
245
RECORD PATH
250

Fig. 2

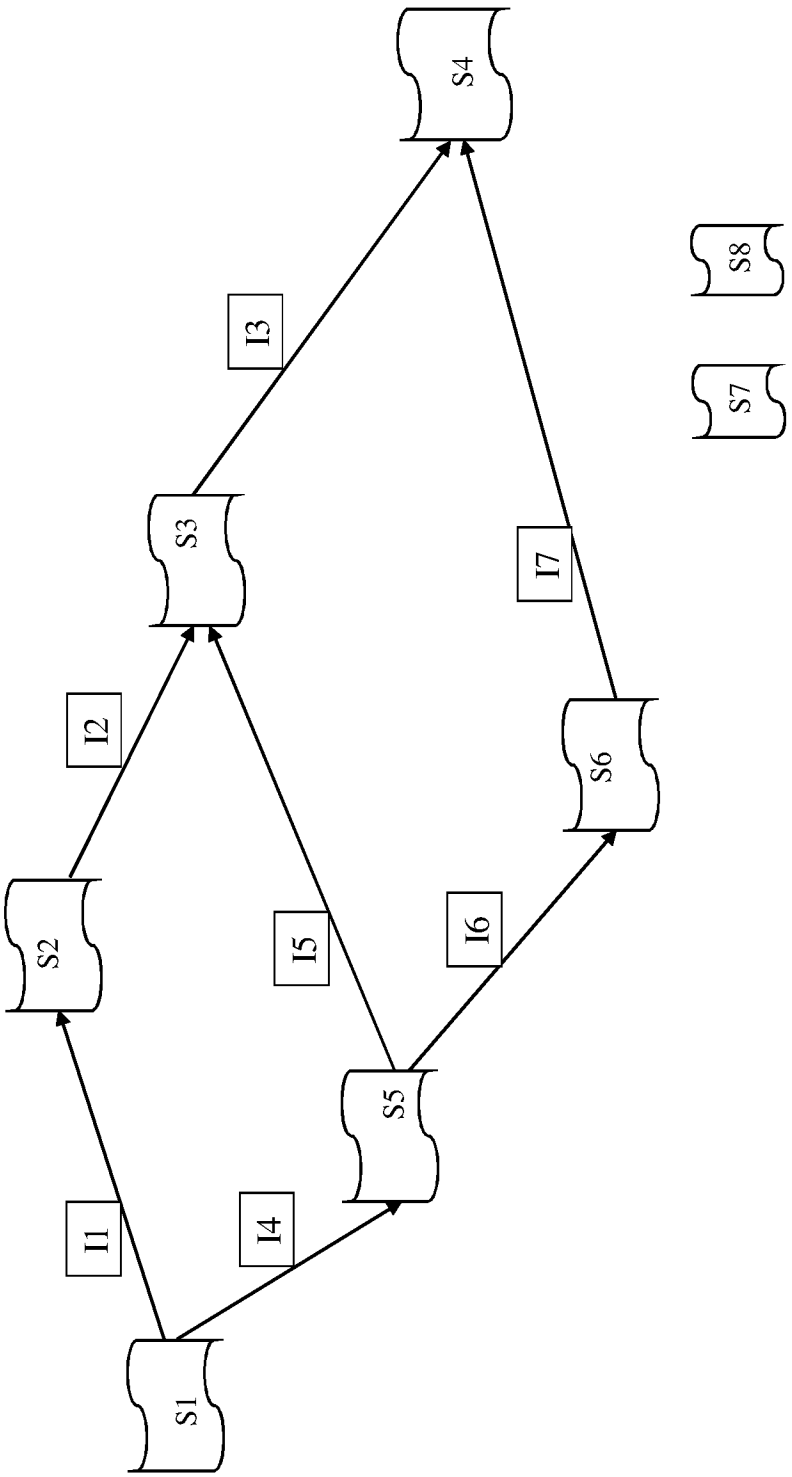

Fig. 3

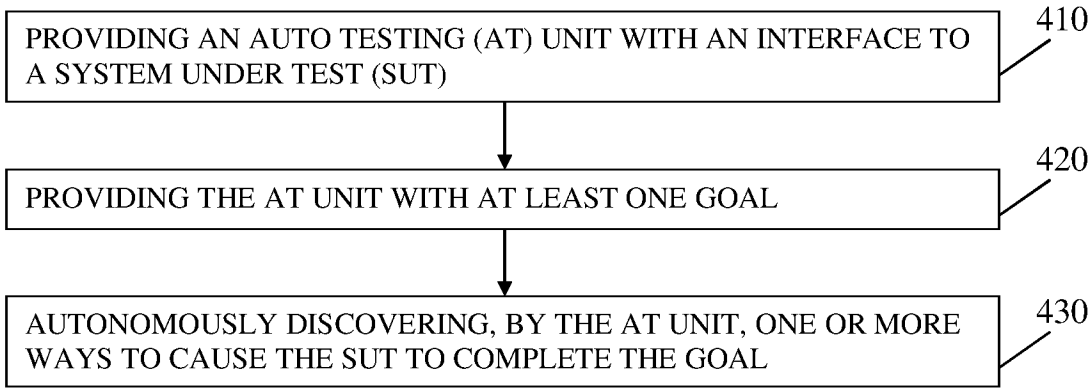
PROVIDING AN AUTO TESTING (AT) UNIT WITH AN INTERFACE TO A SYSTEM UNDER TEST (SUT)
410
PROVIDING THE AT UNIT WITH AT LEAST ONE GOAL
420
AUTONOMOUSLY DISCOVERING, BY THE AT UNIT, ONE OR MORE WAYS TO CAUSE THE SUT TO COMPLETE THE GOAL
430

Fig. 4

PTO/AIA/14 (11-15)
Approved for use through 04/30/2017. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	P-582424-USP
	Application Number	
Title of Invention	SYSTEM AND METHOD FOR AUTOMATED TEST GENERATION AND TESTING AN APPLICATIONS	

The application data sheet is part of the provisional or nonprovisional application for which it is being submitted. The following form contains the bibliographic data arranged in a format specified by the United States Patent and Trademark Office as outlined in 37 CFR 1.76. This document may be completed electronically and submitted to the Office in electronic format using the Electronic Filing System (EFS) or the document may be printed and included in a paper filed application.

Secrecy Order 37 CFR 5.2

☐ Portions or all of the application associated with this Application Data Sheet may fall under a Secrecy Order pursuant to 37 CFR 5.2 (Paper filers only. Applications that fall under Secrecy Order may not be filed electronically.)

Inventor Information:

Inventor 1 Remove

Legal Name

Prefix	Given Name	Middle Name	Family Name	Suffix
	Michael		SHALYT	

Residence Information (Select One) ☐ US Residency ◉ Non US Residency ☐ Active US Military Service

City	Haifa	State/Province		Country of Residence	Israel

Mailing Address of Inventor:

Address 1	3 Gat St.
Address 2	

City	Haifa	State/Province	
Postal Code	3475802	Country	Israel

All Inventors Must Be Listed - Additional Inventor Information blocks may be generated within this form by selecting the **Add** button. Add

Inventor 2 Remove

Legal Name

Prefix	Given Name	Middle Name	Family Name	Suffix
	Nataly		BENDERSKY	

Residence Information (Select One) ☐ US Residency ◉ Non US Residency ☐ Active US Military Service

City	Tel Aviv	State/Province		Country of Residence	Israel

Mailing Address of Inventor:

Address 1	17 Yosef Karo St.
Address 2	

City	Tel Aviv	State/Province	
Postal Code	6701421	Country	Israel

All Inventors Must Be Listed - Additional Inventor Information blocks may be generated within this form by selecting the **Add** button. Add

PTO/AIA/14 (08-12)
Approved for use through 01/31/2014. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	P-582424-USP
	Application Number	
Title of Invention	SYSTEM AND METHOD FOR AUTOMATED TEST GENERATION AND TESTING AN APPLICATIONS	

Correspondence Information:

Enter either Customer Number or complete the Correspondence Information section below. For further information see 37 CFR 1.33(a).			
☐ An Address is being provided for the correspondence Information of this application.			
Customer Number	49443		
Email Address	USPTO@PearlCohen.com	Add Email	Remove Email

Application Information:

Title of the Invention	SYSTEM AND METHOD FOR AUTOMATED TEST GENERATION AND TESTING AN APPLICATIONS		
Attorney Docket Number	P-582424-USP	Small Entity Status Claimed ☒	
Application Type	Provisional		
Subject Matter	Utility		
Total Number of Drawing Sheets (if any)	4	Suggest Figure for Publication (if any)	

Filing By Reference:

Only complete this section when filing an application by reference under 35 U.S.C. 111(c) and 37 CFR 1.57(a). Do not complete this section if application papers including a specification and any drawings are being filed. Any domestic benefit or foreign priority information must be provided in the appropriate section(s) below (i.e., "Domestic Benefit/National Stage Information" and "Foreign Priority Information").

For the purposes of a filing date under 37 CFR 1.53(b), the description and any drawings of the present application are replaced by this reference to the previously filed application, subject to conditions and requirements of 37 CFR 1.57(a).

Application number of the previously filed application	Filing date (YYYY-MM-DD)	Intellectual Property Authority or Country

Publication Information:

☐	Request Early Publication (Fee required at time of Request 37 CFR 1.219)
☐	**Request Not to Publish.** I hereby request that the attached application not be published under 35 U.S.C. 122(b) and certify that the invention disclosed in the attached application **has not and will not** be the subject of an application filed in another country, or under a multilateral international agreement, that requires publication at eighteen months after filing.

Representative Information:

Representative information should be provided for all practitioners having a power of attorney in the application. Providing this information in the Application Data Sheet does not constitute a power of attorney in the application (see 37 CFR 1.32). Either enter Customer Number or complete the Representative Name section below. If both sections are completed the customer Number will be used for the Representative Information during processing.

Please Select One:	☒ Customer Number	☐ US Patent Practitioner	☐ Limited Recognition (37 CFR 11.9)
Customer Number	49443		

PTO/AIA/14 (08-12)
Approved for use through 01/31/2014. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	P-582424-USP
	Application Number	
Title of Invention	SYSTEM AND METHOD FOR AUTOMATED TEST GENERATION AND TESTING AN APPLICATIONS	

Domestic Benefit/National Stage Information:

This section allows for the applicant to either claim benefit under 35 U.S.C. 119(e), 120, 121, or 365(c) or indicate National Stage entry from a PCT application. Providing this information in the application data sheet constitutes the specific reference required by 35 U.S.C. 119(e) or 120, and 37 CFR 1.78.

When referring to the current application, please leave the "Application Number" field blank.

Prior Application Status			Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
Prior Application Status			Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)

Additional Domestic Benefit/National Stage Data may be generated within this form by selecting the **Add** button. Add

Foreign Priority Information:

This section allows for the applicant to claim priority to a foreign application. Providing this information in the application data sheet constitutes the claim for priority as required by 35 U.S.C. 119(b) and 37 CFR 1.55(d). When priority is claimed to a foreign application that is eligible for retrieval under the priority document exchange program (PDX) the information will be used by the Office to automatically attempt retrieval pursuant to 37 CFR 1.55(h)(1) and (2). Under the PDX program, applicant bears the ultimate responsibility for ensuring that a copy of the foreign application is received by the Office from the participating foreign intellectual property office, or a certified copy of the foreign priority application is filed, within the time period specified in 37 CFR 1.55(g)(1).

Remove

Application Number	Country	Filing Date (YYYY-MM-DD)	Access Code (if applicable)

Additional Foreign Priority Data may be generated within this form by selecting the **Add** button. Add

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications

☐ This application (1) claims priority to or the benefit of an application filed before March 16, 2013 and (2) also contains, or contained at any time, a claim to a claimed invention that has an effective filing date on or after March 16, 2013.

NOTE: By providing this statement under 37 CFR 1.55 or 1.78, this application, with a filing date on or after March 16, 2013, will be examined under the first inventor to file provisions of the AIA.

Authorization or Opt-Out of Authorization to Permit Access:

When this Application Data Sheet is properly signed and filed with the application, applicant has provided written authority to permit a participating foreign intellectual property (IP) office access to the instant application-as-filed (see paragraph A in subsection 1 below) and the European Patent Office (EPO) access to any search results from the instant application (see paragraph B in subsection 1 below).

Should applicant choose not to provide an authorization identified in subsection 1 below, applicant must opt-out of the authorization by checking the corresponding box A or B or both in subsection 2 below.

PTO/AIA/14 (08-12)
Approved for use through 01/31/2014. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	P-582424-USP
	Application Number	
Title of Invention	SYSTEM AND METHOD FOR AUTOMATED TEST GENERATION AND TESTING AN APPLICATIONS	

NOTE: This section of the Application Data Sheet is **ONLY** reviewed and processed with the **INITIAL** filing of an application. After the initial filing of an application, an Application Data Sheet cannot be used to provide or rescind authorization for access by a foreign IP office(s). Instead, Form PTO/SB/39 or PTO/SB/69 must be used as appropriate.

1. Authorization to Permit Access by a Foreign Intellectual Property Office(s)

A. Priority Document Exchange (PDX) - Unless box A in subsection 2 (opt-out of authorization) is checked, the undersigned hereby **grants the USPTO authority** to provide the European Patent Office (EPO), the Japan Patent Office (JPO), the Korean Intellectual Property Office (KIPO), the State Intellectual Property Office of the People's Republic of China (SIPO), the World Intellectual Property Organization (WIPO), and any other foreign intellectual property office participating with the USPTO in a bilateral or multilateral priority document exchange agreement in which a foreign application claiming priority to the instant patent application is filed, access to: (1) the instant patent application-as-filed and its related bibliographic data, (2) any foreign or domestic application to which priority or benefit is claimed by the instant application and its related bibliographic data, and (3) the date of filing of this Authorization. See 37 CFR 1.14(h)(1).

B. Search Results from U.S. Application to EPO - Unless box B in subsection 2 (opt-out of authorization) is checked, the undersigned hereby **grants the USPTO authority** to provide the EPO access to the bibliographic data and search results from the instant patent application when a European patent application claiming priority to the instant patent application is filed. See 37 CFR 1.14(h)(2).

The applicant is reminded that the EPO's Rule 141(1) EPC (European Patent Convention) requires applicants to submit a copy of search results from the instant application without delay in a European patent application that claims priority to the instant application.

2. Opt-Out of Authorizations to Permit Access by a Foreign Intellectual Property Office(s)

☐ A. Applicant **DOES NOT** authorize the USPTO to permit a participating foreign IP office access to the instant application-as-filed. If this box is checked, the USPTO will not be providing a participating foreign IP office with any documents and information identified in subsection 1A above.

☐ B. Applicant **DOES NOT** authorize the USPTO to transmit to the EPO any search results from the instant patent application. If this box is checked, the USPTO will not be providing the EPO with search results from the instant application.

NOTE: Once the application has published or is otherwise publicly available, the USPTO may provide access to the application in accordance with 37 CFR 1.14.

Applicant Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.

Applicant 1

If the applicant is the inventor (or the remaining joint inventor or inventors under 37 CFR 1.45), this section should not be completed. The information to be provided in this section is the name and address of the legal representative who is the applicant under 37 CFR 1.43; or the name and address of the assignee, person to whom the inventor is under an obligation to assign the invention, or person who otherwise shows sufficient proprietary interest in the matter who is the applicant under 37 CFR 1.46. If the applicant is an applicant under 37 CFR 1.46 (assignee, person to whom the inventor is obligated to assign, or person who otherwise shows sufficient proprietary interest) together with one or more joint inventors, then the joint inventor or inventors who are also the applicant should be identified in this section. Clear

☐ Assignee	☐ Legal Representative under 35 U.S.C. 117	☐ Joint Inventor
☒ Person to whom the inventor is obligated to assign.	☐ Person who shows sufficient proprietary interest	

If applicant is the legal representative, indicate the authority to file the patent application, the inventor is:

Name of the Deceased or Legally Incapacitated Inventor :	

If the Assignee is an Organization check here. ☒

PTO/AIA/14 (08-12)
Approved for use through 01/31/2014. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	P-582424-USP
	Application Number	
Title of Invention	SYSTEM AND METHOD FOR AUTOMATED TEST GENERATION AND TESTING AN APPLICATIONS	

Organization Name	DESTINATION LABS LTD.

Prefix	Given Name	Middle Name	Family Name	Suffix

Mailing Address Information For Applicant:			
Address 1	17 Yosef Karo St.		
Address 2			
City	Tel Aviv	State/Province	
Country Israel		Postal Code	6701421
Phone Number		Fax Number	
Email Address			

Additional Applicant Data may be generated within this form by selecting the Add button. Add

Assignee Information including Non-Applicant Assignee Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.

Assignee 1

Complete this section if assignee information, including non-applicant assignee information, is desired to be included on the patent application publication. An assignee-applicant identified in the "Applicant Information" section will appear on the patent application publication as an applicant. For an assignee-applicant, complete this section only if identification as an assignee is also desired on the patent application publication.

Prefix	Given Name	Middle Name	Family Name	Suffix

If the Assignee or Non-Applicant Assignee is an Organization check here. ☒

Organization Name	DESTINATION LABS LTD.

Mailing Address Information For Assignee including Non-Applicant Assignee:			
Address 1	17 Yosef Karo St.		
Address 2			
City	Tel Aviv	State/Province	
Country Israel		Postal Code	6701421
Phone Number		Fax Number	
Email Address			

Additional Assignee or Non-Applicant Assignee Data may be generated within this form by selecting the Add button. Add

PTO/AIA/14 (08-12)
Approved for use through 01/31/2014. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	P-582424-USP
	Application Number	
Title of Invention	SYSTEM AND METHOD FOR AUTOMATED TEST GENERATION AND TESTING AN APPLICATIONS	

Signature:

Remove

NOTE: This Application Data Sheet must be signed in accordance with 37 CFR 1.33(b). **However, if this Application Data Sheet is submitted with the INITIAL filing of the application and either box A or B is not checked in subsection 2 of the "Authorization or Opt-Out of Authorization to Permit Access" section, then this form must also be signed in accordance with 37 CFR 1.14(c).**

This Application Data Sheet **must** be signed by a patent practitioner if one or more of the applicants is a **juristic entity** (e.g., corporation or association). If the applicant is two or more joint inventors, this form must be signed by a patent practitioner, alljoint inventors who are the applicant, or one or more joint inventor-applicants who have been given power of attorney (e.g., see USPTO Form PTO/AIA/81) on behalf of **all** joint inventor-applicants.

See 37 CFR 1.4(d) for the manner of making signatures and certifications.

Signature	/Devorah Livneh, Reg. No. 71,347/			Date (YYYY-MM-DD)	2019-03-18
First Name	Devorah	Last Name	Livneh	Registration Number	71,347

Additional Signature may be generated within this form by selecting the Add button. Add

Electronic Patent Application Fee Transmittal

Application Number:	
Filing Date:	
Title of Invention:	SYSTEM AND METHOD FOR AUTOMATED TEST GENERATION AND TESTING AN APPLICATIONS
First Named Inventor/Applicant Name:	Michael SHALYT
Filer:	Devorah N Livneh/Sigal Vegh
Attorney Docket Number:	P-582424-USP

Filed as Small Entity

Filing Fees for Provisional

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
PROVISIONAL APPLICATION FILING FEE	2005	1	140	140
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Extension-of-Time:				
Miscellaneous:				
			Total in USD ($)	**140**

Electronic Acknowledgement Receipt

EFS ID:	35445458
Application Number:	62819690
International Application Number:	
Confirmation Number:	7664
Title of Invention:	SYSTEM AND METHOD FOR AUTOMATED TEST GENERATION AND TESTING AN APPLICATIONS
First Named Inventor/Applicant Name:	Michael SHALYT
Customer Number:	49443
Filer:	Devorah N Livneh/Sigal Vegh
Filer Authorized By:	Devorah N Livneh
Attorney Docket Number:	P-582424-USP
Receipt Date:	18-MAR-2019
Filing Date:	
Time Stamp:	05:28:57
Application Type:	Provisional

Payment information:

Submitted with Payment	yes
Payment Type	CARD
Payment was successfully received in RAM	$140
RAM confirmation Number	031819INTEFSW05294300
Deposit Account	602041
Authorized User	Sigal Vegh

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

37 CFR 1.16 (National application filing, search, and examination fees)

37 CFR 1.17 (Patent application and reexamination processing fees)

37 CFR 1.19 (Document supply fees)

37 CFR 1.20 (Post Issuance fees)

37 CFR 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		P-582424-USP-U-APP-18MAR19.pdf	261295 d881ac521fb31424180796155b3d3cbb6f19c8df	yes	31

	Multipart Description/PDF files in .zip description		
	Document Description	**Start**	**End**
	Application Data Sheet	26	31
	Drawings-only black and white line drawings	22	25
	Claims	21	21
	Specification	2	20
	Provisional Cover Sheet (SB16)	1	1

Warnings:

Information:

2	Fee Worksheet (SB06)	fee-info.pdf	29590 10070aed79e5819c9d22c0deffd0665997a97d1a	no	2

Warnings:

Information:

Total Files Size (in bytes):	290885

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.