<u>בחינה בעריכת פטנט בתחום המחשבים, במועד: 29.6.2015</u>

ממציא מגיע אליך ומסביר שהגה רעיון ליצירת הנחיה אוטומטית של משתמשים בתוכנות ואפליקציות שונות, המדריכה אותם מה ואיך לעשות, ואף מבצעת פעולות מבוקשות. הממציא מסביר שאמצאתו אינה תלוית-תצוגה כיוון שמזהה את תמונת פקד השליטה הרלבנטי (ה-widget שעליו צריך להיות הסמן [ה-cursor] כש'לוחצים' עליו עם העכבר או במגע). על פי אמצאתו, מנתחים אוטומטית תמונת מסך של ממשק משתמש גרפי (GUI) על מנת לאתר פס גלילה או פרמטרים שלו, מה שמאפשר להפעיל אוטומטית את פס הגלילה כדי לגלול את המסך לאזורים נוספים (שהם נסתרים טרם גלילת המסך האוטומטית) ולנתח בהם תמונות מסך נוספות. מתמונת/ות המסך הרלבנטית/יות מקבלים את מיקום הסמן כשהמשתמש לוחץ על פקד השליטה לצורך ביצוע פעולה מבוקשת, מנתחים את תמונת האזור שסביב הסמן כדי לזהות את תמונת פקד השליטה הרלבנטי, ומשתמשים בתמונה לצורך הדרכה לביצוע הפעולה או לצורך ביצועה בפועל, בעת יצירת תסריט הנחיה (guidance script).

שיטה ליצירת סימולציות על פי הידע הקודם מתוארת ב"פרק רקע" המצ"ב.

הממציא מספק לך את התיאור המצורף.

ערוך בקשה לפטנט המגלם חידוש והתקדמות המצאתית ביחס לידע הקודם שבפרק הרקע, כולל תאור המתייחס לשרטוט/ים ותביעות. ניתן להעזר בחומר שסיפק הלקוח.

חלוקת הנקודות באחוזים (ציון עובר מינימלי: 65):

- תיאור - 40
- מערכת תביעות לארה"ב - 35
- לפחות תביעה אחת עצמאית לישראל + תיאור השינויים הנדרשים - 10
- לפחות תביעה אחת עצמאית לאירופה + תיאור השינויים הנדרשים - 10
- שפה - 5

<u>גלישה באינטרנט במהלך הבחינה אסורה בתכלית, וצפויה להוביל לפסילה!</u>

בהצלחה!

Background:

Simulations are produced in the software industry for product tutorials and marketing presentations. These simulations are useful for demonstrating a product and teaching how to use specific features in a product. Currently, software simulations are developed with labor intensive techniques. One production technique uses a video camera to film screen images produced as an experienced user physically steps through a software procedure. After filming the entire procedure, the video tape may be edited to add captions or a sound narrative. Because significant editing of a video tape is costly and time consuming, a procedure is usually re-filmed when an error must be corrected, or if a change in the software product prior to its initial release occurs that affects the procedure shown in the video. The edited video tape of the simulation is typically digitized into an animation format file that is distributed to prospective users of the product, or included with the software product to teach new users about and how to use the software product. When included with the software product, the animation is often accessed as a tutorial option under a Help menu topic. The animation file can be distributed alone or as part of a software product on various digital media such as floppy disks or CD-ROMs, and can be transmitted over networks, as is often done when used as part of a "demo" of a product.

Another common simulation production technique is to record all on-screen activity implemented to demonstrate features of a software product with a screen capture program such as Lotus ScreenCam. The author of a simulation launches a screen capture program that records the physical actions of an expert user, while stepping through a software procedure. The actions may include moving a mouse cursor to specific areas on the screen, selecting program options in dialog boxes and menus, and entering text on a keyboard. After recording the entire procedure in an animation format file, the simulation may be edited to add captions and sound. Significant editing of an animation format file is time consuming, and the procedure usually has to be recorded again when an error must be corrected, or if a change occurs in the software program prior to its distribution that affects the accuracy of the simulation.

A further simulation production technique is to create a separate software program that simulates a particular software product. In this case, the separate software program automatically steps through an entire procedure of the software product to be simulated. This separate software simulation program could be written in the language "C" or some other suitable high-level language. However, significant technical expertise is required to author a separate software simulation program, and changes to the program can only be made by a highly skilled computer programmer.

A significant problem in the creation of a tutorial simulation arises because a product tutorial is normally produced before development of a software product is concluded. As a product is undergoing development,

significant changes to the screens, dialogs, menus, command structure, and user interface may occur just prior to the release date. During this developmental process, it may be necessary to recreate the tutorial simulation repeatedly to remain current with the latest version of the software product.

A Personal Computer (PC) or a mobile device, such as a notebook computer, and a cellular phone, allows users to utilize various applications, for example, word processing applications, spreadsheet applications, e-mail applications, and games. These applications may be able to perform various operations based on the user's command, such as editing, calculating, formatting, file handling, and data sorting. A Graphical User Interface (*GUI*) of an application usually includes various controls ("*widgets*") that operate different functions of the application. Such controls typically include, for example, icons, texts, buttons, input boxes, menus, drop-down lists, sliders, scroll bars, and bars. Some users find it useful to attend a course or seminar which teaches users how to use one or more applications. Some users require assistance from other users in order to be able to complete particular tasks using an application. Some applications include a "help" function, in which the user may utilize an index of pre-defined topics, or a search based on user-entered keywords, in order to retrieve pre-defined textual or visual descriptions which may assist the user in finding how a particular task is performed. Some applications are associated with a user's manual, or with a batch of Frequently Asked Questions (FAQ), which may further guide the user on how to perform particular tasks.

One prior art solution suggests a simulation, demonstrating or teaching a procedure used in a software program, based on a script of commands that are executed by a script engine. The script engine interprets the commands using libraries of functions that interact with the software program and the operating system of the computer on which the script engine is executed. The commands are control actions implemented within the software to emulate the actions of an expert carrying out the procedure being simulated. For example, the commands cause a mouse cursor to select a specified item from a drop down menu list as if the cursor were being controlled by the expert in an appropriate display screen of the software program. A plurality of images are thus created showing the cursor in different positions as it moves on the display screen. Text is entered in response to the commands, causing images showing each letter (or phrase) being entered, as if by the expert. The images are converted into an animation that can be run from within the software program or as a stand alone demo. Author selected text and numeric tokens within the commands are added to a translation table and associated with text in a different language or other numeric values to create simulations in other languages. Modification of a simulation in response to changes in the software program or to correct errors is readily accomplished by editing the commands in the script and rerunning the creation of the images and animation file

INVENTOR DISCLOSURE:

The guidance is performed by executing a guidance script capable of actually operating the application by way of activating one or more controls of the application, such as by imitating cursor moves, cursor clicks or double-clicks, and keyboard strokes in lieu of the user. Guidance is provided in a manner independent of various display properties of the application or the computerized device used ("*display-independent*"). Since an application may be run with different display properties, such as window size, resolution, color, fonts, or themes, a conventional guidance script which was recorded with one set of display properties may encounter difficulties operating an application which uses a different set of display properties. This may happen, for example, because a certain GUI control operable by the conventional guidance script may appear in a new, unrecognized location or form in the actual application on which the script is finally run. Accordingly, the GUI of the application is analyzed by analyzing a screenshot, for matching the required control present in the GUI with a corresponding control of a pre-recorded guidance script. Then, a guidance script execution engine is able to execute the script and operate the necessary control. Thereafter, a guidance script is created by recording a sequence of operations in an application, by capturing and analyzing screenshots of the application's GUI. In the analysis, the recording engine inspects an area surrounding a determined cursor position, such as a position where a GUI control was activated by the user performing a sequence of operations. The activated GUI control is then identified, and an image of it or metadata associated with it is then assigned to the respective step of the operation sequence.

Display-Independent Computerized Guidance

A screenshot of a GUI of a computerized application is captured, in response to a user request for guidance, or activity of the user tracked, to automatically determine if the user is experiencing difficulties with the operation of the application, in order to proactively provide the user with guidance as to the operation in which difficulties were encountered. GUI may include a user interface of what is often referred to as a "*console application*" (also "command line", "command prompt", "text terminal", or "terminal"), which is a software application having a user interface composed mainly of text, and sometimes with the addition of some simple, minimalistic graphic elements or symbols. Due to their lack of extensive graphic elements, their GUI is often referred to simply as a "user interface" (*UI*).

Guidance script includes an executable operation sequence for operating the application. The operation sequence includes steps, each step optionally containing an action pertaining to a GUI control (sometimes referred to simply as a "*control*") of the application. The analysis of a screenshot includes running an image recognition algorithm adapted to identify graphical or textual characteristics of a control in the screenshot, to enable a matching of

the control with a corresponding control of a step of a guidance script. Textual characteristics may be identified using Optical Character Recognition (OCR).

Following the identification of a control in screenshot, it may be matched with a control of guidance script, to enable a guidance script execution engine to execute the guidance script and operate the actual control in the application. There may exist multiple controls in the application's GUI that look essentially the same. In order to correctly match one of these GUI controls with the correct control of the guidance script, it may be necessary to analyze also the surroundings of the control (such as its adjacent controls or adjacent graphical elements), both in the screenshot and in the guidance script, so as to increase the chances of a correct matching.

In case the matching is unsuccessful, namely—no corresponding controls of the screenshot and the guidance script are found, a screenshot analysis engine is utilized for automatically activating one or more scroll bars, for scrolling to a previously-invisible region of the computerized application. A screenshot analysis engine automatically scrolls to previously-invisible regions of text area for searching for a GUI control which may be a match for a control of the guidance script. After the scrolling uncovers a new region, a screenshot of the GUI is captured again, and analysis and matching is repeated for the new region.

The execution may be performed in a semi-automatic mode in which, following an automatic activation of a control, a help text is displayed, asking the user to perform a manual operation. Guidance script execution engine may be adapted to automatically detect when the user finished performing the manual operation. The user may also be presented with a button on which he clicks to indicate to guidance script execution engine that he completed the manual operation and wishes to resume execution of the guidance script.

Recording a Display-Independent Computerized Guidance Script

A guidance script to be later run on the same or a different computerized device is recorded by a software developer or a software developing company and distributed to users for running on their computerized devices. The recording includes a screenshot capture, a screenshot analysis, or an operation sequence recorder. An operation sequence in the computerized application is manually performed by a user, each step optionally containing an action pertaining to a GUI control of the application. These actions may be performed either manually (by a user) or automatically, using software adapted to execute actions in the computerized application. Usage of such software enables faster, more accurate or more efficient recording of an operation sequence, without the need for substantial human intervention. During the recording, one or more controls may be assigned to their respective steps. A guidance script can include a recorded operation sequence and a control assigned to each step.

A screenshot of a GUI of a computerized application can be captured, responsive to a cursor click or a keyboard input by the user. A position at which a cursor was at when performing the click, may be identified in the screenshot. If a keyboard stroke is the manual operation by the user, a position of a control activated by the stroke may be detected. An area surrounding the determined cursor position is analyzed, such as by screenshot analysis enabling identification of a control on which the cursor click was performed or which the keyboard stroke activated. The identified control or its surrounding area may be saved as an image and assigned to the step of operation sequence and optionally stored together with the guidance script.

It is possible thereby to produce a guidance script which is display-independent, namely - due to the storing of an image together with each step, it is later possible to execute the guidance script on a computerized device having display properties from the computerized device of which the recording of method is done. That is, the stored image may be matched, during the execution, with an identified control of the GUI of the computerized application on which the guidance script is executed.

Display-Independent Recognition of a GUI Control

A general-purpose image recognition is used by a control recognizer for recognizing and identifying a GUI control in a screenshot of a GUI of a computerized application, which may or may not be tied to execution or recording of a guidance script, to complement other computerized operations as desired. A position at which a cursor was at when performing the click, if such a click was indeed performed, may be determined by cursor position identifier. If a keyboard stroke was performed by the user, then a position of a control activated by the stroke may be detected. If actions of an optional block are performed, then an area surrounding the determined cursor position is analyzed, such as by a cursor position region analyzer. The analysis may enable identification of a control on which the cursor click was performed or which the keyboard stroke activated.

If actions of the optional block are not performed, then an area of screenshot up to its entirety may be analyzed, such as by cursor position region analyzer. The analysis may enable identification of one or more controls existing in the screenshot. Alternatively, coordinates in the screenshot may be received from a user, and the analysis may be performed on an area surrounding or at least including these coordinates. The identified control or its surrounding area may be stored as an image. The image of the control may then be used in another computerized process or application.

Display-independent scroll bar recognizer.

A scroll bar recognizer may be used for recognizing and identifying a scroll bar in a screenshot of a GUI of a computerized application. The use of a scroll bar recognizer may or may not be tied to execution or recording of a

guidance script. The scroll bar recognizer can include a screenshot capture engine and a scroll bar analyzer, each being a software component or module.

The screenshot is analyzed, in order to identify a scroll bar in a block. The analysis may be performed by scroll bar analyzer. The analysis may include a search for a control (namely, the scroll bar) which complies with the common characteristics of a scroll bar, namely - the existence of a thumb, an elongated trough and one or more arrows. Horizontal scroll bar, for example, can includes a *thumb* (sometimes referred to as a "*bar*"), which is optionally an element adapted to be dragged (using a cursor or a keyboard) along an elongated trough. The dragging of a thumb may cause "scrolling", namely, revelation of previously-invisible areas of text area. For example, if the thumb is dragged, a previously-invisible right area of text area is revealed. A width (or height, in case of a vertical scroll bar) of the thumb may be reflective of the degree of zooming applied. A thumb that completely fills a trough indicates that the entire document is being viewed. Alternatively, scrolling may be achieved by clicking on an arrow, of the scroll bar. Clicking on one of the arrows, for example, may reveal a previously-invisible area of text area and may move the thumb to the side.

The analysis may include a search for at least two elements of a scroll bar, namely, an arrow and a thumb. An arrow may be identified by searching for a continuous island of connected pixels that form a triangular shape. A thumb may be identified by searching in an axis parallel to the pointing direction of the identified arrow. Each element found in that axis may be analyzed, and the existence of a thumb may be determined if an essentially quadrangular island of connected pixels is found. An essentially empty space (represented by an island of connected pixels having the reverse color of the thumb) between the arrow and the thumb may resemble a trough. Alternatively, the two elements searched for may be two arrows of opposing directions, lying on the same axis, which is parallel to their pointing directions.

The identified scroll bar is analyzed, to extract at least one parameter of it in a block. The analysis of the scroll bar may include a determination of one or more of the following parameters: a location of the scroll bar, a size of the scroll bar, a location of a thumb of the scroll bar, a size of a thumb of the scroll bar, a size of an arrow of the scroll bar and a location of an arrow of the scroll bar. Each of these sizes may be denoted in pixels, horizontally or vertically. Each of these locations may be denoted as X-Y coordinates of the scroll bar, the thumb or the arrow in relation to the entirety of the GUI. Following the extraction of the above parameter(s), the scroll bar may be automatically operated for scrolling. For example, its thumb may be dragged or its arrows may be clicked for revealing a previously-invisible area of the GUI.