מיגון מערכות בפני טעויות אנוש

גירסא מקוצרת

אבי הראל
ארגולייט בע"מ

המונח 'טעויות אנוש' מתייחס אל פעולות של המפעיל, כאשר הפונקציה של המערכת בתגובה לפעולה שונה מהפונקציה לה המפעיל ציפה. טעויות המפעיל הן גורם עיקרי לירידת התפוקה של המערכת, ולחוסר שביעות רצון של המפעיל ואף לתאונות. מאמר זה סוקר שיטות למניעת נזקים כתוצאה מטעויות הפעלה.

בעיות קריטיות בתפעול מערכות נמצאות בתחום האפור בין הנדסת מערכת לבין הגורם האנושי. תאונות רבות שקרו בעבר הן תוצאה של התנהגות מפעיל שלא נצפתה מראש על ידי מתכנני המערכת. במקרים מסוימים, ההתנהגות הבלתי צפויה גורמת לכך שהמערכת נמצאת במצב חריג, ואופן התגובה שלה לארועים נוספים עלול להיות קטלני.

אחד המאפיינים העיקריים של מערכות אינטראקטיביות הוא מגוון רחב ביותר של טעויות אפשריות. זאת, בגלל הצורך לאפשר למפעיל שליטה במצבים בלתי צפויים. הטעויות קורות כאשר מניחים שהמפעיל יתנהג על פי אילוצים שאינם מוגדרים פורמלית, בסדר מסוים שמאפיין תרחיש הפעלה. אם המפעיל חורג מאילוצים אלו, אנו נוהגים לומר שהמפעיל טעה. מספר הצירופים של פעולות אפשריות, בכל המצבים האפשריים של המערכת, הוא בדרך כלל גבוה מאוד, ומחייב התייחסות סיסטמטית.

בדרך כלל, אנחנו מתכוונים לכך שהמפעיל הוא זה שעושה את הטעות, אבל למעשה, לטעות יש תמיד שני שותפים: המפתח והמשתמש. תפקיד המפתח הוא לשמור על המערכת בפני טעויות אפשריות באופן השימוש, ואילו המשתמש הוא הקורבן, במקרה שהמפתח נכשל בתפקידו.

באופן מסורתי, האחריות על הגדרת ממשקי ההפעלה של מערכות מוטלת על מומחים בתחום הנדסת גורמי אנוש. בדיסציפלינה זו מתייחסים לטעות על מאפייני פעילות המפעיל. מבחינים בשני סוגי טעות: האחד, כאשר המפעיל טעה בבחירת הפקד, והשני, כאשר הפקד שהמפעיל בחר הוא בהתאם לכוונתו, אבל תגובת המערכת אינה תואמת את ציפיותיו. גישה זו היא בעייתית, מכיוון שבעיות קריטיות בתפעול מערכות נמצאות בתחום האפור בין הנדסת מערכת לבין הגורם האנושי. בהקשר זה, המשימה העיקרית של מהנדסי המערכת היא לנהל את שלשת הדיסציפלינות, מהנדסי המערכת, מהנדסי התוכנה ומהנדסי גורמי אנוש. בנוסף, הם אחראים על נושא ניתוח תקלות, תהליכי איתור תקלות והגדרת דרכים למניעת הסלמה במקרים של תקלות.

הנחת העבודה בתכנון ממשק הפעלה צריכה להיות שהמפעיל יעשה כל טעות אפשרית. הטיפול בטעויות ובפעולות חריגות הוא מסובך, מחייב השקעת תשומת לב בפרטים רבים, ומאפשר הסלמה, תקלות וטעויות נוספות. לפיכך, הדרך המועדפת להגנה בפני טעויות היא על ידי מניעתן. המאמר מציג דרכים אפשריות להמנעות מטעויות מפעיל: על ידי אוטומציה, על ידי צמצום, על ידי פישוט, בעזרת תרחישים, בעזרת גורמי אנוש ועל ידי רגולציה של תהליכי הפיתוח ובקרת איכות המוצרים.

במקרים מסויימים, כאשר הסיכון של פעולה אוטומטית בלתי הולמת הינו גבוה, אין לנו ברירה אלא לאפשר למפעיל לחרוג מהפעילות השגרתית, ולטעות. זאת, כשמדובר בפעולות חיוניות של המפעיל, כגון, לצורך פתרון בעיות, כאשר חייבים לאפשר לו שליטה במערכת. מפתחי המערכת נדרשים לפרוש רשת בטחון למקרים של פעילות מפעיל שהיא בלתי צפויה. בכדי להבטיח שהמערכת תגן בפני טעויות המפעיל, מפרטי האינטראקציה צריכים לתאר את ההפעלה על פי תרחישי ההפעלה. הגדרת האינטראקציה כוללת אובייקטים, מצבים, פונקציות, מקרי-שימוש ופעילויות המתארות את התרחישים. את המודל האינטראקציה צריך ליצא מהמפרטים אל מערכת היעד, על מנת לאפשר איתור מקרים של חריגה של המפעיל מהפעילות המוגדרת על ידי התרחישים. את מפרטי האינטראקציה חשוב לפתח בעזרת תוכנה שמאפשרת עדכונים אמינים, תוך כדי בדיקת המשמעות של שינויים על רגישות המערכת לטעויות המפעיל.

הגירסא המלאה של המאמר:

http://www.ergolight-sw.com/CHI/Company/Articles/HumanErrorsIncose07-TooLong.doc

הבעיה של טעויות מפעיל

טעות בהקלדה. במקום טקסט, הוקלד קיצור לפעולה שמשנה את מצב הטקסט במסך. הפעולה הבאה גורמת למחיקת כל הטקסט מהמסך, ובהמשך, מהקובץ. עבודה של שעה הלכה לטמיון. צריך להקליד הכל מהתחלה (קישור למאמר ברשת).

טעות בהקלדה. בדרכו אל כוכב הלכת מאדים, הלווין הסובייטי פובוס 1 מקבל פקודת סגירה. כל ההשקעה הלכה לטמיון (קישור למאמר ברשת).

תקלה במערכת טלביזיה בכבלים. טעות בתפעול השלט של הממיר הדיגיטלי. רואים שלג. יש מספר סיבות אפשריות, יש מספר אופציות להתגבר על התקלה. לא כולם מסתדרים בעצמם. רבים נאלצים חדשות לבקרים לקרוא לשרות הלקוחות. חברת הכבלים מפסידה, בגלל הוצאות שירות הלקוחות ובגלל אובדן אמון הלקוחות (קישור למאמר ברשת).

תקלה במערכת הקבצים. מערכת ההפעלה מבקשת את אישור המפעיל לפעולת תיקון. המפעיל מאשר. המערכת מוחקת את כל הקבצים מהדיסק. המפעיל מתקשר אל שירות הלקוחות והם מסבירים לו בנימוס שהוא היה אמור לעשות גיבוי לכל הקבצים לפני שאישר את פעולת התיקון.

תקלה במכונית. צינור בלוי גורם לאיבוד מי צינון. נורית התרעה דולקת, אבל המשמעות לא ברורה, לא תמיד, לא לכל אחד, לא בכל מצב. הלך מנוע.

יש אנשים שסוגדים למוצרים טכנולוגיים – מכוניות יוקרה, מצלמות וידאו, טלפונים סלולריים, מערכות וידאו וסטריאו ביתיות – מקדישים להם יותר מאשר לבני משפחתם. ויש אנשים שמנצלים אותה, את הטכנולוגיה, בלי שום סנטימנטים, כאמצעי גרידא להשגת מטרות שונות, כגון הגעה למקום מסוים, יצירת קשר עם אנשים אחרים, הפקת מזכרות, הנאה ממוסיקה, ועוד. אבל כולם, בלי יוצא מהכלל, סובלים ממנה מדי פעם. לעתים די קרובות, כשהמוצר הטכנולוגי לא ממלא אחר צפיותינו, כשאינו משתף פעולה, כשהוא מעניש אותנו על כך שאנחנו פועלים שלא על פי הוראות היצרן : לפעמים בקנס כספי – תשלום לטכנאי על מנת שיחזיר מצב לקדמותו, לפעמים אנחנו רק מבזבזים את זמננו בנסיונות של "איך לעזאזל מזיזים בטלפון הזה את השעה לשעון קיץ?", ותמיד משלמים בעצבים שלנו. בבושה שאנחנו כל כך מטומטמים, שהצלחנו לקלקל את המכשיר המעולה הזה, או בתסכול שבזבזנו עליו כל כך הרבה זמן, והבן הקטן סידר את זה בתוך שניה, ועכשיו זה סוף-סוף עובד כמו שצריך ... עד הפעם הבאה.

טעויות קטלניות

הרוח טורקת את הדלת של מכון לבדיקת מי הירדן. לדלת ידית נשלפת, והידית בחוץ. בתוך המכון, עובד המכון שהגיע על מנת לבדוק את המים. המכון נמצא במקום מבודד ולכלוא בפנים אין קשר עם הסביבה. במכון סבורים שהוא נמצא בחופשה. בנס, הוא חולץ כשעודו בחיים (קישור לרפרנס ברשת).

שנת 1979, תקלה בכור גרעיני. האינדיקציות למקור התקלה בלתי ברורות ובלתי מובנות לצוות התפעול. בנס נמנע אסון סביבתי (Walker, 2004).

שנת 1967, מיכלית נפט בדרך לנמל בדרום ווילס. מישהו בצוות המיכלית הזיז את בקרת ההיגוי בטעות למצב 'נייטרל'. המיכלית מתקרבת לשרטון. הקברניט לא שולט בהיגוי. המיכלית מתרסקת. האסון האקולוגי הגדול ביותר של המאה שעברה (קישור למאמר ב-Wiki) .

שנת 2001, טעות במינון תרופה במיון ילדים. הילדה, בת שלש, במצב קריטי. ההתרעה הקולית נותקה, כי בעבר כבר היו התרעות שווא רבות. הפעוטה לא טופלה בזמן ויצאה מבית החולים במצב של שיתוק כללי (קישור למאמר ברשת).

שנת 1988, טעויות תכנות בבקרת הטיסה של מטוס נוסעים איירבאס 320. במפגן אוירי, בטיסה 296 של חברת אייר פראנס. המטוס אינו מגיב לפקודת נסיקה. המטוס מתרסק (קישור למאמר ב-Wiki) .

השנים 87-1985, המפעילה של מערכת רדיותרפיה פועלת מהר מדי, המערכת מקרינה בעוצמות גבוהות בסדרי גודל מהמתוכנן. ששה נפגעים, אחד מהם מת מהקרינה. (קישור למאמר ב-Wiki)

שנת 1990, תרגיל סיוע לגייסות בצאלים. קציין הקישור הארטילרי טועה במילת הקוד של בקרת האש. קציני בקרת התרגיל עסוקים במשימות אחרות. מפקד סוללת התותחנים אינו יודע את מיקום הגייסות. חמישה הרוגים ועשרה פצועים (קישור למאמר ב-Wiki)

האם ניתן היה למנוע את הטעויות הללו? על מי הוטלה האחריות? מהם הלקחים שהופקו? האם הטעות הבאה נמנעה? תשובה לשאלות הללו עבור חלק מהדוגמאות לעיל, ודוגמאות אחרות, ניתן למצוא אצל Casey (1998).

טעויות מפעיל

המונח 'טעויות מפעיל' הוא רב משמעי, והוא מטעה. בדרך כלל, הוא מתייחס למצב שהוא בלתי צפוי מבחינת המפעיל, שהוא תוצאה מפעילות קודמת שלו. הפעילות הקודמת יכולה להיות בכוונה או בטעות.

לצורך הדוגמא, נניח שכפתור בקרת ההתרעה הקולית במוניטור בבית חולים עבר למצב השתקה. הסיבה לכך יכולה להיות בטעות, למשל, בגלל שמישהו שעבר ליד המוניטור, נגע בכפתור מבלי משים, או בגלל שילד בביקור חולים לחץ עליו מתוך סקרנות. הסיבה יכולה להיות גם במכוון, למשל, לצורך תחזוקה, או בשעת ביקור רופאים. במקרים שונים, תגובת המערכת צריכה להיות שונה. במקרה של השתקה בטעות, היינו רוצים שהערכת תתריע על כך. במקרה של השתקה במתכוון, היינו רוצים שהמערכת לא תטריד את המפעילים בהתרעות שווא. במקרה של ביקור רופאים, היינו רוצים שהמערכת תתריע כעבור זמן מוקצב מראש, שנקבע על פי הערכה של משך הביקור. המקרה של טעות מתייחס למצבים בהם המערכת אינה מגיבה בהתאם לציפיות המפעיל.

מהם התנאים המאפשרים טעויות מפעיל?

טעויות מפעיל נפוצות במערכות בהן ממשקי ההפעלה מאפשרים לו שליטה מוחלטת, להפעיל כל פונקציה, בכל מצב. לדוגמא, אם המערכת מאפשרת למפעיל להתניע, כאשר המנוע כבר מופעל, הוא עלול לא לשים לב לכך שהמנוע מופעל, לנסות להניע ובמצבים מסויימים, לגרום בכך נזק למתנע. להמחשה, השלט-רחוק של הממיר הדיגיטלי מוצג בתרשים להלן:

תרשים1 - כפל משמעות הפקדים בשלט-רחוק של הממיר הדיגיטלי

בשלט-רחוק של הממיר הדיגיטלי ניתן למצוא 30-20 כפתורים, והמפעיל יכול ללחוץ על כל אחד מהם בכל רגע נתון. בפועל, המפתחים מניחים שהמפעיל יתנהג על פי אילוצים שאינם מוגדרים פורמלית, בסדר מסוים שמאפיין תרחיש הפעלה. אם המפעיל חורג מאילוצים אלו, אנו נוהגים לומר שהמפעיל טעה. בדוגמת השלט-רחוק, אם המפעיל מבקש להדליק את הטלביזיה באמצעות השלט, אנו מצפים ממנו שיעביר את השלט למוד טלביזיה לפני שהוא לוחץ על מקש הכיבוי-הדלקה. אבל, המערכת מאפשרת לו ללחוץ על המקש כיבוי-הדלקה

גם מבלי שהעביר את השלט למוד טלביזיה. במקרה כזה, השלט מכבה את הממיר, ולארוע כזה אנו קוראים ״טעות מפעיל״.

עקרון ודאות הכשל

עקרון ודאות הכשל לגבי מערכות אינטראקטיביות הוא שכל טעות אפשרית של המפעיל – תתרחש ביום מן הימים. כללית, טעויות מפעיל נשמעות לחוקי מרפי המוכרים בתחום של הנדסת מערכת: אם המערכת מאפשרת לו לטעות – במוקדם או במאוחר, הוא יטעה. המפעיל יפול בכל פח שהמערכת טומנת לו. זאת, משלשה טעמים: א. לצורך לימוד על ידי ניסוי וטעיה, ב. בתפעול שוטף: המפעיל עלול להפעיל פונקציות שונות שלא במתכוון, ו-ג. בפתרון בעיות: המפעיל יכול להפעיל פונקציה מסויימת במתכוון, למרות שהמערכת אינה תומכת בפונקציה הספציפית הזו במצב הספציפי, או למרות שאינו מבין את מלוא המשמעות של הפעלת הפונקציה במצב החריג.

סוגי טעויות מפעיל

טעות מפעיל, כאמור לעיל, פירושה תגובת מערכת לפעולת מפעיל, שאינה תואמת את ציפיות המפעיל. ניתן לסווג את טעויות המפעיל במספר אופנים. אחד הסיווגים המוקדמים הוא לפי טעויות פסיכומוטוריות לעומת טעויות החלטה. בין הטעויות הפסיכומוטוריות כוללים החמצת פקד, לחיצה כפולה, החטאה, ואינרציה בחזרות על סדרה של פקדים. בין טעויות ההחלטה כוללים בעיות בזיהוי המצב וקשיים בזכירת פרטים שאינם מוצגים בפני המפעיל (Norman, 1983). אופן הסיווג של טעויות מפעיל במאמר זה דומה, והוא כולל שני סוגים: האחד, כאשר המפעיל טעה בבחירת הפקד, והשני, כאשר הפקד שהמפעיל בחר הוא בהתאם לכוונתו, אבל תגובת המערכת אינה תואמת את ציפיותיו.

טעויות בבחירת הפקד יכולות לנבוע משני מקורות: טעות בכוונה, או טעות שלא במתכוון. טעות בכוונה יכולה לקרות כתוצאה מאי ידיעת הפונקציה שהפקד מפעיל, או מתוך הרגל מהפעלת מערכת אחרת, או מבלבול רגעי, למשל, במצבי לחץ. לדוגמא, אם המשתמש במערכת פיתוח רגיל להשתמש במקש הקיצור Ctrl-R להחלפת טקסט, והוא מבקש להחליף טקסט במסמך אופיס, הוא עלול להשתמש בקיצור זה במקום הקיצור Ctrl-H, ולשנות את אופן ישור הטקסט, לשוליים הימניים.

טעות שלא במתכוון, שנקראת גם טעות פסיכו-מוטורית, נובעת מבעית תיאום במערכת הבקרה הביולוגית של המפעיל. דוגמאות של טעויות פסיכו-מוטוריות כוללות החמצה (פגיעה בפקד סמוך לפקד הרצוי), כגון החלקת העכבר בבחירה מתפריט ולחיצה כפולה. לדוגמא, אם במקום ללחוץ על המקש Shift המשתמש בתוכנת מחשב לוחץ על המקש Ctrl הסמוך לו, ומפעיל פונקציה שלא במתכוון.

טעויות בתגובתיות המערכת יכולות לנבוע משני מקורות: מאי הבנת האופן בו המערכת צריכה להגיב לפקד, או משינוי במצב המערכת, שהמפעיל אינו מודע לו. טעות שמקורה באי הבנת אופן תגובת המערכת היא אופיינית למצבים חריגים. טעות שמקורה בשינוי במצב המערכת הינה אופיינית לפקדים בעלי כפל משמעות. למשל, מקשי הספרות בשלט-רחוק של הממיר הדיגיטלי הם בעלי כפל משמעות, והמשתמש עלול להחליף בטעות את ערוץ הטלביזיה במקום את ערוץ הממיר. במערכות ממוחשבות רבות, כגון ישומי חלונות רבים, כפל המשמעות נשלט על ידי פרמטרים שערכם נקבע ברשימה של ׳אופציות׳.

אחריות המפתחים

מפתחי מערכות רואים זאת כחלק בלתי נפרד מתפקידם לבחון את נושא האמינות של רכיבים אלקטרוניים, ולנקוט בצעדים להבטיח את אמינות המערכת גם במקרים של תקלות ברכיבים. לא כך הדבר כשמדובר באמינות המפעיל. מפתחי מערכות אינם מוכנים לקחת אחריות על טעויות של המפעיל, ולעתים מנצלים את טעויות המפעיל להצדקת מחדלים בתכנון.

בדרך כלל, אנחנו מתכוונים לכך שהמפעיל או המשתמש הם אילו שעושים את הטעות, אבל למעשה, לטעות יש תמיד שני שותפים: המפתח והמשתמש. תפקיד המפתח הוא לשמור על המערכת בפני טעויות אפשריות באופן השימוש, ואילו המשתמש הוא הקורבן, במקרה שהמפתח נכשל בתפקידו.

באופן טיפוסי, מפתחי מערכות נוטים להכחיש את עצם קיום הבעיה. ההליך המקובל של הכחשת בעיה הוא על ידי בחינה עצמית: אם אני מסתדר עם הפעלת המערכת, אין שום סיבה מדוע אחרים לא יוכלו להסתדר עם זה. באופן תגובה זה, המפתחים מתעלמים מקבוצה חשובה של משתמשים, אנשי עסקים, אנשי אקדמיה, מנהלים ומפקדים בכירים, שאין להם עודפי זמן, לבזבז על לימוד דרך החשיבה של מפתחי המערכת.

במקרים של טעות שמסתיימת בתאונה, אין אפשרות להתכחש לקיום הבעיה, ולכן במקרה זה תגובת המפתחים הטבעית היא להטיל את האשמה על המפעיל. במקרים בהם חוקרי התאונה מתקשים לפענח את מנגנון הכשל, הדרך המקובלת לחפות על האין-אונות שלהם היא על ידי הענשת הקורבן, דהיינו, המפעיל שנפל בפח שהטמינו לו, בחוסר תשומת לב, המפתחים. כך, למשל, קברניט המיכלית שגרמה לאסון האקולוגי בשנת 1967 פוטר מעבודתו, והושעה מכל פעילות ימית (קישור למאמר ב-Wiki). להצדקת האשמת הקורבן, בדרך כלל נוהגים לחקור את הארוע באופן יסודי ולמצוא מספר נקודות בהן הקורבן לכאורה התנהג שלא כראוי. כך, למשל, קברניט מטוס האיירבס 320 בטיסה 296 של אייר פראנס, הואשם בשורה של נושאים שאינם קשורים כלל לסיבת התאונה, כגון, שהצוות לא התכונן לתמרון, בחוסר תיאום בין אנשי הצוות, בתמרון שלא על פי התכנון, ברצף ארועים מהיר מדי, בזחיחות דעת, ביהירות, באוירת החג ובהשפעת הדיילות שביקרו בתא הטייס. זאת, על מנת לאפשר את הרשעתו באשמת הריגה, ולדון אותו למאסר (קישור למאמר ב-Wiki). כך גם, בין השאר, קציני צה"ל שהיו מעורבים בתאונת האימונים בצאלים, הורשעו בדין והוכנסו לכלא, באמתלות שאינן קשורות לפליטת הפה של קצין הקישור (קישור למאמר ב-Wiki).

הבעיה בגישה של הענשת קורבן הטעות היא בכך שהיא מסיחה את הדעת מהגורמים המבניים לתאונה. במקום לחקור את התאונה, לנתח ולהבין את הסיבות לה, משליכים כמה אנשים לכלא. בגישה זו, החוקרים לא פעלו למנוע בעיות דומות בעתיד (Norman, 1990). התוצאה היא שהתאונה חוזרת על עצמה. כך, 19 חודשים לאחר התאונה של מטוס האיירבאס 320 במפגן האוירי, התרסק מטוס נוסעים מדגם זה בבאנגלור שבהודו ו-94 אנשים מצאו את מותם, בגלל אותה תקלה. חקירה רצינית של הגורמים לתאונה הראשונה החלה רק אחרי התאונה השניה. כך גם, שנתיים לאחר תאונת צאלים א', קרתה תאונת צאלים ב', בגלל כשל חוזר של מניעת ירי על כוחותינו. נהלי הבטיחות של הסיוע הארטילרי, שלא מנעו את תאונת צאלים א' נבחנו מחדש רק לאחר תאונת צאלים ב'.

סמנטיקה של טעויות אנוש

נצא מנקודת הנחה שמפתחי מערכות עושים כמיטב יכולתם על מנת למנוע את טעויות המפעיל. למרות זאת, המפעיל "מצליח" להפתיע אותנו, כאשר הוא פועל שלא על פי הציפיות שלנו. לעתים, מקור ההפתעה הוא בטעות, ולעתים, ההפתעה היא כשהמפעיל משוכנע שהמערכת פועלת באופן מסוים, או במצב מסוים, אבל המערכת אינה תומכת באופן פעולה זה, או במצב פעולה זה. במקרים אילו, אין לייחס את הבעיה למפעיל, אלא למפתח. הבעיה היא של חוסר תיאום ציפיות בין המפעיל לבין המפתח. האתגר של מפתח המערכת הוא לצפות מראש את כל הטעויות האפשריות של המפעיל, ולמצוא להן מענה. אם המפתח נכשל בכך, זה בגלל שהוא טעה בחיזוי אופן התנהגות המפעיל, ולכן מדובר ב"טעות מפתח". הסמנטיקה של "טעות מפעיל" מטילה את האחריות למצב של חוסר תיאום על המפעיל, ומשחררת למעשה את המפתח מאחריות למניעת ההפתעות. על מנת להבטיח התייחסות נאותה של המפתחים למצבים של חוסר תיאום, במקום להשתמש במונחים של "טעות", עדיף שנשתמש במונחים של "הפתעה" או "פעולה חריגה", או "פעולה בלתי צפויה של המפעיל".

מגבלת התכנון על פי מקרי-שימוש (use cases)

טעות נפוצה בקרב מפתחים היא להניח שהמשתמש יכול ומעוניין ללמוד להכיר את המוצר באותה רמת פירוט שהם מכירים אותו. טעות זו קשורה למתודולוגית אפיון מערכות בעזרת 'מקרי שימוש' (use cases), המקובלת בתהליכי אפיון תוכנה בשימוש בשפת המודלים UML. מתודולוגיה זו מאפשרת להגדיר היטב מה המערכת מסוגלת לעשות, אבל היא אינה מתארת איך המפעיל אמור להשתמש ב'מקרי השימוש'. הדרך ש-UML מציע לתאר אופני שימוש היא בעזרת תרחישים (Jacobson, 1987). הבעיה קיימת כאשר תיאור התרחישים אינו מבוסס על חקירה יסודית של פעילות המשתמש. במקרים אלו, ממשק ההפעלה מספק גישה לפונקציות המערכת ברמה של מקרי-שימוש, שמספקת למפעיל הזדמנויות רבות לטעות. לדוגמא, בגירסאות של תוכנות אופיס של מיקרוסופט התומכות בעברית (ובערבית) ניתן למצוא אופציות לשליטה בכיוון תנועת הסמן, כיוון ההצגה של הטקסט במסך, ישור הפיסקה לימין או לשמאל, באופן הצגת הסמן ובסוג הפונטים שיוזן למסמך. כמו כן, אופן ההתנהגות של אותיות Caps, של מספרים ושל סימני פיסוק תלוי בקומבינציה של האופציות

השונות. בגלל שהממשק אינו מוגדר על פי תרחישים, המשתמשים בתוכנות אופיס מבזבזים זמן רב על נסיונות לשלוט בתנועת הסמן. הגדרה על פי תרחישים כגון: עריכת עברית, אנגלית, Caps , מספרים וסימני פיסוק, היתה חוסכת זמן רב של המשתמשים, כאשר מה שנדרש זה תרגום אוטומטי של מספר מצומצם של תרחישים לקומבינציות המתאימות של אופציות השליטה בטקסט.

מגבלת זכרון המפעיל

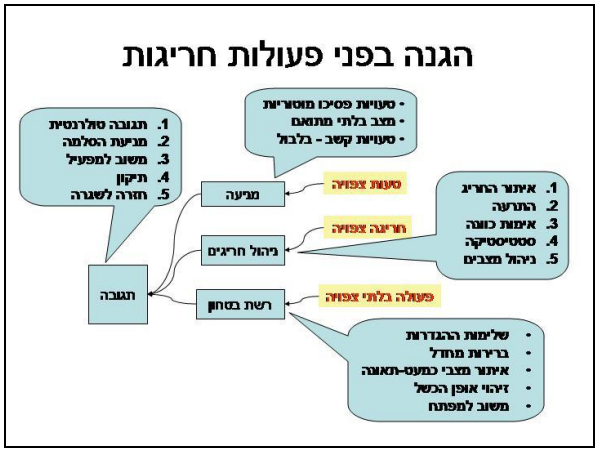
טעות נוספת שהיא נפוצה בקרב מפתחים היא להניח שהמפעיל זוכר באותו אופן שהמחשב זוכר. סוג טעות זה מודגם במקרה של נהיגה ברכב עם בקרת מהירות מופעלת, במהירות שהיא גבוהה ממהירות ההצבה. הנהג עלול לשכוח שהבקרה מופעלת, ולמצוא עצמו נוהג במהירות שהיא גבוהה מדי בתנאי הכביש (Andre & Degani, 1997). ההנחה של מפתחי בקרת המהירות היתה שהנהג זוכר את מצב פעולה של בקרת המהירות. הנחה זו היתה מוצדקת, אבל לא לאורך זמן. לאחר נהיגה של מספר דקות, המידע לגבי מצב הפקד נמחק מהזכרון של הנהג, ושיטת הנהיגה התחלפה, על פי הרגלים קודמים, המבוססים על המידע שאגור בזכרון לטווח ארוך.

אפשרויות למיגון בפני טעויות מפעיל

במקרים מסויימים, כאשר הסיכון של פעולה אוטומטית בלתי הולמת הינו גבוה, אין לנו ברירה אלא לאפשר למפעיל לחרוג מהפעילות השגרתית, ולטעות. זאת, כשמדובר בפעולות חיוניות של המפעיל, כגון, לצורך פתרון בעיות, כאשר חייבים לאפשר לו שליטה במערכת.

כאשר המפעיל טועה בפעולה, המערכת אינה יכולה לדעת אם הפעולה היתה בכוונה תחילה, או בטעות. אחת הדרכים לברר זאת היא לשאול את המפעיל, לוודא שהוא התכוון ברצינות, ולא בטעות. במקרים כאלו, המערכת חייבת להגיב נכון: ראשית, לברר מה היתה כוונת המשתמש בביצוע פעולה הבעייתית, האם היא היתה בטעות או שאולי במתכוון, לצורך פתרון בעיה שחורגת מהשגרה, ולהגיב בהתאם. יש להמנע במקרים אלו מעודף שאלות, ולהתחשב במגבלת קיבולת העומס המנטאלי שלו. אם המפעיל נדרש לבדיקות רבות, הוא עלול להחמיץ את הבדיקות החשובות. לדוגמא, בקרת התרגיל בתאונת צאלים א' נכשלה בזיהוי האיום של ירי על המסתייע בין השאר מפני שהוראות הבטיחות נבנו טלאי על טלאי כלקחים מתאונות קודמות, וכללו בדיקות חובה שהעמיסו את בקרת התרגיל יתר על המידה.

מכיוון שהמפעיל עלול לטעות בכל שלב שהוא, המשמעות של פתרון מסוג זה הוא שהמערכת תעמיס את המפעיל בשאלות מסוג "האם אתה בטוח?", כאשר במרבית המקרים התשובה היא חיובית. ברור שפתרון זה אינו סביר. לדוגמא, אין זה סביר שישומי אופיס יטרידו את המפעיל בכל לחיצת מקש בשאלה האם הוא בטוח שזה מה שהוא התכוון. לפיכך, אנו נאלצים לוותר מראש על כך שהמערכת תדע על כל טעויות המפעיל. משמעות הדבר היא שאת עיקר המאמץ בהגנה בפני טעויות מפעיל צריך להשקיע במניעה, ולהשאיר מעט ככל האפשר לתפקידים של איתור וניהול מצבי טעות. התרשים הבא מציג את האפשרויות העומדות בפנינו כשאנחנו מבקשים להגן על המערכת במצבים של פעילות חריגה של המפעיל:

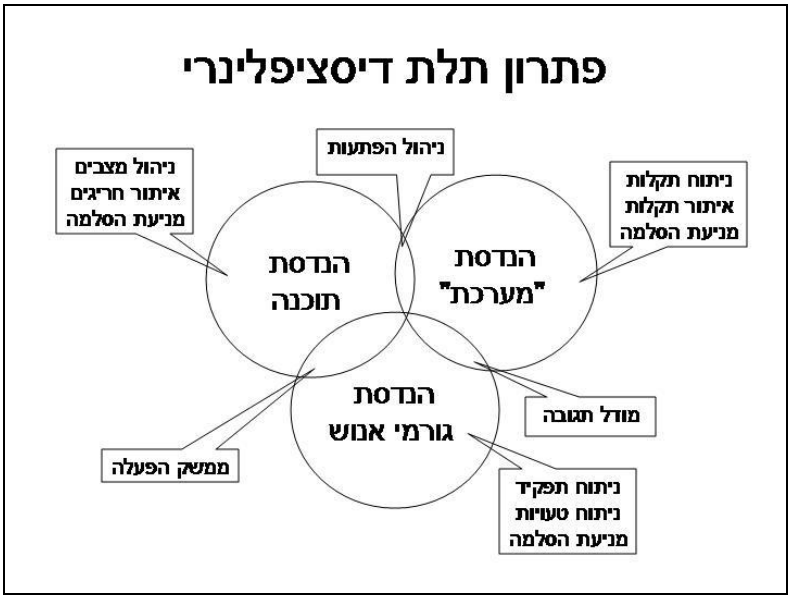


תרשים2 - אפשרויות ההגנה בפני פעילות חריגה של המפעיל

התרשים מבחין בין שני סוגים של פעולות מפעיל: פעולה בטעות ופעולה חריגה. את הפעולות שמוגדרות כטעות, ניתן לעתים למנוע. בנוסף, התרשים מבחין בין שני סוגים של פעולה חריגה. חריגה צפויה ופעולה בלתי צפויה. פעולה חריגה שהיא צפויה, ניתן לנהל, להתעלם ממנה, לדווח עליה למפעיל, או להיענות לה, בהתאם לצורך, כפי שהוא מוגדר במפרטים. פעולה בלתי צפויה אין אפשרות לנהל. ניתן לאתר אותה, ניתן להתעלם ממנה, ניתן לדווח עליה למפעיל ולמפתח, אבל אין אפשרות להיענות לה. בכל מקרה, צריך לדאוג לכך שהתגובה תהיה הולמת: טולרנטית, תמנע הסלמה, תאפשר תיקון המצב וחזרה להפעלה שגרתית.

שיתוף וחלוקת אחריות בין המפתחים

בהקשר של טעויות המפעיל, המשימה העיקרית של מהנדסי המערכת היא לנהל את שלשת הדיסציפלינות, מהנדסי המערכת, מהנדסי התוכנה ומהנדסי גורמי אנוש. בנוסף, הם אחראים על נושא ניתוח תקלות, תהליכי איתור תקלות והגדרת דרכים למניעת הסלמה במקרים של תקלות. התפקיד של מהנדסי התוכנה הוא לנהל את מצבי המערכת, לאתר מצבים חריגים וארועים חריגים ולמנוע הסלמה במצבים אלו. התפקיד של מהנדסי גורמי האנוש הוא לנתח את תפקידי המפעיל, להגדיר תרחישי הפעלה, לנתח את הטעויות האפשריות ולמנוע הסלמה במצבים של טעות. בנוסף, להגנה בפני טעויות מפעיל מתחייב שיתוף פעולה של בין גורמי ההנדסה השונים, כפי שהתרשים הבא ממחיש:



תרשים3 - שילוב הנדסות לצורך מיגון בפני טעויות המפעיל

מודל התגובה לפעולות המפעיל מוגדר בשיתוף בין מהנדסי המערכת לבין מהנדסי גורמי אנוש. מודל ניהול ההפתעות מוגדר בשיתוף בין מהנדסי המערכת לבין מהנדסי התוכנה. הארגון הפונקציונלי של ממשק ההפעלה מוגדר בשיתוף בין מהנדסי גורמי האנוש לבין מהנדסי התוכנה. עיצוב ממשק ההפעלה נעשה בשיתוף מהנדסי גורמי האנוש עם המעצבים.

מיגון על ידי המנעות

הדרך הנוחה ביותר למיגון בפני טעויות היא על ידי המנעות ממצבים בהם הטעות מתאפשרת. זאת, מכיוון שהטיפול בטעויות ובפעולות חריגות הוא מסובך, מחייב השקעת תשומת לב בפרטים רבים, ומאפשר הסלמה.

המנעות על ידי אוטומציה

הדרך הראשונה למנוע טעויות מפעיל היא על ידי אוטומציה. אוטומציה מאפשרת המנעות מטעויות מסויימות. לדוגמא, גיר אוטומטי במכונית מונע מצבים רבים של שילוב הילוך שאינו מתאים למהירות הנסיעה.

אוטומציה מיודעת בראש וראשונה לחסוך זמן ואנרגיה של המפעיל, אבל חסכון זה יכול לעלות ביוקר. למשל, ספקי המכשירים הסלולריים בארה״ב מחוייבים לספק אותם עם מקש קיצור למוקד החירום 911. מקש הקיצור גורם להתרעות שווא רבות, שגורם למוד החירום להתעלם מקריאות רבות. במספר מקרים, התעלמות זו עלתה בחיי אדם.

המנעות על ידי צמצום

הדרך השניה למנוע טעויות מפעיל היא על ידי צמצום. נזיר אנגלי בשם ויליאם מאוקאם (קישור למאמר ב-Wiki) הציע עוד במאה ה-14 את עקרון הצמצום בתחום של תיאוריות מדעיות. עקרון זה ניתן לישום גם עבור הנדסת מערכות, ומשמעו שעלינו להסיר פרטים מיותרים, שניתן להסתדר בלעדיהם. סוג הפרטים הרלבנטי לנושא של טעויות מפעיל כוללים ישויות יסודיות, וקשרים בין הישויות. הישויות היסודיות כוללות: הפונקציות, האופציות והמצבים, הפקדים, התצוגה למפעיל.

המנעות על ידי פישוט

הדרך השלישית למנוע טעויות מפעיל היא על ידי פישוט הסיבוכיות התפעולית. הסיבוכיות התפעולית מוגדרת על ידי מספר הקשרים בין הישויות של ממשק ההפעלה (הפונקציות, האופציות והמצבים, הפקדים והתצוגות) אליהן המפעיל צריך להיות מודע בכדי להמנע מטעות. בדוגמא של המיכלית שעלתה על שרטון, הסיבוכיות התפעולית נבעה מהקשר שבין פונקצית ההיגוי לבין מצב ידית בקרת ההיגוי. בדוגמא של השלט-רחוק של הממיר הדיגיטלי, הסיבוכיות התפעולית נובעת מהקשר שבין פונקציות הכיבוי-הדלקה ובחירת הערוצים לבין מצבי השלט.

הסיבות העיקריות מדוע מעניקים לפקד משמעות כפולה הן: א. דמיון פונקציונלי, מתוך רצון להקל על ההתמצאות בתהליכי ההפעלה, ב. לחסוך בשטח של פאנל ההפעלה, ו-ג. מעודף רצון להקל בתהליך, לנחש מה תהיה כוונתו של המפעיל במצבים השונים, ולהתאים את הפונקציה לכוונה הצפויה. הבעיה היא שהמפעיל לא תמיד מזהה את המצב באותה רמה של דיוק שהמפתח תכנן או חזה, ולכן תגובת המערכת עלולה להפתיע את המפעיל. האתגר של המתכנן הוא לפשט את תהליכי ההפעלה לחלוטין, כלומר, לבטל את כל הקשרים בין ישויות ממשק ההפעלה – לאפס את הסיבוכיות התפעולית.

אחת הטעויות הנפוצות בעיצוב פאנל ההפעלה היא להתבסס על תרשימי זרימת מצבים statecharts. במערכת שמוגדרת בדרך זו, מעברי המצבים תלויים במצב, ולכן הם רגישים לטעויות מפעיל. על מנת להמנע מטעויות מצב, יש לעצב את פאנל ההפעלה על עקרון המיפוי הישיר מפונקציה של המפעיל לפונקציה של המערכת, תוך שימוש בפקדים שמעבירים מצב ללא תלות בתרחישים. המקש TV בשלט-רחוק מהווה דוגמא למימוש עקרון זה.

אחת הדרכים לפישוט ההפעלה היא על ידי הכפלת הפקדים שהם בעלי כפל משמעות, לפי מספר המשמעויות. בדוגמא של השלט, את המקש כיבוי-הדלקה ניתן להחליף בשני מקשי כיבוי-הדלקה: אחד עבור הטלביזיה והשני עבור הממיר. אם בשלב זה מוותרים על כך שהשלט ישמש להחלפת ערוצים בטלביזיה, אזי ניתן לבטל גם את מקשי המוד טלביזיה-ממיר, כך ששלשת המקשים מוחלפים בשניים, וחסכנו בכך גם בשטח פאנל השלט.

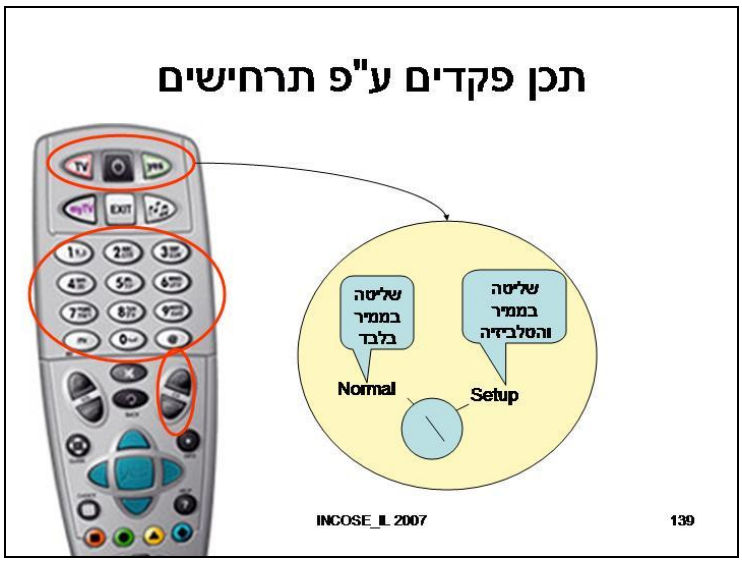
את הפקדים ניתן לשכפל במספר דרכים: שכפול פיזי בשכבות, ושכפול וירטואלי. שכפול וירטואלי אפשרי בעזרת מודים, דוגמת מודים של מתג בורר, או המודים של ה-Caps או ה-Alt במקלדת או בעזרת לחצן עזר, דוגמת השימוש במקשי Ctrl או Shift. המודים הזמניים, דוגמת Alt עדיפים למניעת טעויות, מכיוון שצריכת הקשב שלהם היא זמנית.

המנעות בעזרת תרחישים

הדרך הרביעית למנוע טעויות מפעיל היא על ידי הגדרת ממשק ההפעלה במונחים של תרחישים. התרחישים מאפשרים למפתחים לתכנן, ולעצב ולנסח את ממשק ההפעלה במונחים של המפעילים.

שפת המודלים האוניברסלית UML כוללת 'תחביר' שמאפשר להגדיר את האינטראקציה באמצעות תרחישים. מכיוון שהתרחישים מתארים את נקודת המבט של המפעיל, הם מאפשרים לאתר ולזהות אופני כשל מערכתיים שנובעים מטעויות-לכאורה. למשל, בדוגמא של השלט-רחוק של הממיר הדיגיטלי, בתרחיש של שימוש יומיומי, ניתן לזהות כשל של כיבוי הממיר במקום הטלביזיה, וכשל של החלפת ערוץ הטלביזיה במקום הממיר. בדוגמא של תאונת המיכלית, בתרחיש של שיוט, ניתן לזהות כשל של שיוט במצב 'ניוטרל'.

הדרך להמנע מהכשלים לעיל היא בשני שלבים: בשלב הראשון, מחלקים את ממשק ההפעלה לתת-ממשקים, על פי התרחישים, ומשכפלים את תהליך ההפעלה בהתאם. בשלב השני, מטפלים באופני הכשל ברמה של תת הממשק. את התרחישים ניתן להגדיר באופנים רבים, והם תלויים במידה רבה בסוג הישום. בדוגמה של המיכלית, התרחישים הרלבנטיים הם של שיוט בים ושל תחזוקה. לגבי השיוט בים, ניתן הגדיר שני תת-תרחישים, האחד של ניהוג אוטומטי והשני של ניהוג ידני. בתפעול השלט-רחוק של הממיר הדיגיטלי ניתן להגדיר שני תרחישים עיקריים: תרחיש התקנה, איתחול ופתרון בעיות, ותרחיש תפעול שוטף, יומיומי. דוגמא של פתרון המבוסס על תרחישים מוצגת בתרשים הבא:



תרשים4 – דוגמת פתרון בעית המודים עבור השלט-רחוק של הממיר הדיגיטלי

המנעות בעזרת גורמי אנוש

הדרך החמישית למנוע טעויות מפעיל היא על ידי התחשבות במאפיינים של הגורם האנושי. על בסיס מאפיינים אלו מגדירים כללים המסווגים על פי סוגי טעויות מפעיל: טעויות פסיכומוטוריות, טעויות בבחירת פקד, טעויות בהבנהת אופן התנהגות המערכת וטעויות מצב.

המנעות על ידי רגולציה

הדרך הששית למנוע טעויות מפעיל היא על ידי רגולציה. הכוונה היא לכללים שמנוסחים כהנחיות, חוקים, המלצות וכד', שמאפשרים לשפר את האיכות התפעולית של המערכת. מטרות הרגולציה קשורות לחסכון במשאבי פיתוח ובזמן פיתוח, על ידי המנעות מטעויות באפיון, בתיכון ובעיצוב ממשק ההפעלה.

ניתן לסווג את הכללים השונים לארבע קבוצות

- כללי זהב – אלו הם כללים שאינם מחייבים, שאינם מאורגנים במסגרת של מסמך תקן ומשמשים בגדר המלצה של מומחים בתחום. לדוגמא שמונת כללי הזהב של Shneiderman (2004)
- תקנים רשמיים – אלו הם כללים שהוכרו על ידי ועדות רשמיות, דוגמת מכון התקנים. סקירה של התקנים הבינלאומיים בנושא שימושיות ניתן למצוא אצל: Bevan (2001)

- תקנים דה-פקטו – אלו הם כללים שהוגדרו ונכתבו כהצעה לתקן רשמי או כתקן פנימי בארגון, זכו להכרה על ידי קהילת אנשי הפיתוח. דוגמא לכך זהו התקן LUCID של חברת היעוץ Cognetics
- הנחיות יצרנים – אלו הם המלצות של יצרני מערכות פיתוח, שהבולט והמוכר מהן הוא מדריך לעיצוב ממשקי משתמש של Microsoft (2007).

תכנון התגובה לפעילות חריגה של המפעיל

טעויות-לכאורה

במקרים מסויימים, כאשר הסיכון של חוסר שליטה הוא גבוה, אין לנו ברירה אלא לאפשר למפעיל לטעות. זאת, כשמדובר בפעולות חיוניות של המפעיל, כגון, לצורך פתרון בעיות במצבי חירום, וחייבים לאפשר לו שליטה במערכת. כך למשל, בדוגמא של השתקת ההתרעה במוניטור רפואי. במקרים כאלו, המערכת חייבת להגיב נכון: ראשית, לברר מה היתה כוונת המשתמש בביצוע פעולה הבעייתית, האם היא היתה בטעות או שאולי במתכוון, לצורך פתרון בעיה שחורגת מהשגרה, ולהגיב בהתאם. פעולות שמחייבות בירור לגבי כוונת המשתמש נקראות כאן "טעויות לכאורה".

גילוי מצבים של טעות-לכאורה

במערכת שהפעלתה מוגדרת על ידי תרחישים יש אפשרות לגלות מקרים של טעויות לכאורה, על ידי מעקב אחר אופן התנהלות הפעילות בהשוואה לתרחיש. לצורך הדוגמא, נניח שהשלט-רחוק של הממיר הדיגיטלי צריך לתמוך גם בתרחיש של איתחול, כלומר, בנוסף לטלביזיה, השלט צריך לכבות ולהדליק גם את הממיר הדיגיטלי, וכן בנוסף לממיר, השלט צריך להחליף ערוצים גם בטלביזיה. כעת, נניח שהמפעיל תוך כדי סריקת תחנות בממיר לוחץ בטעות על מקש הכיבוי-הדלקה. אם השלט מתוכנן לפעול על פי תרחישים, השלט יכול להתריע על החריגה, ולספק אינדיקציה לכך שהמפעיל צריך להעביר את השלט למצב שליטה בטלביזיה לפני שימוש במקש זה. דוגמא אחרת היא תאונת האימונים צאלים א'. אם מניחים שאת מהלך התרגיל ניתן היה להגדיר כתרחיש, הרי שטעות של פליטת פה יכולה היתה להתגלות כחריגה ממהלך התרחיש.

תהליך הגילוי של מצבי טעות-לכאורה מבוסס על תרחישי הפעלה. לשם כך, תרחישי ההפעלה צריכים להיות מוגדרים היטב, וצריך לממש אותם כחלק אינטגראלי של המערכת, כדי שבזמן ריצה, המערכת תוכל להשוות את פעילות המפעיל מול התרחישים.

תכנון לפעולות בלתי צפויות

פעולות בלתי צפויות של המפעיל עלולות להתבטא בתגובה בלתי צפויה של המערכת. כשאנחנו מגדירים מערכת, אנחנו מגדירים את ההתנהגות שלה על פי תרחישים. פעולה בלתי צפויה זוהי פעולה במצב שאינו מתאים לאף תרחיש.

כאשר פעולת המפעיל היא בלתי צפויה, המערכת יכולה להגיב במספר דרכים. המערכת יכולה להתעלם מהפעולה, להציג אזהרה בולטת פחות או יותר באופנים שונים, או להודיע הודעת שגיאה שמחייבת את אישור המפעיל, תוך התעלמות מהפעולה. המערכת יכולה גם לאשר את הפעולה במספר דרכים: להפסיק את המשימה הנוכחית, עם או ללא אישור המפעיל, ולעבור למשימה חדשה, לסנכרן את מצב המערכת למצב סטנדרטי של התחלת פעילות או למצב הקודם, וכד'.

האתגרים שבניהול פעולות בלתי צפויות הם: א. להבטיח שתגובת המערכת תהיה מוגדרת היטב, ב. להבטיח שהמפעיל יוכל להווכח בטעות שלו ולהבין את תגובת המערכת, ג. לוודא שתגובת המערכת היא ראויה וד. לאתר מצבים של כמעט-תאונה.

מפרטי האינטראקציה

בכדי להבטיח שהמערכת תגן בפני טעויות המפעיל, מפרטי האינטראקציה צריכים לתאר את ההפעלה על פי תרחישי ההפעלה. התרחישים מאפשרים שפה משותפת בין אנשי הפיתוח לבין המשתמשים. שפת המודלים האוניבסלית UML מציעה דרך לתאר את ההפעלה בעזרת תרשימי פעילות. תרשימי הפעילות מאפשרים

למפתחים לנסח את הפתרון במונחים של בעית התפעול, כאשר משלבים את פעילות המפעיל ביחד עם פעילות מערכת. את תרשימי פעילות ברמת המערכת ניתן לגזור ממודל הפעילות המשולב, על ידי קיצור המעברים בין פעילויות המפעיל.

הישום של המודל המבטא את התרשימים האלו מאפשר למפתח לזהות מצבים של טעות-לכאורה. למשל, אם המפעיל ניסה להתניע, אבל עוד לפני שהמנוע נכנס לפעולה, המפעיל ביקש לדומם את המנוע. זהו מצב לא נורמלי, שאינו מבוטא באף תרחיש סביר, ולכן המערכת צריכה להתריע על החריגה ולברר עם המפעיל האם הוא התכוון לפעולה האחרונה, או שזו בוצעה בטעות.

מגבלה של המודל הפנים-מערכתי היא בכך שהוא אינו כולל תיאור של המשוב למפעיל. למשל, בסיום שלב ההתנעה, צריך ליידע את המפעיל אם ההתנעה הצליחה או נכשלה. התיאור בעזרת מודל הפעילות המשולבת כולל את המשוב שהמפתח נדרש לספק למפעיל, בעוד התיאור המקוצר אינו כולל משוב זה. תיאור האינטראקציה בעזרת מודל הפעילות המשולבת מאפשר מעבר סיסטמטי משלב האפיון אל שלב התיכון, כאשר בשלב התיכון מגדירים את הפקדים ואת הפרוטוקולים של התקשורת, במונחים של ארועים.

מודל האינטראקציה

את המודל האינטראקציה צריך ליצא מהמפרטים אל מערכת היעד, על מנת לאפשר איתור מקרים של חריגה של המפעיל מהפעילות המוגדרת על ידי התרחישים. את האלמנטים המרכיבים את האינטראקציה ניתן ליצג בפורמט כגון Excel כך שיהיו ניתנים לקליטה כטבלאות של מערכת היעד. מערכת היעד צריכה לכלול שכבה מעל לשכבת ניהול הקלט, שתאפשר השוואה של פעילות המפעיל לתרחישים, והתרעה על פי תפריט של התרעות, שגם הוא ספציפי לאופי הישום. התרשים הבא מציג דוגמא של מערכת להגדרת מודל האינטראקציה במונחים של תרחישים, שמאפשר יצוא של המודל לצורך מימוש במערכת היעד:

תרשים 5 – מפרטי האינטראקציה עבור השלט-רחוק של הממיר הדיגיטלי

הגדרת האינטראקציה כוללת אובייקטים, מצבים, פונקציות, מקרי-שימוש ופעילויות המתארות את התרחישים. את מפרטי האינטראקציה חשוב לפתח בעזרת תוכנה שמאפשרת עדכונים אמינים, תוך כדי בדיקת המשמעות של שינויים על רגישות המערכת לטעויות המפעיל.

מקורות

Andre A. and Degani A., 1997, Do you know what mode you're in? An analysis of mode error in everyday things. in M. Mouloua and J.M. Moonce (Ed.), *Human-automation interaction: Research and practice*, pp. 19-28, Mahwah, N.J.: Lawence Erlbaum. (available at: http://ic.arc.nasa.gov/people/asaf/hai/pdf/Do you know what mode.pdf)

Bevan N., 2001, International standards for HCI and usability, *International Journal of Human-Computer Studies*, Volume 55, Number 4, pp. 533-552(20), Academic Press (available at: http://www.usabilitynet.org/tools/r_international.htm)

Casey, S., 1998, *Set Phasers on Stun*, Aegean Publishing Company, Santa Barbara, Ca.

Jacobson, I., 1987, Object-oriented development in an industrial environment, Conference proceedings on Object-oriented programming systems, languages and applications, p.183-191, October 04-08, Orlando, Florida, United States

Microsoft, 2007. Office System Document: UI Style Guide for Solutions and Add Ins (available at: http://www.microsoft.com/downloads/details.aspx?familyid=19E3BF38-434B-4DDD-9592-3749F6647105&displaylang=en)

Norman, D. A., 1983, Design rules based on analyses of human error, *Communications of the ACM*, v.26 n.4, p.254-258, April 1983 (available at http://cpe.njit.edu/dlnotes/CIS/CIS732_447/Cis732_1R.pdf)

Norman, D. A., 1990, Commentary: Human Error and the Design of Computer Systems, Editorial published in *Communications of the ACM, 1990*, 33, 4-7.

Walker, J.S., 2004, *Three Mile Island: A Nuclear Crisis in Historical Perspective*,. Berkeley: University of California Press

Wikipedia, תאונת האימונים צאלים א' (http://he.wikipedia.org/wiki/%D7%90%D7%A1%D7%95%D7%9F_%D7%A6%D7%90%D7%9C%D7%99%D7%9D_%D7%90%27).

ביו - אבי הראל

מתמטיקאי, מומחה בשילוב גורמי אנוש בממשקי הפעלה. מעל שלושים שנים מפתח ממשקי הפעלה למערכות הנדסיות. היה אחראי על פיתוח התוכנה למחשב הארטילרי "דוד" מתוצרת רפא"ל ולמוצרי התוכנה של חברת "ארגולייט", המאפשרים למפתחי תוכנה ואתרי אינטרנט לשלב גורמי אנוש במוצרים שלהם. עבודותיו זכו להכרה בינלאומית של המומחים בתחום השימושיות של מוצרים ומערכות. על עבודת המאסטר הוא זכה בפרס ע"ש לנדאו. במסגרת תערוכת קומדקס-ישראל, הוא זכה בפרס הראשון עבור המוצר של ארגולייט שבודק את השימושיות של אתרי אינטרנט ובפרס השני עבור המוצר שבודק ישומי חלונות. כיום חבר ההנהלה של UPA ישראל, וראש הוועדה הטכנית בנושא שימושיות במכון התקנים (standards@upaisrael.org).

מאמרים: http://www.ergolight-sw.com/CHI/Company/Articles/Articles.html