

מסקנות מהניסוי בתכנת EZBD

רקע

תכנת EZBD פותחה במטרה לבדוק את היתכנות המודל ההיברידי, שנועד לשפר את השימושיות של עריכת טקסטים דו-כיווניים.

התוכנה מאפשרת שמירת שמירת טקסטים ואיחזורם בסידור הזותי, כלומר, בסדר כפי שהוא נראה על גבי המסך, ועריכת הטקסטים כשהם מיוצגים בזכרון בסידור הזותי. עריכה מחדש (כגון, במקרים של גלישת טקסט משורה או במקרים של שינוי גודל חלון) נעשית על ידי סידור הטקסט בסידור כרונולוגי, וחישוב הסידור החזותי באילוצים החדשים של שטח המסך המוקצה לתצוגת הטקסט.

שיקולים במימוש התוכנה

בעיה מרכזית במימוש במעבר מהסידור החזותי לסידור הכרונולוגי נובעת מהצורך להבחין בין תוים שהשתמש הקליד לבין תוים המשתמשים לעריכה. ספציפית, קיימת בעיה להבחין בין מצבים של סוף פיסקה, שנקבעים על ידי הקלדה על המקש Enter, לבין מצבים של סוף שורה, שנגזרים מאילוצי שטח התצוגה. במימוש הנדון, האבחנה נעשית על ידי תוספת תו בתחילת כל שורה ביצוג החזותי.

כמו כן, קיים צורך להבחין בין פיסקאות שהכיווניות הראשית שלהם היא מימין לשמאל, לבין פיסקאות שהכיווניות הראשית שלהן היא הפוכה. ציפיות המשתמש לגבי סידור הטקסט מחדש בשני המקרים הן שונות. לפיכך, התו הנוסף בתחילת השורות מייצג לא רק תחילת פיסקה, כי אם גם את הכיווניות שלה.

בעיה עקרונית במצבים של עריכה מחדש היא חוסר הוודאות לגבי כוונת המשתמש במקרים שאינם סטנדרטיים. ניתן להגדיר מצבים סטנדרטיים על ידי רצף של טקסט בשפה אחת, כאשר הטקסט כולל רווחים וסימני פיסוק. במקרים אלו ניתן להגדיר ללא קושי אלגוריתמים הפיכים (דו-כיווניים) למעבר מסידור הזותי לסידור כרונולוגי. הקושי העיקרי הוא במעברים בין שפות, וביחוד כאשר המעברים כוללים סימני פיסוק. במימוש הנדון, האבחנה נעשית על ידי הגדרת שני תוי מרווח: מרווח 'עברי' ומרווח 'לועזי'. מימוש זה יוצר בעייה במצבים של צורך בטקסט מעורב, הכולל תוים משני סוגי הכיווניות.

בעיה עקרונית נוספת בעריכת טקסטים דו-כיווניים נובעת מהצורך לשלב בין שתי דרישות נוגדות: בדרך כלל, תצוגת הטקסט במסך צריכה להקבע על פי סדר ההקלדה ועל פי כיווניות השפה, אבל לעתים, המשתמש מצפה לסידורים חריגים, וצריך לאפשר לו לסמן את קטעי הטקסט החריגים, ולכפות את הסידור החריג. תכנת EZBD מאפשרת למשתמש להמיר את כל תווי המרווח בקטע המסומן על פי כיווניות כרצונו.

בדיקות

הבדיקות נעשו על ידי מר מתי אלוש, מומחה לארכיטקטורה של טקסטים דו כיווניים ממרכז הפיתוח של IBM בירושלים, ויו"ר וועדת התקינה לישומי עברית במחשבים ובאינטרנט. רשימה חלקית של ההערות מוצגת בנספח, הכולל גם התייחסות למהות הבעיות. הממצאים העיקריים של הבדיקות הם:

א'. בדרך כלל, במצבים הסטנדרטיים, התוכנה עונה על הציפיות.

ב'. במצבים מסויימים, בעיקר במצבים חריגים, יש צורך בשיפורים ובתיקונים.

ג'. שיטת האיחסון החזותית היא בעייתית, מכיוון שאינה תואמת את הישומים הנפוצים במחשבים ובאינטרנט. המקש המאפשר יצוא הטקסט בסידור הכרונולוגי אינו מספק לתואמות מליאה, מכיוון שחסרה אופציה ליבוא טקסט בסידור כרונולוגי.

ד'. הפורמט הכולל תוים בראשית שורה אינו תואם את הפורמטים בתעשייה.

ה'. שיטת המימוש על ידי תוי רווח כיווניים היא בעייתית, מכיוון שהיא מאלצת את המשתמש להוספת רווחים, שאינם תמיד רצויים. במקרה שהמשתמש מחליט על הסרת המרווחים המיותרים, תגובת התוכנה עלולה לא להתאים לציפיות.

מסקנות עיקריות

1. הסידור החזותי תואם את ציפיות המשתמש, ולכן הוא שימושי, בהשוואה לסידור הכרונולוגי.
2. המימוש על ידי תוי רווח כיווניים הוא בעייתי, ולכן יש לשקול מימוש בשיטות אחרות, למשל, בעזרת תוים בלתי נראים.
3. יש להגדיר ממשקים לתיאום בין טקסטים ביצוג החזותי הפרטי של EZBD לבין הפורמטים הנפוצים במערכות מחשב ואינטרנט.

נספח: ממצאים בבדיקה של מר מתי אלוש מה-14 במאי

כוללת התייחסות שלי להערות

1) Start the program, switch to the UI tab. Reduce progressively the window's width, until the English run ("EZBD) EasyBidi") cannot be contained in the first line. "EasyBidi" is moved to the next line but the comma staid on the first line.

התייחסות: הסיבה לכך היא שהפסיק נכלל במקטע שבאנגלית. התוצאה אינה תואמת את ציפיות המשתמש. יש לשפר את האלגוריתם כך שסימן הפיסוק שבסוף המקטע יוצמד אל סוף המקטע (אני מניח שהאלגוריתם של יוניקוד מטפל בנושא זה כראוי).

2) Start the program, switch to UI. Put the cursor on the 3rd line, press Home. Type "xyz".

a) "xyz" is moved to the next line, although there is still room in the current line.

b) "xyz" goes to the right side of the fourth line, which is semantically incorrect.

התייחסות: לא הצלחתי לשחזר את הבעיה. אני מציע להתייחס אל התופעה כאל באג, ולהתעלם ממנה בחלק של המסקנות.

3) Start the program, switch to UI, press "New". Switch to Hebrew mapping if needed, type "ABC" (where upper case represents Hebrew letters). Switch to Latin mapping, type "x". Two spaces are inserted between "C" and "x". This is not wanted.

התייחסות: הוספת המרווחים נועדה להפריד בין מקטע בעברית לבין מקטע באנגלית. שיטה זו היא בעייתית, ורצוי למצוא שיטה חליפית, שתאפשר הפרדה גם ללא תוספת תוי רווח. השאלה הבאה תהיה, כיצד ליידע את המשתמש לגבי הכיוונית של כל נקודה במסך. זה יחייב דיון נפרד.

4) I want to create the following sentence:

.new york NI EVIL I

In order to make the fullstop appear on the left side of "new york", I have to press "End" after typing "new york". This step is not needed with other common interfaces.

התייחסות: ראשית, גם אם פה ושם המשתמש נאלץ לפעולה נוספת, אין לראות בכך אסון. בעקרון, המשתמש מסתגל בקלות לדרישות שהן קונסיסטנטיות. שנית, במקרה ספציפי זה, המשתמש אינו נדרש לפעולה נוספת: הלחיצה על End מחליפה את הצורך ללחוץ על Alt+Tab.

5) Start the program, switch to UI, press "New". Switch to Hebrew mapping if needed, type "ABC" (where upper case represents Hebrew letters). Do NOT switch to Latin mapping, press Shift and type "x". The mapping is switched to Latin. Release Shift, type "y". This is typed as Latin.

In current common interfaces, releasing Shift would get back into Hebrew mapping. Your implementation may be more convenient in some cases, and will be less convenient in other cases (many users type English words within Hebrew sentences in all upper case so they do not need to change the keyboard mapping explicitly ever). Changing from common usage needs proof that it is considered improvement by a majority of users.

התייחסות לצורך לשאול את דעתם של מרבית המשתמשים:

א. האם מישוהו שאל את המשתמשים לדעתם בנושא זה כשנקבעו מסמרות בנושא זה בעבר? אם כן, כדאי לבחון את המצב לאור הסקר שנעשה בעבר.

ב. קיים שיקול של שינוי בהרגלי עבודה של משתמשים. אם התועלת של השינוי אינה גדולה, עדיף לתמוך בהרגלי העבודה הקיימים.

ג. באופן כללי, דעתם של המשתמשים היא אינדיקטור מהימן לניבוי ההתנגדות לשינוי. במקרים בהם ההתנגדות לשינוי אינה גדולה, ניתן לשקול שינויים שתורמים לשימושיות.

התייחסות לנושא השימושיות:

א. את השימושיות ניתן לבדוק תיאורטית, על ידי ניתוח תפקיד. במקרה של חילוקי דיעות, יש לערוך ניסוי שימושיות. ניסויים אלו הם יקרים ולוקחים זמן רב. לכן, בשלב זה, עדיף שננסה את הדרך של ניתוח תפקיד.

ב. השאלה כאן היא האם אצל מרבית המשתמשים, אחרי אות גדולה באנגלית תבוא אות קטנה באנגלית, אות גדולה, אות עברית, ספרה או רווח. במקרה של רווח, יש גם שאלה לגבי מה תהיה האות הבאה אחרי הרווח. ההיכרות שלי עם טקסטים במשרד, בתעשייה ובאקדמיה היא שאחרי אות גדולה באנגלית בדרך כלל מופיעה אות קטנה באנגלית. בטקסטים מתמטיים, לעתים קרובות נכתבת אות בודדת באנגלית, ואז הטקסט ממשיך בעברית (אחרי רווח).

ג. אל המקרה של הקלדה ב-CAPS LOCK עדיף להתייחס כאל תוצאה ולא כאל סיבה. זוהי דרך של הסתגלות המשתמשים לבעיה שאחרי הקלדת אות גדולה באנגלית, המקלדת עוברת למצב עברית, במקום להשאיר במצב אנגלית.

ד. כמובן, תמיד קיימת האופציה לאפשר למשתמש לבחור את האופציה החביבה עליו. במיקרוסופט הם אלופים בזה. במקרה כזה, כדאי מאוד שנגדיר מה תהיה ברירת המחדל. לדעתי, ברירת המחדל צריכה להיות כמו בתוכנת ההדגמה, מכיוון שהיא מכוונת לקהל של פקידים. אקדמאים יכולים למצוא את הפרמטר שמגדיר את האופציה שמתאימה לצרכיהם.

6) It seems that the clipboard functions work only within the program itself. We need to see what happens when copying/pasting text with other applications.

התייחסות: בישום הנוכחי, ההעתקה היא של הטקסט בסידור החזותי, אבל ללא תוי הבקרה בראשית השורות. אני מסכים שחשוב יותר לאפשר העתקה והדבקה בסידור הכרונולוגי.

7) In a RTL para, in order to move the cursor from the first (rightmost) word of a line to the previous word (leftmost word on the previous line), I need to press Up, End and then Right arrow. To move to the previous word when both words are on the same line, I only need to use Right arrow.

Other common interfaces allow to use Right arrow whether both words are on the same line or not, which is more convenient, IMHO.

התייחסות: יש פנים לכאן ולכאן. מבחינת הרגלי השימוש, והתאמה לישומים המקובלים, ההערה במקום. מבחינת גורמי אנוש, הישומים המקובלים טועים במידה מסויימת, אבל לא נראה שנוכל לשנות את העולם.

8) Your application works in its own ecosystem, working only on files that it has created. In order to allow proper evaluation of the interface, you need to provide the possibility to use it on texts which have been created using e.g. Notepad.

התייחסות: הערה זו מקובלת עלי לחלוטין.