

ויזואליזציות בתחום ארכיטקטורת המחשב – טקסונומיה

ססיל יחזקאל

המחלקה להוראת המדעים, מכון ויצמן למדע

מבוא

מקובל לחלק את הויזואליזציות לשתי קבוצות עיקריות: AV) Algorithm Visualization (לחקירת אלגוריתמים מופשטים ו-PV) Program visualization (להמחשת קוד או נתוני התוכנית בזמן הביצוע).

בחלקו הראשון של מאמר זה מוצגת טקסונומיה שפיתחתי הממוקדת בקבוצת PV להמחשת ארכיטקטורת המחשב. מטרת הטקסונומיה לסייע באפיון ומיון של סביבות ויזואליזציה של תוכניות הקיימות בתחום. היא מהווה כלי להכוונת אנשי פיתוח או אנשי הוראה לעיצוב או בבחירת סביבה על פי קריטריונים דידקטיים. הטקסונומיה מתייחסת לסביבת ויזואליזציה כמכלול הכולל את הויזואליזציה ואת הפעילויות להפעלת התלמיד בשל חשיבותן לתהליך הלמידה ולא כמקובל בטקסונומיות אחרות לויזואליזציה בלבד.

בחלקו השני של המאמר נעשה שימוש בטקסונומיה זו לתיאור סביבת ויזואליזציה EasyCPU שפותחה לתלמידי תיכון הלומדים מדעי המחשב ואלקטרוניקה, בה הושם דגש מיוחד להיבטים הקוגניטיביים והדידקטיים.

סקירת טקסונומיות העוסקות ב-PV

טקסונומיה של ויזואליזציה מתבססת על קריטריונים של מיון ומתמקדת במאפיינים של ויזואליזציה. בספרות מוצגות טקסונומיות שונות של ויזואליזציה. Myer [4] ממין סביבות ויזואליזציה לאורך שני צירים: אספקטים של הקוד (קוד, נתונים, או אלגוריתם) וסגנון ההצגה (סטטי או דינמי). Roman ו-Cox [8] בנו טקסונומיה על פי מודל תיאורטי. הם רואים את הויזואליזציה כמיפוי של התוכנית והצגה גרפית. הטקסונומיה שהם בנו מחולקת לחמש קטגוריות: תחום, הפשטה, שיטה, ממשק והצגה.

מאמר זה מציג טקסונומיה של סביבות ויזואליזציה^(*). ישנו מגוון רחב מאוד של ויזואליזציות העוסקות בתחום של מדעי המחשב המכונה בשם Software Visualizations (SV). מטרת הויזואליזציות מגוונות: הן פותחו לאנשי מקצוע ככלי פיתוח, או ככלי הדגמה להוראה, או ככלי הדגמה ותרגול להוראה-למידה אינטרקטיבית.

הספקטרום הרחב של ציפיות מויזואליזציה, הגישות השונות למימושן וההתמקדות בדגשים שונים הנוגעים לדרך המימוש משתקפים במגוון ההגדרות שניתנו למונח זה על-ידי חוקרים בתחום.

במאמר זה אתייחס לויזואליזציה על פי ההגדרה הבאה: שימוש בכלים ממוחשבים לעיצוב גרפי סטטי ודינמי, תוך ניצול הפוטנציאל והאפקטים המצויים באינטראקציה אדם/מכונה (או באנגלית Man Machine Interface - MMI) לתיאור והסבר של מודלים.

סביבות ויזואליזציה של תוכנה למטרות דידקטיות עוררו ציפיות רבות בקרב אנשי פיתוח והוראה. במחקרים שנעשו לאחרונה ושנועדו להערכה של תועלתן של סביבות ויזואליזציה של תוכנה הגיעו למסקנה שהגורם המכריע לגבי התועלת של סביבת הויזואליזציה הינו מידת האינטראקציה של התלמיד עם הסביבה ואיכות הפעילויות הניתנות לתלמיד ולא תכונות הסביבה עצמה ([2], [3]).

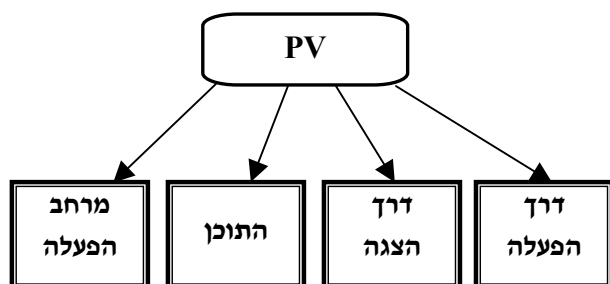
(*) תקציר של מאמר:

“Visualization of Computer Architecture: Taxonomy”, Yehezkel, C., Innovation and Technology in Computer Science Education, ITiCSE, 2002.

הטקסונומיה של סביבות ויזואליזציה של תוכניות

הטקסונומיה המוצגת במאמר זה הינה ייחודית בהתייחסותה לסביבת ויזואליזציה כמכלול. היא מתייחסת לדרך ההפעלה של התלמיד בסביבה. היא מאפיינת את השיטות שבהן סביבת ויזואליזציה יוצרת אינטראקציה בין התלמיד לויזואליזציה. הטקסונומיה מהווה כלי יעיל לניתוח והערכה של סביבות ויזואליזציה קיימות במטרה לבחון את מידת התאמתן לתוכנית לימודים ולקהל יעד. היא מהווה גם כלי עזר לאפיון ולפיתוח של סביבות ויזואליזציה חדשות.

הטקסונומיה בנויה מקטגוריות במספר רבדים. הרובד העליון של הטקסונומיה מכיל את הקטגוריות הבאות:



הרובד העליון

ברובד העליון של הטקסונומיה השתמשתי בחלק מן הקטגוריות המוגדרות על ידי Price ועמיתיו [7]. בניתי את הרבדים המשנים (מלבד הענף של דרך הפעלה שאמצתי בחלקו) בחלוקה לקטגוריות בהתאם לתחום של סביבות ויזואליזציה השייכות לתחום. בחלוקה התמקדתי בהיבטים הדידקטיים והקוגניטיביים. בקטגוריה של **מרחב הפעלה** הכוונה היא למיון של סביבות הויזואליזציה על פי פירוט טכני של אפשרויות הפעלה והשלכותיהם על תפוצת הסביבה וסוגי המשימות שניתן להטיל על התלמידים בסביבת הלמידה. הקטגוריה של **התוכן** מתארת את דרכי ארגון התוכן לצורך הדגמתו בויזואליזציה. הקטגוריה **דרך הצגה** מפרטת את מגוון האפשרויות העומדות לרשות המפתח להצגת המודל והשלכותיהן על התלמיד. הקטגוריה **דרך הפעלה** מתארת כיצד המפתח של הסביבה יוצר אינטראקציה בין התלמיד לסביבה.

Price ועמיתיו [7] בנו טקסונומיה רחבה מאד המתפרסת על סביבות ויזואליזציות דידיקטיות והנדסיות במדעי המחשב. הם בחנו את הויזואליזציות על פי קריטריונים כלליים ולא התמקדו באספקטים הדידקטיים והקוגניטיביים של הוראה-למידה באמצעות ויזואליזציה.

סביבות הויזואליזציה שנדון בהן במאמר הנוכחי הן סביבות ויזואליזציה העוסקות בתחום הנושק לתוכנה ולחומרה (תחום של הנדסת אלקטרוניקה ומחשבים) העוסקות בהוראה-למידה ומשתייכות לקבוצת ויזואליזציה PV.

דוגמאות מייצגות של סביבות PV בתחום

Cassel ועמיתיה [1] ערכו רשימה של סביבות PV מייצגות שמשמשים בהן בהוראת ארכיטקטורת המחשב. כדי להמחיש היבטים מסוימים של הטקסונומיה בחרתי שלוש סביבות PV מייצגות: EasyCPU, RTLsim ו-LMC. סביבות אלה מתוארות בפירוט במאמר [10].

סביבת EasyCPU פותחה בישראל כחלק אינטגרלי של יחידת הלימוד ארגון המחשב ושפת סף במסגרת תכנית הלימודים החדשה במדעי המחשב לתלמידי תיכון ([10],[11]). הסביבה כוללת ויזואליזציה של מודל מפורט של מחשב משובץ מעבד ממשפחת אינטל 80X86.

RTLsim היא סביבה שמדמה את מעברי הנתונים במעבד MIPS. כאשר מריצים את ההדמיה, הסטודנט נדרש לתפקד כיחידת בקרה ועליו להפעיל את קווי הבקרה המתאימים בתזמון נכון [10].

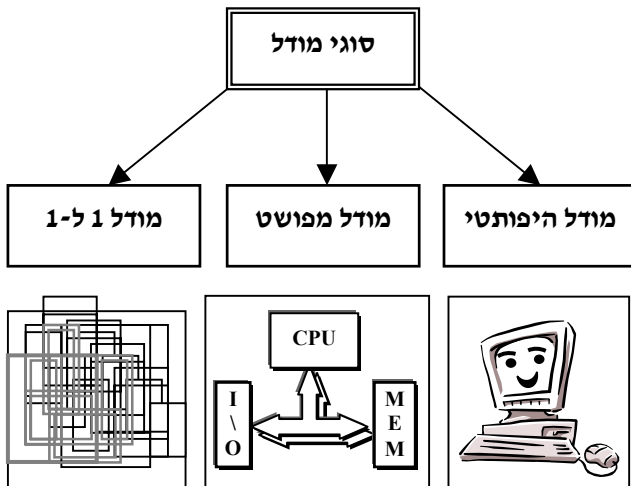
LMC היא סביבה המבוססת על הפרדיגמה של Little Man Computer. עובד דואר קטן מבצע פעולות מיון בתוך חדר דואר כאנלוגיה ליחידת הבקרה של המחשב. המטרה היא לספק לסטודנט ויזואליזציה של מרכיבי ארכיטקטורת המחשב ולאפשר לו לעקוב אחר מחזורי הבאת הוראה במהלך ביצוע תוכנית [12].

בטקסונומיה אשתמש בסביבות אלה כדי להדגים את מאפייני הקטגוריות השונות (ההדגמות יצוינו בכתב נטוי).

2.1 סוגי מודל

השאלה המתבקשת כאשר עוסקים בתוכן היא: מהו סוג המודל שאנו מציגים?

חילקתי את מגוון האפשרויות לשלוש קטגוריות אופייניות: **מודל "השואף להציג מחשב 1 ל-1"**, **"מודל מפושט" (simplified)**, **"מודל היפותטי"**. במודל השואף להציג מחשב 1 ל-1 מטרת המפתח היא לנסות לדמות או להציג בנאמנות גבוהה ככל האפשר את המחשב האמיתי. במודל מפושט הכוונה היא למודל קרוב למחשב האמיתי אבל פשוט יותר. במודל היפותטי, הכוונה היא למודל שאינו מתיימר לדמות מחשב מסוים אלא מציג עקרונות כלליים על פיהם פועל המחשב.

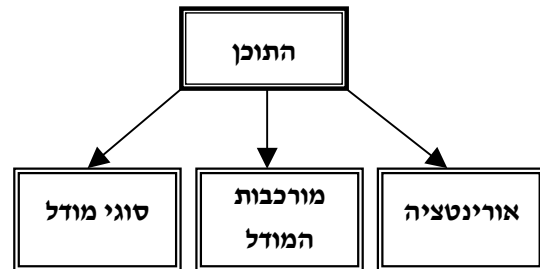


1. מרחב הפעלה

מרחב הפעלה כולל את הפירוט הטכני של **תנאי ההפעלה** של הסביבה ואת **מגבלות הסביבה**. מגבלות הסביבה כוללות את פירוט המגבלות הפנימיות של הסביבה (לדוגמה הגודל המרבי של התוכנית שניתן להמחיש את ביצועה). הנתונים אומנם טכניים אבל יש להם, לעיתים, השלכות מהותיות לגבי נידות, תפוצה, נגישות ומורכבות הפעילויות שניתן לבצע בסביבה.

2. התוכן

קטגוריה זו מתייחסת לדרכי ארגון המידע לצורך הדגמתו בעזרת ויזואליזציה. ניתן לאפיין את דרכי ארגון המידע בשלושה אופנים: **סוג המודל** שבו המידע מעוגן, **מורכבות המודל** והאורינטציה של המודל.



אנו עוסקים באפיון סביבות להצגת ארכיטקטורה והדגמת תפקוד המחשב. ההדגמה מבוססת על מודל. מפתח סביבת למידה יכול לבנות את הסביבה בהתבסס על **סוגים שונים של מודלים**. המפתח בוחר את המודל על פי קהל היעד של הסביבה ועל פי גישתו הדידקטית.

המפתח יתאים למודל שבחר את רמת **מורכבות המודל** ליכולות הקוגניטיביות של קהל היעד. הרקע המקצועי של המפתחים והאורינטציה המקצועית של קהל היעד קובעים גם את הדגשים בתחום התוכן. על פי הדגשים מגדירים את **האורינטציה של המודל**.

בסוף דצמבר התקיימה במכון ויצמן סדנה בנושא "הבטים דידיקטיים בהוראת ארגון המחשב ושפת סף. החומרים מהסדנה נמצאים באתר המרכז הארצי. אתם מוזמנים להיכנס לאתר ולראות את המצגת.

הגישה הזו נראית נכונה, אבל לדעתי, יש לה השלכות לא רצויות בתחום שבו אנו עוסקים. בתחום זה קיים שילוב של חומרה ותוכנה המחייב ירידה לפרטים. הכללה מונעת תיאור פרטים השייכים והמזהים מחשב ספציפי. מרחב הפעילויות שניתן לבצע בסביבה במקרה זה מוגבל. הרחבת מרחב הפעילויות האפשריות בסביבה מחייבת "המצאה" של פרטים, ותוצאות הלוואי הן בכך שמאלצים את הלומד להעמיס את זיכרונו בפרטים שהם ייחודיים לסביבה וירטואלית ואין בהם שימוש מחוצה לה.

המפתח יכול לבנות מודל מקורי על פי השקפתו, או להתבסס על מודל קיים בספרות, או להתבסס בבניית מודל על פי פרדיגמה המוצגת על ידי מומחים בתחום התוכן בספרות.

מפתחי LMC החליטו להציג את הפרדיגמה ה- "Little Man Computer" של Madnick ב- (Englander, 1996) על פיה עושים אנלוגיה בין המעבד לדוור הממייין את הדואר הנכנס לתאים שונים.

2.2 מורכבות המודל

מורכבות המודל נמדדת **ברמת התיאור וברמת הפירוט** (בדרך ההצגה של המודל). ניתן להציג מרכיבים או מידע בדרגות שונות של **רמת הפירוט**.

2.3 אוריינטציה

קטגוריה זו מאפשרת לדרג באופן יחסי סביבות לפי התכנים המודגשים בהן. התחום שבו אנו עוסקים מורכב מבחינה זו שהוא שוכן בחפיפה בין שני תחומי תוכן שנוהגים להתייחס אליהם כשני תחומים נפרדים: חומרה ותוכנה. גם כאן יש מגוון אפשרויות בחירה של רמת ההפשטה (Abstraction). התוכנה נחשבת לרמת הפשטה גבוהה יותר מהחומרה בהיררכיה תוכנה-חומרה. באמצעות מודל ניתן לתאר את ארכיטקטורת המחשב ואופן פעולתו ברמות הפשטה שונות. כאשר הדגש מושם על מרכיבי התוכנה רמת ההפשטה (abstraction) גבוהה יותר ולהיפך. על רקע זה יצרתי שלוש קטגוריות; האחת מאפיינת את הדגשת **התוכנה**, השנייה מאפיינת את השילוב בין **תוכנה וחומרה** והשלישית מאפיינת הדגמת היבטי **חומרה**.

2.1.1 מודל "תואם מחשב מסוים 1-1"

כאשר המפתח בוחר לנסות ולדמות או להציג מחשב מסוים על כל מרכיביו בשאיפה ליצור סביבה נאמנה ככל שניתן למציאות (Fidelity ו-Completeness). המודל יקרא **"תואם מחשב מסוים 1 ל-1"**. לגישה זו ישנן מגבלות רבות. אחת מהן היא הקושי לשמור על נאמנות למציאות בשל המורכבות של המחשב ובשל מגבלות כלי הדמיה וההצגה. מפתחים הדוגלים "בנאמנות למציאות" נאלצים להתפשר בגלל מגבלות כלי ההצגה. לא רק כלי ההצגה מוגבלים גם כישורי החישה והזכירה של התלמיד מוגבלים. פיתוח שאינו מתחשב במגבלות אלה עלול לחולל טעויות כתוצאה מתפיסה לקויה, להקשות על איתור המידע הרלוונטי או ליצור עומס גדול מידי על הזיכרון אצל הלומד. ניתן לשייך לקטגוריה זו סימולציות וכלי ניפוי מקצועיים. בגלל ריבוי הפריטים ומורכבות המידע המוצג, כלים אלו אינם מתאימים לתלמיד מתחיל.

2.1.2 מודל מפושט

כאשר המפתח בוחר לתכנן את המודל מראש, בהתחשב במגבלות הן של ההצגה והן של קהל היעד ולפשט את המודל בהתאם למגבלות ולמטרות הדידקטיות, המודל יקרא **"מודל מפושט"**.

בפיתוח סביבת EasyCPU בחרנו לדמות מחשב מסוים; מחשב התואם את המחשב שעליו התלמיד עובד (מחשב משויבץ במעבד ממשפחה Intel 80X86) אך מכיון שמבחינה דידיקטית אין רציונאל לדמות את המחשב על כל מרכיבי החומרה והתוכנה שבו, התאמנו את רמת הפירוט והמורכבות לאוכלוסיית היעד.

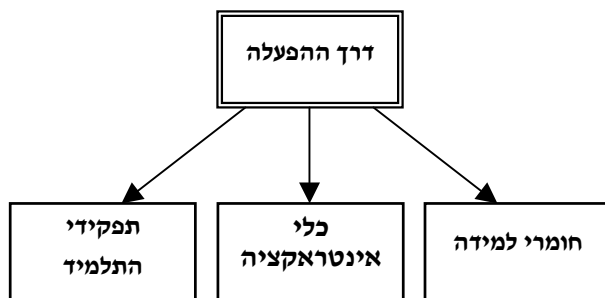
2.1.3 מודל היפותטי

ישנם מפתחים המעוניינים להציג מודל היפותטי. מודל שאינו מדמה מחשב מסוים. בדרך כלל המניע הוא רצון להכללה. הכללה במובן שהמודל לא נתפס לפרטים המאפיינים מחשב מסוים, אלה מנסה להמחיש עקרונות כלליים המשקפים ארכיטקטורה של משפחה של מחשבים או מנגנון כללי שעל פיו פועלת קבוצה מסוימת של מחשבים.

4. דרך ההפעלה

עד כה עסקנו באפיון ההיבטים שמפתח הסביבה מציג לתלמיד וכיצד הם מוצגים לתלמיד. כעת נעסוק באינטראקציה שהמפתח יוצר בין התלמיד לסביבה. המפתח הוא שמגדיר את התפקיד שהתלמיד ימלא בסביבה: **תפקידי התלמיד** בסביבה. הוא מפקיד בידו את התפקיד, את הכלים למילוי התפקיד וכלים נוספים החיוניים להפעלת הסביבה כולה: **כלי אינטראקציה**. התפקיד שממלא התלמיד הוא בעל משמעות עמוקה מבחינה דידקטית. לכלים אין משמעות אם אין הפעלה של התלמיד. הפעלת התלמיד מתבצעת על ידי מתן משימות מותאמות לתפקידים שהופקדו בידו ולכלים העומדים ברשותו.

המשימות יכללו **בחומרי הלמידה**. סביבת היוזואליזציה יחד עם חומרי הלמידה מהווים סביבת למידה שלמה.



חומרי הלמידה כוללים פעילויות אשר מטרתן להפעיל את התלמיד על ידי הגדרת משימות. כאמור המפתח בונה תפקיד אשר התלמיד צריך למלא במסגרת הסביבה. הפקדת התפקיד בידי התלמיד לא מספיקה שכן יש להטיל עליו משימות קונקרטיות לביצוע וזאת על מנת שתהליך הלימודים יהיה מובנה ומונחה. ניתן לשבץ את המשימות בתוך סביבת העבודה או בחומרי למידה נלווים כמו חוברת פעילויות.

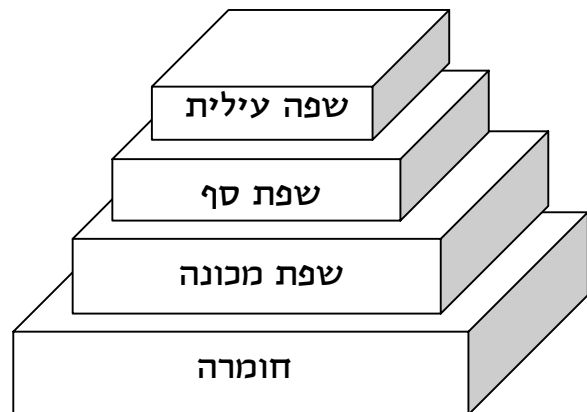
סביבת ה-*EasyCPU* כוללת מדריך למשתמש שבו התלמיד לומד להתמצא בסביבת ה-*EasyCPU*, ולפרש את מה שמוצג לו בחוברת בת אחת-עשרה פעילויות עם קבצי משימות מותאמים. הפעילויות מהוות רצף מדורג של משימות לעבודה עצמית שבמהלכן התלמיד לומד את יסודות מבנה המחשב והתכנות בשפת סף.

2.3.1 תוכנה

2.3.2 תוכנה-חומרה

2.3.3 חומרה

אם נדרג את שלוש הסביבות: *EasyCPU*, *LMC* ו-*RTLsim*, הרי שסביבת *LMC* מתאימה לקטגוריה של הדגשת התוכנה, סביבת *EasyCPU* מתאימה לקטגוריה של השילוב תוכנה-חומרה וסביבת *RTLsim* מתאימה לקטגוריה של היבטי חומרה.



3. דרך ההצגה

המטרה המשותפת של מפתחי SV על פי Petre ועמיתיה [6] היא לגרום למידע חבוי להיות נגיש, גלוי ומובן מאליו.

בקטגוריה זו נדון בשיטות הצגה שבהן משתמש המפתח כדי לנסות להבליט את המידע הרלוונטי ולעשות אותו "מובן מאליו". ישנם אופנים שונים שבהם ניתן להציג את אותם מרכיבים במודל. לעיתים השימוש בריבוי הצגות מאפשר לתאר פנים שונות של מרכיבי המודל, ליצור תמונה שלמה יותר ולסייע בהבנה מקיפה של המשמעות השונות. כפי שנאמר לגבי המורכבות (ריבוי פרטים ופריטים), בשל מגבלות ההצגה, לעיתים קיים אילוץ בהעלמת פריטים או פרטים: **אפשרויות צמצום ומיקוד**.

ב-*EasyCPU* יש לתלמיד אפשרות לבחור את בסיס המידע המוצג. המידע המוצג על המסך עם העלאת התוכנה הוא בבסיס בינרי כפי שהוא קיים במציאות במחשב. יחד עם זאת ניתנת לתלמיד אפשרות לבחור בייצוגים חלופיים.

תיאור של סביבת ה-EasyCPU במסגרת הטקסונומיה

סעיף זה מתאר את מאפייני סביבת ה-EasyCPU במסגרת הטקסונומיה שהוצגה לעיל. התיאור כולל ציון של הקטגוריות השונות של הטקסונומיה המתאימות למאפייני הסביבה.

סביבת ה-EasyCPU פותחה כמכלול להוראה-למידה הכולל בתוכו את הויזואליזציה וכן חומרי למידה (ספר עיוני, מדריך למשתמש וחוברת פעילויות) שפותחו במקביל, תוך שמירה על תאימות מלאה.

מטרת הסביבה והגישות הדידקטיות לפיתוחה:

מטרת הסביבה מבחינת התוכן היא לתת "Insight" (תובנה) לארכיטקטורה של המחשב, לאופן תפקודו ולתהליך ביצוע התוכנית, וזאת בהתאם לדרישות תוכנית הלימודים ולאוכלוסיית היעד: תלמידי תיכון בכיתות י' ו-יא' במדעי המחשב, טכ"מ ובאלקטרוניקה. *תהליכי התאמת המודל ליכולות הקוגניטיביות של קהל היעד, ולדרישות התוכן שייכים לקטגוריות 2.2 ו-2.3 (מורכבות המודל ואורייטציה) בהתאמה.*

שפת סף היא שפת תכנות ברמה אחת מעל שפת המכונה הנמוכה ביותר בהיררכית שפות המחשב. *מיקום התכנים בהיררכיה חומרה-תוכנה שייך לקטגוריה 2.3.2 (חומרה-תוכנה).*

לימוד שפת סף אינו מטרה בפני עצמה, אלא אמצעי להבין את משמעות הפעולות האלמנטריות (ברמה הבסיסית ביותר) של המחשב, להכיר את המרכיבים היסודיים שעליהם בנויות שפות עיליות, להכיר את אבני הבניין של טיפוסים נתונים וארגונים, וכן להבין את תהליכי התרגום, ההרצה והביצוע של תוכנית מחשב על המרכיבים השונים המעורבים בתהליך. מרכיבים אלו הינם בעלי מנגנוני פעולה מורכבים שתלמידים מתקשים להבין. לכן, כאמור, בחרתי משיקולים דידיקטיים ליצור מודל מפושט (simplified) של מחשב מסוים (מחשב משובץ במעבד ממשפחה Intel80X86). *גישה דידיקטית המתאימה לקטגוריה 2.1.2 (מודל מפושט)*

תהליך הלמידה בסביבת ה-EasyCPU מחולק לשני שלבים: לתלמיד המתחיל (Basic mode) ולתלמיד המתקדם (Advanced Mode). חלוקה זו נועדה לדרג את התפקידים המוטלים על התלמיד בהתאם לשלבי הלמידה שבהם הוא נמצא.

דירוג תפקידים, בהתאם לשלבי הלמידה, שייך למאפייני קטגוריה 4 (תפקידי התלמיד).

המטרה של רמת העבודה הבסיסית (basic mode) היא לאפשר לתלמיד להפנים את המתרחש במחשב במהלך ביצוע ההוראה באמצעות ההמחשה הויזואלית. במסך ה-Basic mode מוצגים בחלונות נפרדים דגמי המעבד, הזיכרון ויחידת קלט/פלט בסיסית ופסי המידע המחברים בין היחידות השונות. תיאור היחידות תואם את מרכיבי המחשב המדומה (המחשב שעליו רצה תוכנת ה-EasyCPU) אך יחד עם זה התיאור מפושט מאד. *קטגוריה של מודל מפושט 2.1.2.*

ברמת העבודה הבסיסית התלמיד לומד להכיר את מבנה ההוראה, את משמעותה ואת האופן בו היא מתבצעת. את המבנה הוא לומד על ידי תהליך הרכבת ההוראה באמצעות תיבת דו-שית. לאחר בניית ההוראה, התלמיד צופה באופן ביצועה.

כאשר תלמיד בשל לפענח תוכנית או לכתוב תוכנית קצרה, הוא מוזמן להיכנס ל-Advanced Mode. ברמה זו מוצגים בחלונות נפרדים דגמי המעבד, הזיכרון ויחידות קלט/פלט בסיסיות.

ברמת עבודה מתקדמת (Advanced Mode) אין הצגה של פסי המידע המחברים בין היחידות. ברמה זו עומדים לרשות התלמיד כלים לפיתוח תוכניות: חלון עריכה, כלי הידור וקישור וכן כלי הרצה וניפוי השגיאות. בשלב זה ניתנת לתלמיד הזדמנות לכתוב תוכנית (רצף של הוראות) ולעבור דרך כל שלבי התהליך של כתיבת התוכנית, עד שלבי ההרצה ומיפוי השגיאות, בסביבה נוחה ומובנת (ניתנים כלים נוחים מותאמים לתלמיד למילוי תפקיד המתכנת – כלי האינטראקציה 4.2). כדי לבדוק את נכונות תוכניתו, תלמיד אמור לעקוב אחר ביצוע התוכנית ועיבוד המידע באמצעות כלי לניפוי שגיאות שנקרא debugger.

כלים לכתיבת תוכנית ובדיקת נכונותה, המותאמים לתלמיד עשויים להוות בו-זמנית כלים להמחשת התהליכים המתרחשים במחשב.

כלי האינטראקציה תורמים להעמקת הבנת התהליכים המתרחשים. כלי אינטראקציה שייכים לקטגוריה 4.2.

- (technical Report GIT-GVU-99-10). Gvu Center, Georgia Institute of Technology, Atlanta, GA.
4. Myers, B., A., Taxonomy of visual programming and programming visualization. *Journal of Visual Languages and Computing*, 1(1), 1990, 97-123.
 5. Pearson, M.W., McGregor, A.J. and Holmes, G.. Teaching computer systems to majors: A MIPS based solution. *IEEE Computer Society Computer Architecture Technical Committee Newsletter*, 1999, 22– 24.
 6. Petre, M. Blackwell, A. and Green, T. Cognitive questions in software visualization. In Stasko, J., Domingue, J., Brown, M. and Price, B. (eds.) *Software Visualization*, MIT Press, 1998, 453-480.
 7. Price, B., A. Baecker, R., M and Small, I. An introduction to software visualization. In Stasko, J., Domingue, J., Brown, M. and Price, B. (eds.) *Software Visualization*, MIT Press, 1998, 3-34.
 8. Roman, G.-C. and Cox, K. A taxonomy of program visualization systems. *Computer* 26 (12), 1993, 11-24.
 9. Stasko, J. and Lawrence, A. (eds.). (1998). Empirically assessment algorithm animation as learning aids. In Stasko, J., Domingue, J., Brown, M. and Price, B. (eds.) *Software Visualization*, MIT Press, 1998, 419-438.
 10. Yehezkel, C. Yurcik, W. and Pearson, M. Dean, A. Three Simulator Tools For Teaching Computer Architecture: EasyCPU, Little Man Computer, and RTLsim *Special Issue on General Computer Architecture Simulators JERIC*, 2002, 1(4).
 11. EasyCPU - סביבה ממוחשבת ללמידת מבנה המחשב מ. רוני, מ. אליהו, ס. יחזקאל, והתנסות בכתיבת תוכנה Abstracts p.100, April 95.
 12. Yurcik, W. Vila, J. and Brumbaugh, L. A web-based little man computer simulator, *Proceedings of SIGCSE '01(2001)*, 204-208.

בזכות התאימות של סביבת EasyCPU למחשב הממשי, תלמידים מתקדמים הלומדים את היחידה, "ארגון המחשב ושפת סף", יכולים לעבור לסביבת תכנות מקצועית (לדוגמה MASM או TASM) לקראת סוף שנת הלימודים.

סיכום

כאמור, לא נבנתה עד כה טקסונומיה ממוקדת בוויזואליזציה של תוכניות וארכיטקטורת המחשב. עובדה זו עוררה את המוטיבציה לבניית טקסונומיה ממוקדת בתחום הזה.

הטקסונומיה (המוצגת כאן בחלקה) פורסת מפה של מרחב אפשרויות מורכב. היא משרטטת קווים מנחים כדי לסייע למפתחים ולאנשי הוראה לנווט ולבחון את מגוון האפשרויות בכל הנוגע לתכנים, לדרישות תוכנית הלימודים, לגישה הדידקטית, להתאמה לאוכלוסיות היעד ולמגבלות כלי הפיתוח וההצגה. המגוון הרחב של הסביבות הקיימות מאפשר לשלב בהוראה מספר סביבות בהתאם לתכנים שבהם עוסקים, לפתח פעילויות עבורם ועל ידי כך לחשוף את התלמיד למודל המחשב ברמות שונות של הפשטה.

תודות

ברצוני להודות למנחים שלי בעבודת הדוקטורט: פרופ' מ. בן-ארי ופרופ' ט. דריפוס על העידוד והתמיכה ולד"ר ב. הברמן על הערותיה.

רשימת מקורות

1. Cassel, L., Kumar, D., Bolding, K., Davies, J., Holliday, M., Impagliazzo, J., Pearson, M., Wolffe, G. and Yurcik, W., Distributed Expertise for Teaching Computer Organization and Architecture (*ITiCSE 2000 Working Group Report*), *ACM SIGCSE Bulletin* 33 (2), 2001, 111-126.
2. Hundhausen, C.D., Douglas, S.A., and Stasko, J.T. (2001). A Meta-Study of Algorithm Visualization Effectiveness. Under review by the *Journal of Visual Languages and Computing*, 13(3), 259-290.
3. Kehoe, C., Stasko, J. and Taylor, A. *Rethinking the evaluation of algorithm Animation as learning aids: An Observational Study*.