אפיון ותכן מערכות מידע

הרצאה 1

[iyads@yvc.ac.il](mailto:iyads@yvc.ac.il) [iyaddd@gmail.com](mailto:iyaddd@gmail.com)

PDM = object process model

UML =unified modeling language

מערכת מידע:

זרימת המידע בתוך הארגון, תהליך זרימת המידע והמרתה למערכת ממוחשבת.

מערכת מידע ארגונית:

1. מידע = נתונים בעלי משמעות הרלוונטיים למשתמש
2. מערכת = קבות יישויות הקשורים אחד לשני אשר ישלהם מטרה משותפת
3. ארגון= מורכב מאוסף של תת מערכות אחד מהם הוא מערכת המידע.

נתונים = חומר גלם, כלומר תכונות הישויות.

מידע = נתון עם משמעות

ידע = מידע צבור

מרכיבים עיקריים במערכת המידע: קלט => תהליך => פלט => משוב בקרה => קלט וכו'..

* מודל קבלת החלטות של סיימון:

תהליך קבלת החלטות: שלב הזיהוי => שלב הניתוח והעיצוב => שלב הבחירה והבקרה.

בכל שלב ישנה מידע שונה שנדרש.

* המודל ההיררכי של אנתוני:

1. רמת התכנון האסטרטגי= מנהלים בכירים.
2. רמת הבקרה הניהולית= מנהלים דרג בינים.
3. רמת הבקרה התפעולית = מנהלים ברמה הנמוכה.
4. דרג תפעולי.

שיווק, כספים, ייצור, מערכות מידע, רכש.

תפקידה של מערכת המידע הוא לתמוך בתהליך קבלת החלטות של רמות הניהול השונות בארגון ע"י הספקת מידע הנדרש בכל שלב.

מחזור החיים של מערכת:

דרישה למערכת => פיתוח => הקמה => שימוש במערכת => שיפור המערכת => שימוש במערכת.

החלפת המערכת כאשר היא לא עונה על הדרישות.

Sdlc = מערכת הנבנת בשלבים, ניתוח ואפיון, בניה, הטמעה ותפעול.

ניתוח ואפיון= הגדרה:

1. ייזום המערכת=מנתח המערכות יושב עם הלקוח ומזהה בעיות.
2. איסוף נתונים וחקר מצב קיים=איגוד המידע למסמך על הארגון וסביבתו
3. ניתוח דרישות וחקר ישימות= זיהוי הדרישות למערכת וכתיבת דו"ח ישימות אל מול הדרישות.
4. הגדרה ואפיון המערכת= אפיון מפורט של המערכת.

השלבים לאחר אפיון:

1. עיצוב המערכת- בנייה
2. יישום המערכת-בנייה
3. הדרכה והטמעה-הטמעה
4. תחזוקה, בקרה וסיום-תפעול

Mooc = ללמד המון קורסם באותו זמן

Conrsera = סוג של אתר ללימוד המונים

Udemy = סוג של אתר ללימוד המונים

Full stack developer = סוג של אתר ללימוד המונים

מעצב תוכניות = כותב תרשימי זרימה.

QA=בודק איכות תוכנה

Maximize profit = לעלות רווחים.

R&d = מחקר ופיתוח, המביאה לגרסה חדשה או מוצר חדש.

תרגול 1 – 13.3.17

כתובת מייל אביטל שולנר [avitalshulner@gmail.com](mailto:avitalshulner@gmail.com)

הרצאה 2 – 16.3.17

פרוייקט LinkedIn, קבוצה S62

תרגול 2 – 20.3.17

הרצאה 3 – 23.3.17

גישת DFD:

מכילה תהליכים, יישויות, מאגרי מידע וזרימת המידע.

תהליך= שאילתא

יישות חיצונית=

זרם מידע=

תרגול 3.4.17

Dfd תרשים להצגת תהליכים של מערכת.

תהליך פשוט יסומן בעיגול, תהליך מורכב יסומן בעיגול כפול.

יישות תסומן כרגיל במלבן- השם תהיה ביחיד.

מאגר מידע השם יהיה ברבים.

כללים:

1. חוק שימור המידע= סנכרון בין קצוות הפונקציות בתרשים.
2. לכל תהליך יש יישות שהיא הטריגר לתהליך ויש יישות שהיא הפלט של התהליך.
3. בתרשים הקשר לא נסמן כלל את המאגרי מידע.

לדוגמא:

רמת 0= יוצגו התהליכים המרכזיים במערכת יכלול את מאגרי המידע.

רמה 1 = פירוק תהליך מרכזי

הרצאה 20.4.17

עבודת הגשה DFD

1. תיאור של המערכת
2. Context diagram
3. Dfd רמה 0 = 3 תהליכים.
4. Dfd רמה 1 = פתיחת התהליכים בdfd רמה 0.
5. מילון נתונים עבור DFD

הערות נוספות:

תהליכים:

* לכל תהליך חייב להיות לפחות קלט אחד(הקלט עובר למאגר או לתהליך אחר או ליישות).
* לכל תהליך חייב להיות לפחות פלט אחד
* אסור שיהיה קישור בין יישות למאגר ללא תהליך.
* אסור קשר בין מאגר למאגר
* אסור קשר בין יישויות
* בין מאגרים תמיד יהיה תהליך באמצע
* לא קיים תהליך רקורסיבי
* הנתונים שנכנסים למאגר הם לא אותם הנתונים שיוצאים מהמאגר

הרצאה 11.5.2017

דרישות תרשים ERD

עד 3 תהליכים בכל רמה.

הסבר ויקפדיה

בהנדסת תוכנה, Unified Modeling Language או בקיצור UML (בעברית: שפת מידול מאוחדת) היא שפת מפרט תקנית לעיצוב מונחה-עצמים. השפה פותחה במקור על ידי ג'יימס רמבאו, גריידי בוץ' ואיוואר יאקובסון בעת שעבדו בחברת התוכנה Rational (כיום חלק מחברת IBM). התיווי בשפה הוא גרפי ומאפשר תיאור מופשט של מפרטי המערכת, בדרגות שונות של דיוק. בסוף שנות התשעים, ארגון התקינה OMG תיקנן את השפה, וכיום היא מפותחת על ידו.

בשל אופיה הוויזואלי, UML היא שפה קלה-יחסית ללימוד, בהשוואה לשפות מפרט אחרות. לרוב, כל בעלי התפקידים בצוות פיתוח תוכנה מכירים את הדיאגרמות העיקריות בשפה, והדבר מסייע לתקשר את מפרטי התוכנה ביניהם.

ב-UML קיימים מספר סוגי תרשימים (Diagrams), המשמשים למטרות שונות, ומתחלקים לסוגים שונים. התרשימים השונים מורכבים משלוש אבני בניין עיקריות, גורמי יסוד, יחסים בין גורמי היסוד ודיאגרמות - התרשימים שמאגדים את גורמי היסוד והיחסים לגורמים בעלי משמעות.

גורמי יסוד[עריכת קוד מקור | עריכה]

גורמי היסוד (elements) מתחלקים לארבע קבוצות עיקריות:

גורמים מבניים (structural): אלה הם "שמות עצם" המרכיבים כל מודל ואשר בדרך כלל מתארים גורמים פיזיים או קונספטואליים. לדוגמה שחקן (Actor), שמסומן על ידי דמות אדם.

גורמים התנהגותיים (behavioral): אלה הם החלקים הדינמיים של מודל UML אשר מוגדרים כ"פעלים" של המודל. גורמים אלו מגדירים את ההתנהגות של גורם מסוים בזמן, או ביחס לגורמים אחרים במודל.

גורמים מקבצים (grouping): אלה הם החלקים שמאפשרים ארגון טוב יותר של המודל. הגורם הנפוץ ביותר השייך לקבוצה הוא package. גורם זה הוא מעין ריבוע שמאפשר לארגן בתוכו גורמים מסוגים שונים לקבוצה אחת שאיתה ניתן לקבץ גורמים התנהגותיים, מבניים ואף גורמים מקבצים נוספים. בין הגורמים השייכים לקבוצה זו: מודל, מערכת (framework), תת-מערכת (sub-system).

גורמים מפרשים (annotational): גורמים אשר בעזרתם ניתן להסביר ולפרש גורמים במודל או מודל UML. הגורם העיקרי בקבוצה זו הוא ההערה (note). בהערה המשויכת לגורם, ניתן לרשום הערות בטקסט חופשי או אילוצים (constraints) המתייחסים לגורם ומגדירים עליו אי אלו הגבלות.

יחסים[עריכת קוד מקור | עריכה]

היחסים (relationships) מגדירים את היחסים והקשרים בין האלמנטים. אף הם מתחלקים ל־4 סוגים:

תלות (dependency): יחס סמנטי בין שני גורמים. משמעותו היא ששינוי כלשהו בעצם הבלתי תלוי, עשוי להשפיע על העצם התלוי בו. תלות מסומנת באמצעות חץ מקווקו.

חיבור (association): מתורגם לקשר בין שני גורמים. מסומן באמצעות חץ פשוט.

הכללה (generalization): זהו יחס הנקרא גם generalization/specialization כלומר הכללה\ייחוד. באופן אינטואיטיבי ניתן לחשוב על כך כעל "הורשה" של מחלקות, כאשר האב הוא הגורם המכליל (generalized), והבן הוא הגורם ה"מיוחד" (specialized). מסומן באמצעות חץ עם משולש בקצהו.

מימוש (realization): זהו קשר סמנטי, שמשמעותו גורם אשר פעולתו מבוצעת על ידי גורם אחר. בדרך כלל משתמשים ביחס זה בין ממשק (interface) לבין המחלקה המממשת אותו. מסומן באמצעות חץ מקווקו עם משולש בקצהו.

דיאגרמות[עריכת קוד מקור | עריכה]

הדיאגרמות מאגדות את גורמי היסוד והיחסים לגורמים בעלי משמעות.

הירארכיה של דיאגרמות UML 2.0, מוצגות כדיאגרמת מחלקה

ב-UML 2.0 יש שלושה עשר סוגים של דיאגרמות. כדי להבין את הסוגים השונים של הדיאגרמות נוח לחלק אותן לשלוש קבוצות הירארכיות כמו בתרשים משמאל.

דיאגרמות מבניות:

דיאגרמת מחלקה (class diagram) - מתארת את מחלקות התוכנה ואת היחסים ביניהן.

דיאגרמת רכיבים (Component diagram) - מתארת את רכיבי המערכת כדוגמת טבלאות וקבצים.

Composite structure diagram

דיאגרמת פריסה (Deployment diagram) - מתארת את פריסת המערכת באופן פיזי אצל הלקוח.

דיאגרמת אובייקט (Object diagram)

דיאגרמת חבילה (Packages diagram)

דיאגרמות התנהגותיות:

דיאגרמת פעילות - מתארת תהליכים המתרחשים בתוכנה בתגובה לפעולה של המשתמש.

דיאגרמת מצב (State diagram) - מתארת את העצם במצבים שונים.

דיאגרמת אופן שימוש (Use case diagram) - מתארת את השימוש במערכת.

דיאגרמות אינטרקציה:

דיאגרמת רצף (Sequence diagram) - מתארת קשר בין עצמי המערכת על ציר הזמן.

דיאגרמת שיתוף פעולה (Collaboration diagram) - דיאגרמה זו חלופית לדיאגרמת רצף.

דיאגרמת סקירת אינטרקציה (Interaction overview diagram)

דיאגרמת תזמון (Timing diagram)

בקורס זה אנו נעבוד על 4 סוגי תרשימים:

1. דיאגרמות מבניות-דיאגרמת מחלקה (class diagram) - מתארת את מחלקות התוכנה ואת היחסים ביניהן.
2. דיאגרמות התנהגותיות: דיאגרמת פעילות - מתארת תהליכים המתרחשים בתוכנה בתגובה לפעולה של המשתמש.
3. דיאגרמות התנהגותיות: דיאגרמת מצב (State diagram) - מתארת את העצם במצבים שונים.
4. דיאגרמות התנהגותיות: דיאגרמת אופן שימוש (Use case diagram) - מתארת את השימוש במערכת.

**דיאגרמת מחלקה**

**דיאגרמת מחלקה, בשפת UML, היא סוג של תרשים סטטי המתאר את מבנה המערכת על ידי הצגת מחלוקתיה, תכונותיהם והקשרים בין המחלקות.**

רשים המחלקה הוא אבן הבניין המרכזית במידול מונחה עצמים. התרשים משמש לשתי מטרות עיקריות והן

מידול כללי-תפישתי של רכיבי האפליקציה

מידול מפורט למען תרגום המודל

המחלקות בדיאגרמת המחלקה מייצגות גם את האובייקטים העיקריים וגם את היחסים בין האפליקציה ובין האובייקטים אשר יומרו לקוד מקור. תרשים המחלקה מציג את המחלקה בתיבות אשר כוללות שלושה חלקים:

חלקה העליון של התיבה יכלול את השם המחלקה

חלקה האמצעי של התיבה יכלול את מאפייני המחלקה

חלקה התחתון של התיבה מכיל את המתודות או הפעולות אשר המחלקה יכולה לבצע

BankAccount1.svg

בתרשים עיצוב מערכת, המחלקות מאוחדות יחדיו בתרשים מחלקה אשר מסייע לקבוע את הקשרים הסטטיים בין האובייקטים.

מרכיבים:UML מספק מנגנונים לייצוג מרכיבי מחלקה כגון : מאפיינים, מתודות ומידע נוסף לגביהם.

ראות:כדי לציין את רמת הראות (Visibility) של מרכיב מחלקה (למשל: מאפיין או מתודה) ישנן דרכי ציון אשר יופיעו לפני שם המרכיב:

(+) Public

(#) Protected

(-) Private

(~) Package

יחסים בין מחלקות:היחסים הקיימים בדיאגרמת מחלקה הינם:

Association - יחס כלשהו בין מחלקות. המספרים בקצוות מציינים את מספר (או טווח) העצמים בכל צד של היחס (Multiplicity).

Inheritance - ירושה ממחלקת בסיס.

Realization / Implementation - מימוש ממשק.

Dependency - תלות של מחלקה אחת במחלקה אחרת.

Aggregation - יחס הכלה חלש: מחלקה מכילה מצביע לעצם, כאשר העצם המוצבע יכול להתקיים ללא המחלקה המכילה.

Composite Aggregation - יחס הכלה חזק: מחלקה מכילה עצם, כאשר העצם המוכל מתקיים רק עם העצם החיצוני.

טווח:שפת UML מגדירה שני סוגי טווחים (Scope) עבור מרכיבי המחלקה: מופע ומסווג (Classifier). במקרה של מרכיב מסוג מופע, הטווח הוא המופע הספציפי. עבור מאפייני המחלקה, הכוונה היא כי עריכתם משתנה בין כל מופע למופע. עבור שגרות הכוונה היא שהכניסה אליהן תלויה במצבן הנוכחי. לעומת זאת, במקרה של מרכיב מסוג מסווג, הטווח הוא המחלקה. עבור המתודות, כניסה אליהן אינה משפיעה או מושפעת ממצב האובייקטים. מרכיבי המסווג לעתים מוכרים כמשתנים סטטיים בשפות תכנות רבות. כדי להצהיר על כך שמרכיב מסוים מקבל טווח מסווג שמות יוצג עם קו תחתון. אחרת, המחלקה תהיה בטווח מופע.

**דיאגרמת אופן השימוש**

**דיאגרמת אופן שימוש בשפת המידול האחודה (UML) היא סוג של דיאגרמה התנהגותית אשר מורכבת מניתוח תרחישי שימוש. מטרת הדיאגרמה היא הצגת רקע פונקציונאלי של המערכת עם שימוש במושגים : שחקנים, מטרה (אשר תתואר כתרחיש שימוש) והתלות בין תרחישי השימוש.**

**המטרה הכללית של דיאגרמת אופן השימוש היא הצגת הפעולות המבוצעות על ידי כל שחקן. תפקידי השחקן ישורטטו בתרשים.**

**דיאגרמת אופן השימוש מופיעה באופן רשמי בשתי שפות מידול אשר יוצאו על ידי ארגון OMG - שפת המידול המאוחדת (UML) ושפת מידול המערכת (SysML).**

תרגול 15.5.17

בניית דיאגרמת use case:

שחקנים:

* משתמש
* משתמש המחפש קשר
* משתמש המחפש עבודה
* משתמש המחפש עובדים
* משתמש מנהל קבוצה
* משתמש חבר בקבוצה

שירותים:

* שליחת הודעה
* צפייה בתוצאות חיפוש משתמשים
* צפייה באנשים שעשוי להכיר
* צפייה בתוצאות משרות רלוונטיות
* רישום להצעת עבודה
* פרסום משרה
* צפייה במועמדים לעבודה
* צפייה בתוצאות משתמשים לעבודה
* שליחת הצעת עבודה
* יצירת קבוצה
* אישור בקשות(פרסום וחבר בקבוצה)
* שליחת בקשת הצטרפות

בניית דיאגרמה class:

מחלקות:

1. User
   1. אימייל
   2. שם מלא
   3. שם פרופיל
   4. סיסמא
   5. עבודה
   6. שם החברה
   7. סוג תעשייה(כמה ערכים)
   8. שם מוסד לימודים אחרון
   9. תחילת לימודים
   10. סיום לימודים
   11. גיל
   12. תמונה
   13. קורות חיים
   14. אתר אישי
   15. לינק לפרופיל
   16. מערך חברים
   17. מערך הודעות
   18. מערך מיומנויות
   19. מערך תחומי עניין
   20. מערך המלצות
2. Massege
   1. מס' הודעה
   2. תאריך
   3. שעה
   4. אימייל מקור
   5. אימייל יעד
   6. תוכן
   7. מדיה
3. Post
   1. מס' פוסט
   2. תאריך
   3. שעה
   4. תוכן
   5. מדיה
4. Group
   1. אימייל מנהל קבוצה
   2. מערך משתמשים חברים
   3. מערך פוסטים
5. Job
   1. אימייל מפרסם
   2. שם העבודה
   3. פירוט העבודה
   4. תאריך
   5. שעה
6. Advertising
   1. אימייל מפרסם
   2. לינק לאתר פרסום
   3. תוכן
   4. מדיה
   5. תאריך
   6. שעה
7. מיומנויות(קבועים מראש)
8. תחומי עניין(קבועים מראש)
9. המלצות
   1. אימייל ממליץ
   2. אימייל מומלץ
   3. מערך מיומנויות
   4. תוכן
   5. תאריך
   6. שעה

הרצאה 18.5.17

דיאגרמת אופן הפעילות:

צמתים מורכבים מפעילות אובייקטים ובקרה

חצים מורכבים מבקרת הזרימה(מתן שם, סלקטור, הגבלת משקל) וזרימת אובייקטים

פעולות:

Call action = פעולה הגורמת לפלט

= פעולה הגורמת לפעולה אחרת או להכנסת מידע.

= פעולה שגורמת לאירוע מותנה בקלט.

אובייקטים:

צומת של אובייקט= סוג מחלקה.

בקרה:

Fork= פיצול, עבור פיצול של פעילות.

Join = איחוד, עבור איחוד של פעולות שמביאות לפעולה אחרת.

איך לבנות את הדיאגרמה?

1. לזהות את הפעלים/פעולות בפעילות במערכת.
2. הוספת תחילת פעולה וסוף פעולה.
3. לזהות את התנאים בפעילות.

בניית דיאגרמות:

Use case = נדרש מינימום 3

Class = חייב את כל המחלקות שבמערכת.

Activity= לפי ה-dfd רמה 0.