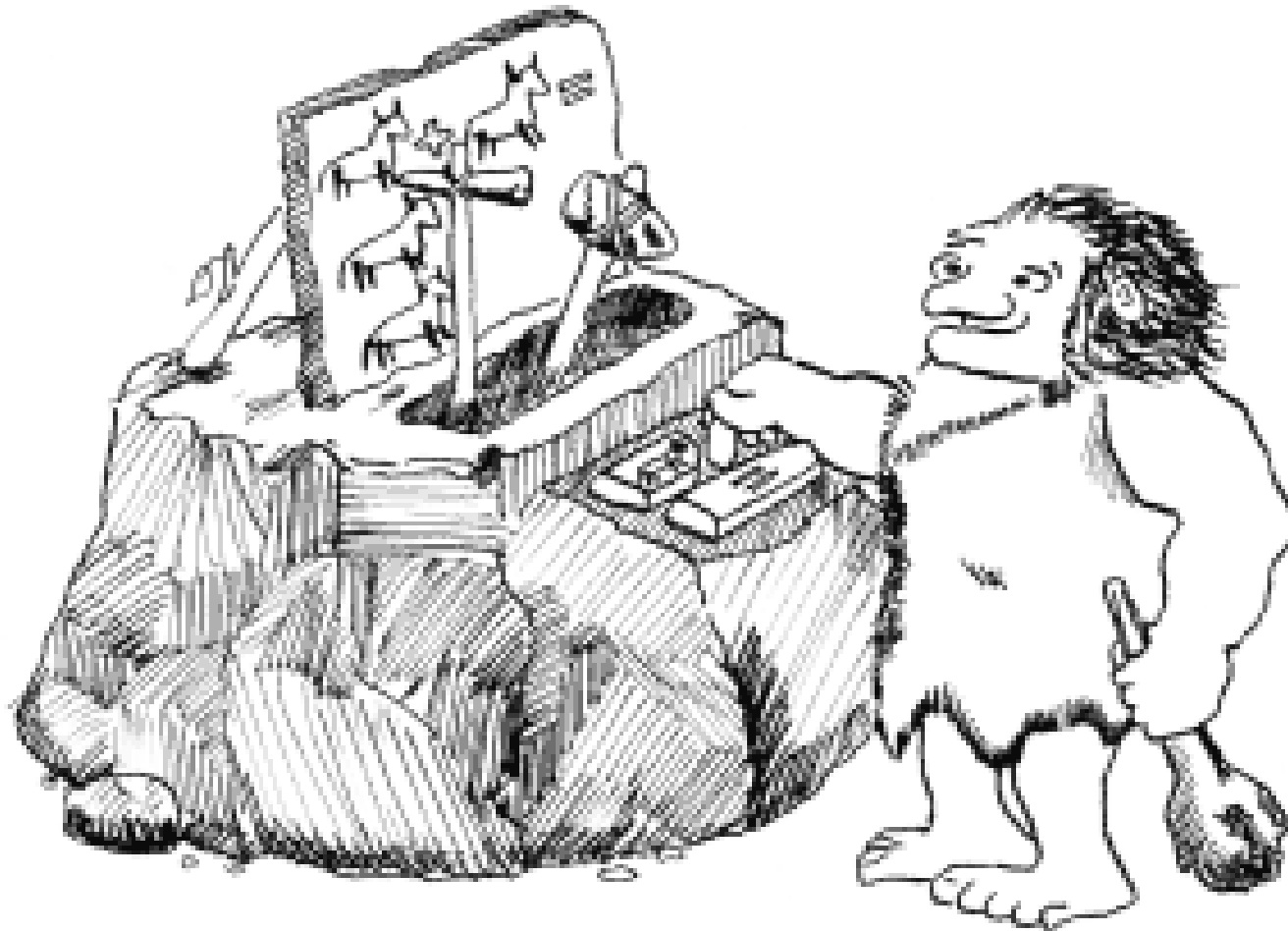


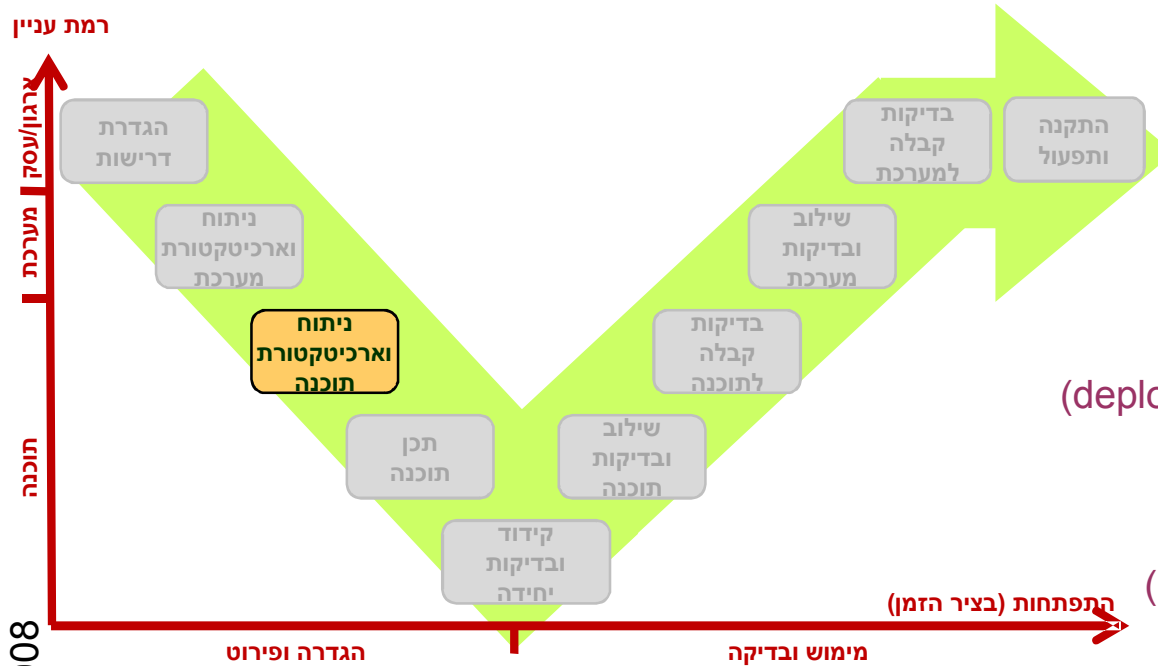
ניתוח וארכיטקטורת תוכנה

# Software Requirements Analysis and Architecture



© ד"ר עמיר תומר, 1999-2008

# פעילות ניתוח וארכיטקטורת התוכנה



© 1999-2008, ד"ר עמיר תומר

## • קלט

- ארכיטקטורת המערכת
  - מודל פריסה (deployment model)
- מפרט תהליכי המערכת
  - use case model
  - מודל פעילות (activity model)
- טבלת דרישות מעודכנת

## • תוצרים עיקריים

- ארכיטקטורת התוכנה
  - מודל רכיבים (component model)
- מפרט תהליכי התוכנה
  - use case model – לכל רכיב בנפרד

# תוכן העניינים

- הגדרת דרישות התוכנה
- ניתוח דרישות וארכיטקטורת תוכנה
- ארכיטקטורת התוכנה
- תהליכי התוכנה
- מבנה וארגון התוכנה
- המודל הניהולי – ארגון חבילות העבודה

# יצירת טבלת דרישות לתוכנה

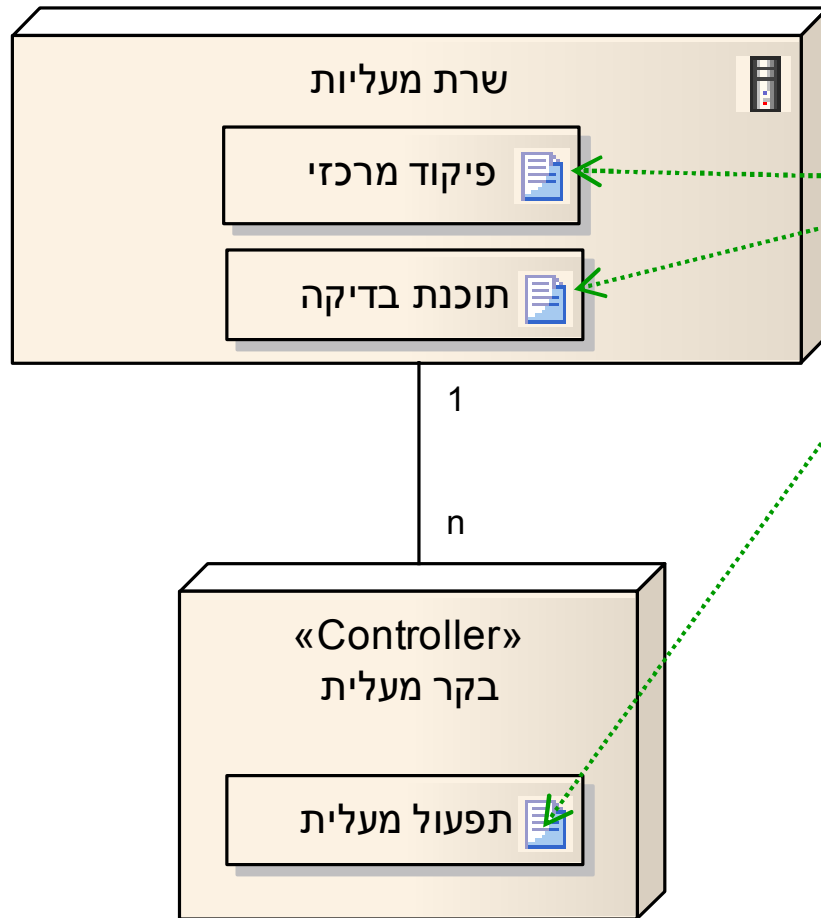
- אפשרות א' – הכנסת דרישות נגזרות לטבלת דרישות המערכת
  - דרישות שהתבהרו או פורטו במהלך הניתוח
    - לדוגמה: כאשר המעלית הגיע לקצה המסלול, או שאין יותר בקשות באותו כיוון, היא הופכת את כיוון הנסיעה
  - דרישות פונקציונליות שנועדו לממש דרישות לא-פונקציונליות
    - לדוגמה: פירוט פרוצדורת הבדיקה
- אפשרות ב' – הכנסת מפרט הדרישות (Use Cases) לטבלה חדשה
  - כל תרחיש מהווה דרישה (MSS, כל חלופה, כל חריגה)
    - מאחר ומפרט הדרישות מהווה כיסוי מלא של כל הדרישות התפעוליות ניתן להחליף אותן בדרישות התרחישים
  - כל הדרישות שאין להן כיסוי ב- Use Cases יועברו לטבלה החדשה כפי שהן

# תוכן העניינים

---

- הגדרת דרישות התוכנה
- ניתוח דרישות וארכיטקטורת תוכנה
- ארכיטקטורת התוכנה
- תהליכי התוכנה

# מערכת המעליות – תרשים פריסה – תזכורת



פריטי תוכנה  
(software artifact)

- פריט תוכנה (Software Artifact) – מרכיב פיזי של התוכנה שנוצר בתהליך הפיתוח

# מודל רכיבים (Component Model)

- רכיב (Component)

- חלק מודולרי של התכן המסתיר את מימושו מאחורי קבוצת ממשקים חיצוניים.
- בתוך מערכת, רכיבים המספקים ממשקים ניתנים להחלפה זה בזה באופן חופשי

- מרכיבי המודל

- רכיבים

- שרותים פנימיים
- ספריות
- Plug-ins

- התלויות והממשקים בין רכיבים

# הקשר בין רכיבי תוכנה לפריטי תוכנה

- רכיבי תוכנה (software components)

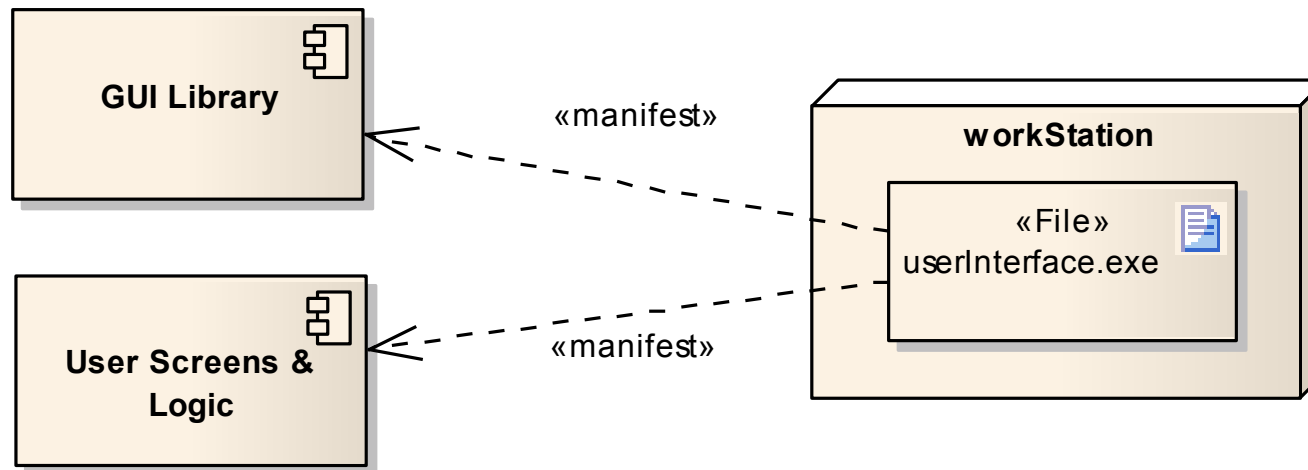
– המרכיבים מהם יוצרים את התוכנה בסביבת הפיתוח

- חלקי קוד
- ספריות ותוכנות מדף
- אובייקטים מקושרים דינמית (DLL)

- פריטי תוכנה (software artifacts)

– פריטים המותקנים בסביבת הריצה ונחוצים לפעולה תקינה של המערכת

- פריטי תוכנה משקפים (<<manifest>>) רכיבי תוכנה

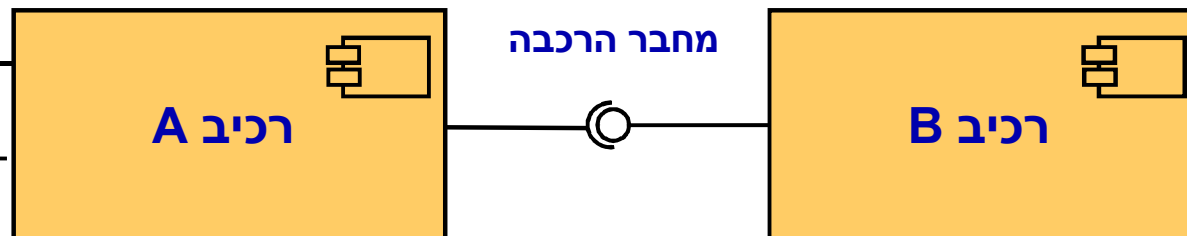


# ממשקים (Interfaces)

- רכיב הוא "קופסה שחורה" המקיים קשרים עם רכיבים אחרים באמצעות ממשקים
- ממשקים מסופקים (Provided Interfaces)
  - הממשקים אותם רכיב חושף לסביבתו, ובאמצעותם ניתן לקבל את שירותיו
    - המימוש נמצא ברכיב עצמו או בכל אלמנט אחר המממש אותו (כאשר הוא אבסטרקטי)
- ממשקים נדרשים (Required Interfaces)
  - הממשקים שהרכיב צריך מרכיבים אחרים על מנת לאפשר לו לספק את מלוא הפונקציונליות שלו
    - הרכיב יכול להשתמש בממשקים (ע"י פניה לרכיבים אחרים)
    - הרכיב יכול להגדיר כיצד רכיבים אחרים יספקו לו את הממשק
- מחבר הרכבה (Assembly Connector)
  - חיבור ישיר בין שני רכיבים, כאשר אחד דורש ממשק והשני מספק אותו

ממשק מסופק

ממשק נדרש



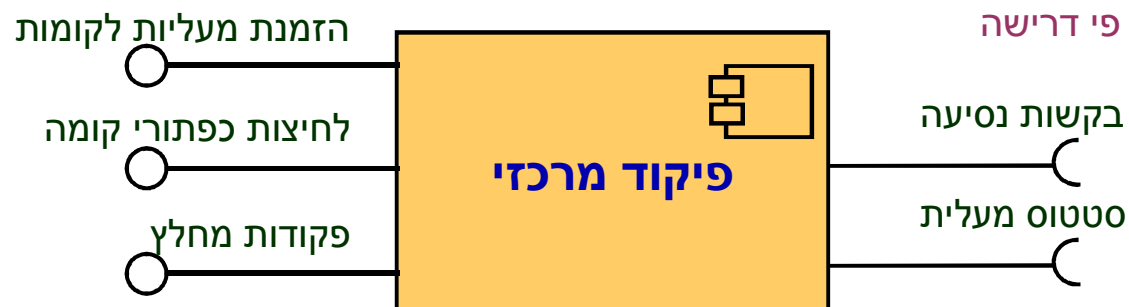
# מערכת המעליות – רכיב תפעול מעלית

- תוכנת הבקר של מעלית אחת
- ממשקים מסופקים
  - לחיצות כפתורי מעלית
    - ממשק לקליטת בקשות נסיעה המגיעות מכפתורי המעלית
    - בקשות נסיעה
      - ממשק לקליטת בקשות נסיעה שלא באמצעות כפתורים
      - סטטוס מעלית
        - ממשק להספקת קוד סטטוס על פי בקשה
  - ממשקים נדרשים
    - אין



# מערכת המעליות – רכיב פיקוד מרכזי

- תוכנת השליטה על מערכת המעליות
- ממשקים מסופקים
  - לחיצות כפתורי קומה
    - ממשק לקליטת הזמנות נסיעה המגיעות מכפתורי קומות
  - הזמנת מעליות לקומות
    - ממשק לקליטת הזמנות נסיעה שלא באמצעות כפתורים
  - פקודות מחלץ
    - ממשק לקליטת פקודות הפעלה באמצעות מפתח/כפתורים יעודיים
- ממשקים נדרשים
  - בקשות נסיעה
    - הפניית בקשות נסיעה למעלית
  - סטטוס מעלית
    - קבלת סטטוס מעלית על פי דרישה

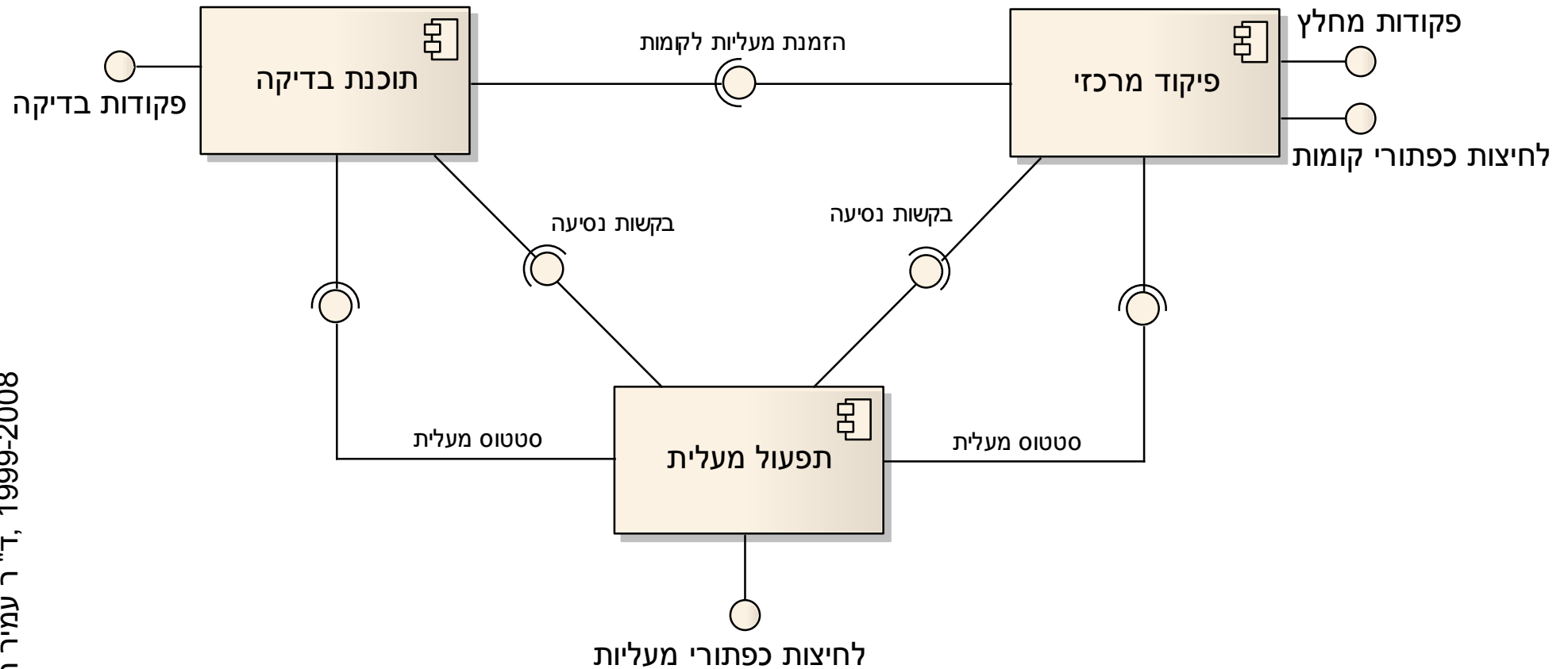


# מערכת המעליות – רכיב תוכנת בדיקה

- התוכנה שבשימוש הטכנאי
- ממשקים מסופקים
  - פקודות בדיקה
    - ממשק-משתמש לטכנאי
  - ממשקים נדרשים
    - הזמנת מעליות לקומות
      - הפניית הזמנות נסיעה לקומות
    - בקשות נסיעה
      - הפניית בקשות נסיעה למעלית
    - סטטוס מעלית
      - קבלת סטטוס מעלית על פי דרישה



# מערכת המעליות – תרשים רכיבים (Component Diagram)



# תוכן העניינים

---

- הגדרת דרישות התוכנה
- ניתוח דרישות וארכיטקטורת תוכנה
- ארכיטקטורת התוכנה
- תהליכי התוכנה

# תהליכי תוכנה

- Use Case Diagram ברמת כל רכיב תוכנה

- רכיב אחד הוא שחקן של רכיב אחר

- מפרט Use Case

- טקסטואלי

- באמצעות תרשים פעילות (Activity Diagram)

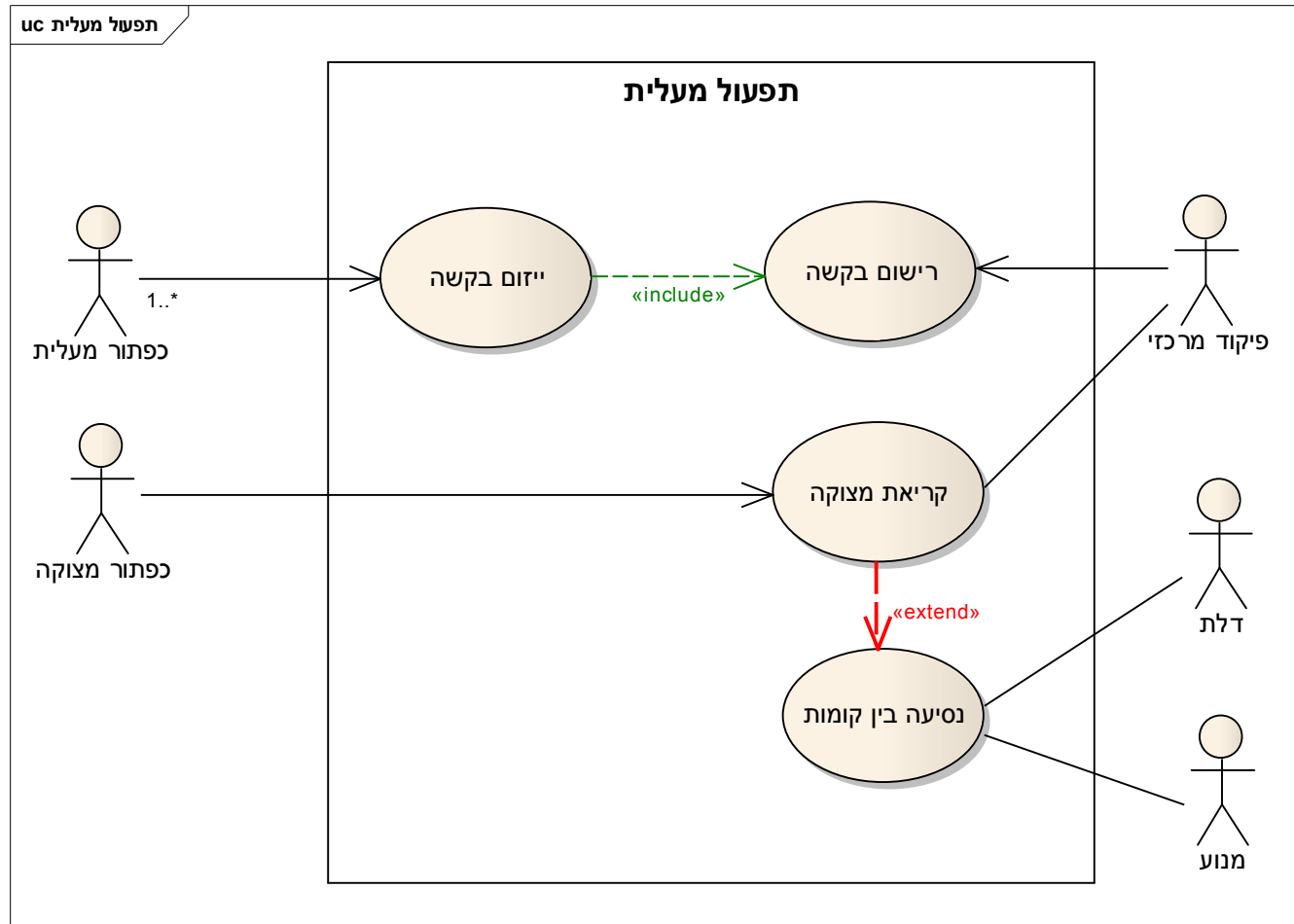
- מימוש (Use Case Realization)

- האופן בו ימומשו התהליכים בקוד

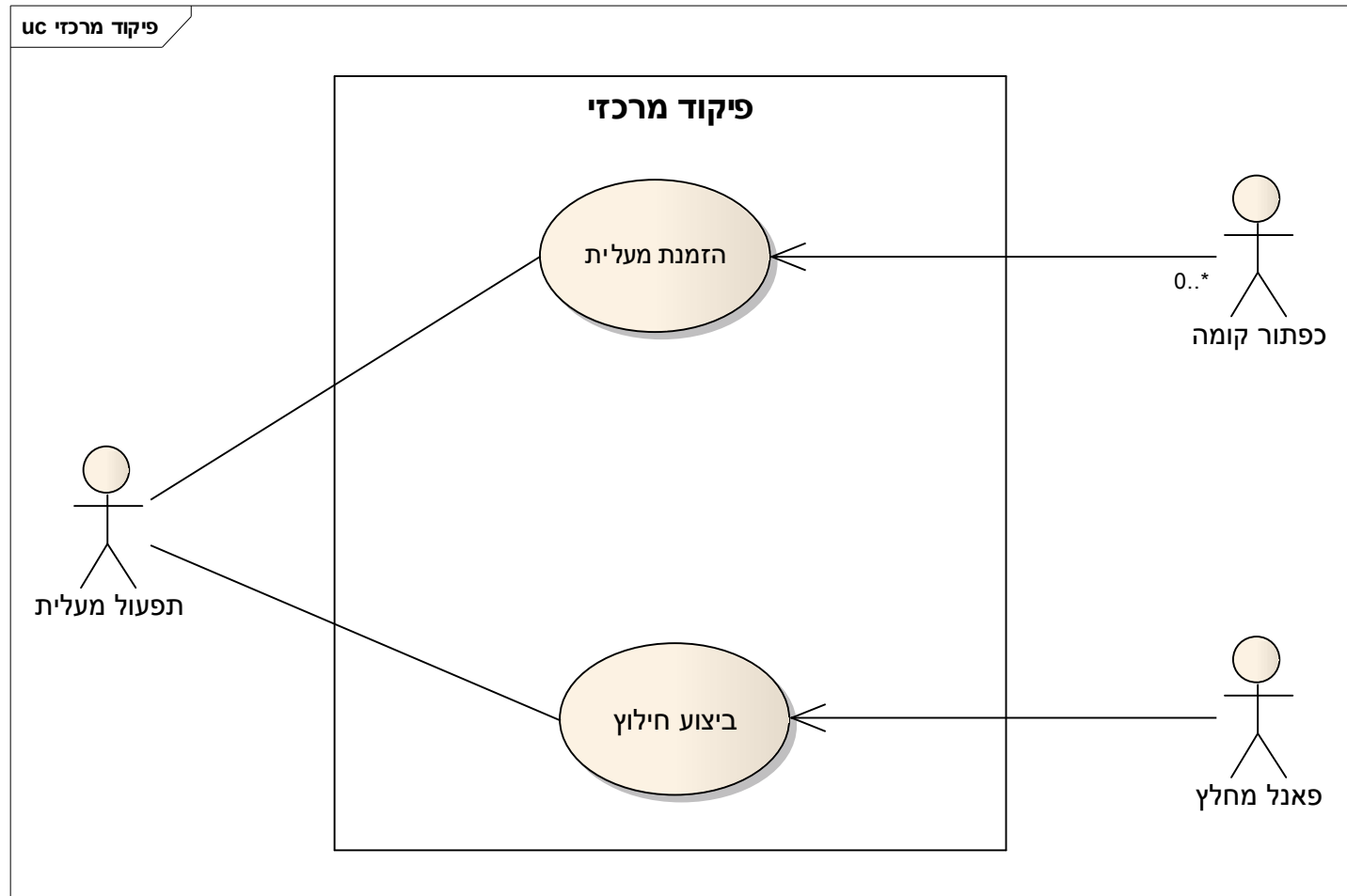
- אופן מימוש = תכן

- תלוי בפרדיגמה התכנותית (תוכנה מבנית, תוכנה מונחית-עצמים וכו')

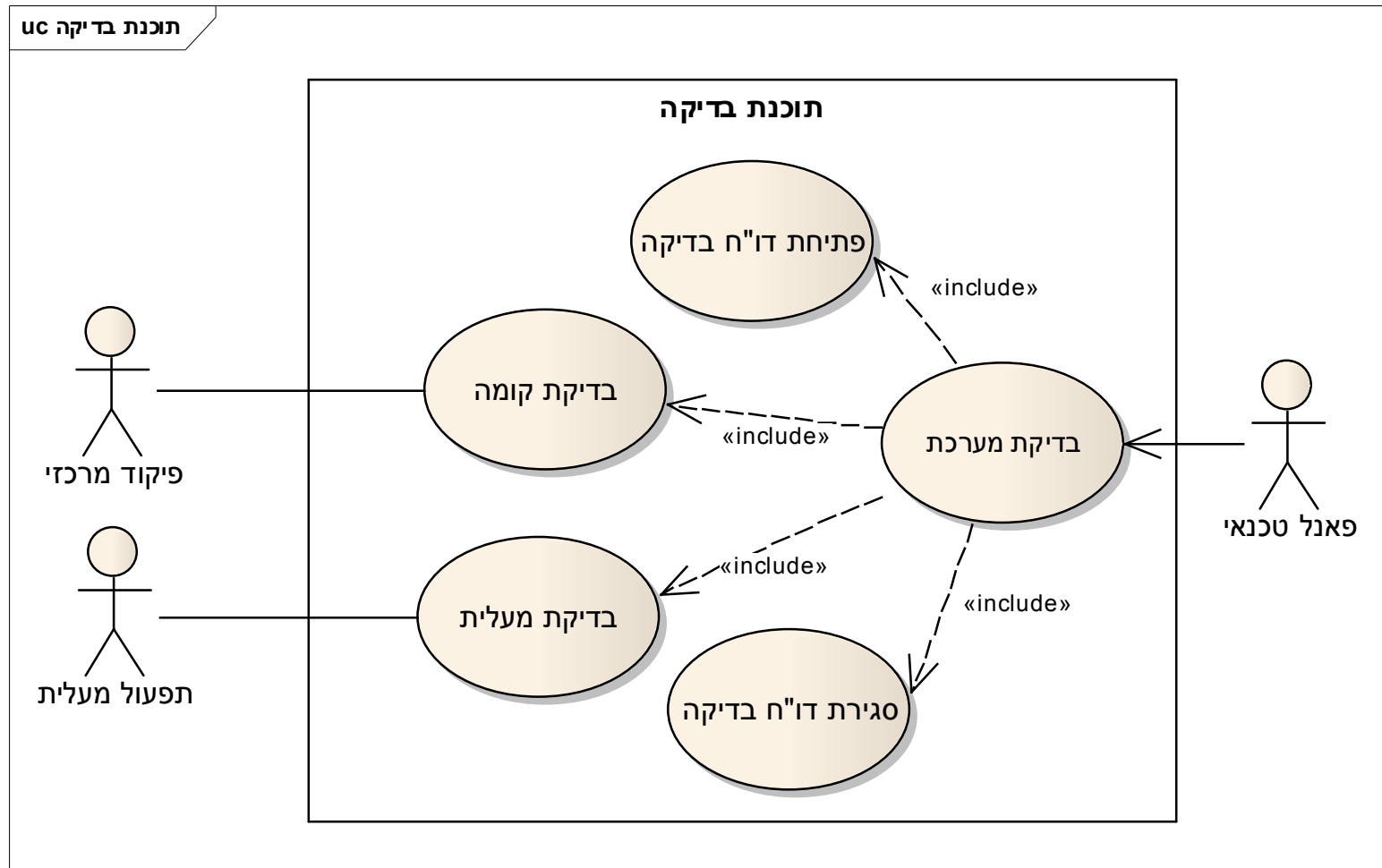
# UCD – תוכנת תפעול מעלית



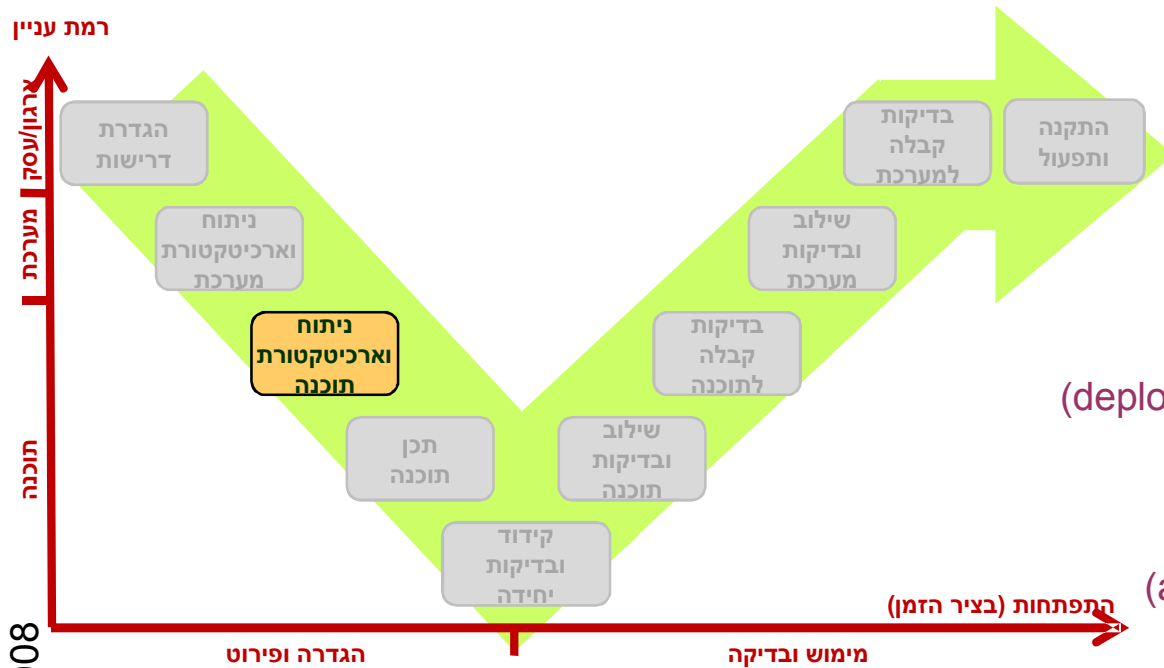
# UCD – תוכנת פיקוד מרכזי



# UCD – תוכנת בדיקה



# פעילות ניתוח וארכיטקטורת התוכנה - סיכום



© 1999-2008, ד"ר עמיר תומר

- קלט
  - ארכיטקטורת המערכת
    - מודל פריסה (deployment model)
  - מפרט תהליכי המערכת
    - use case model
    - מודל פעילות (activity model)
  - טבלת דרישות מעודכנת
- תוצרים עיקריים
  - ארכיטקטורת התוכנה
    - מודל רכיבים (component model)
  - מפרט תהליכי התוכנה
    - use case model – לכל רכיב בנפרד

## הגדרת דרישות התוכנה

- יש להקים טבלת דרישות תוכנה ע"י תוספת דרישות בטבלת דרישות המערכת או כטבלה חדשה על בסיס מפרט ה- Use Cases

## ניתוח דרישות וארכיטקטורת תוכנה

- ארכיטקטורת התוכנה
  - בניית מודל רכיבים המשקף את רכיבי התוכנה וממשיקהם
- תהליכי התוכנה
  - בניית מודל Use Cases עבור כל רכיב תוכנה