

סוגי הקשרים ב UML

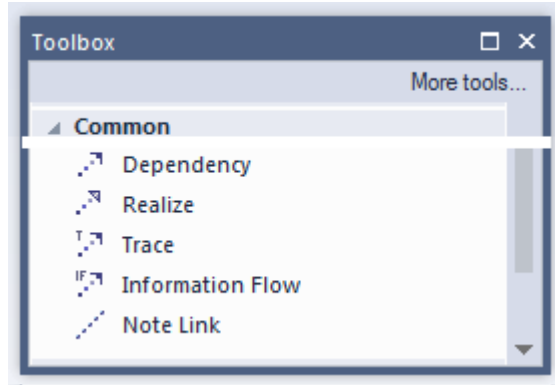
מחבר: איציק סיון

1. מבוא
2. קשרים משותפים (Common)
 - Dependency
 - Realization
 - Information Flow
 - Trace
 - Note Link
3. היוזואליות של הקשר (Line Style)
4. קשרים יחודיים
 - Requirement Diagram
 - Class Diagram
 - Use Case Diagram
 - Activity Diagram
 - Package Diagram

1. מבוא
 - UML תומך במספר רב של סוגי אלמנטים מטבע הדברים האלמנטים קשורים ביניהם בצורות שונות על פי הצורך. על מנת לשלוט ב UML חייבים להבין את המשמעות של הקשרים השונים כל קשר הוא אלמנט (אובייקט) בפני עצמו ויש לו Properties. קיימות שתי קבוצות של קשרים. הקבוצה הראשונה כולל 5 קשרים משותפים לכלל האלמנטים. קבוצה זו מופיעה בכל ארגזי הכלים של התרשימים השונים הקבוצה השניה הינה הקשרים הייחודיים לכל סוג של תרשים. הקשר מאויר פיסית על ידי קו מקשר עם צורות שונות של חץ מסיים וצורה של קו. לפי הצורך ניתן לפרט את מהות הקשר על ידי Stereotype מתאים ומידע נוסף בשדה ה Notes

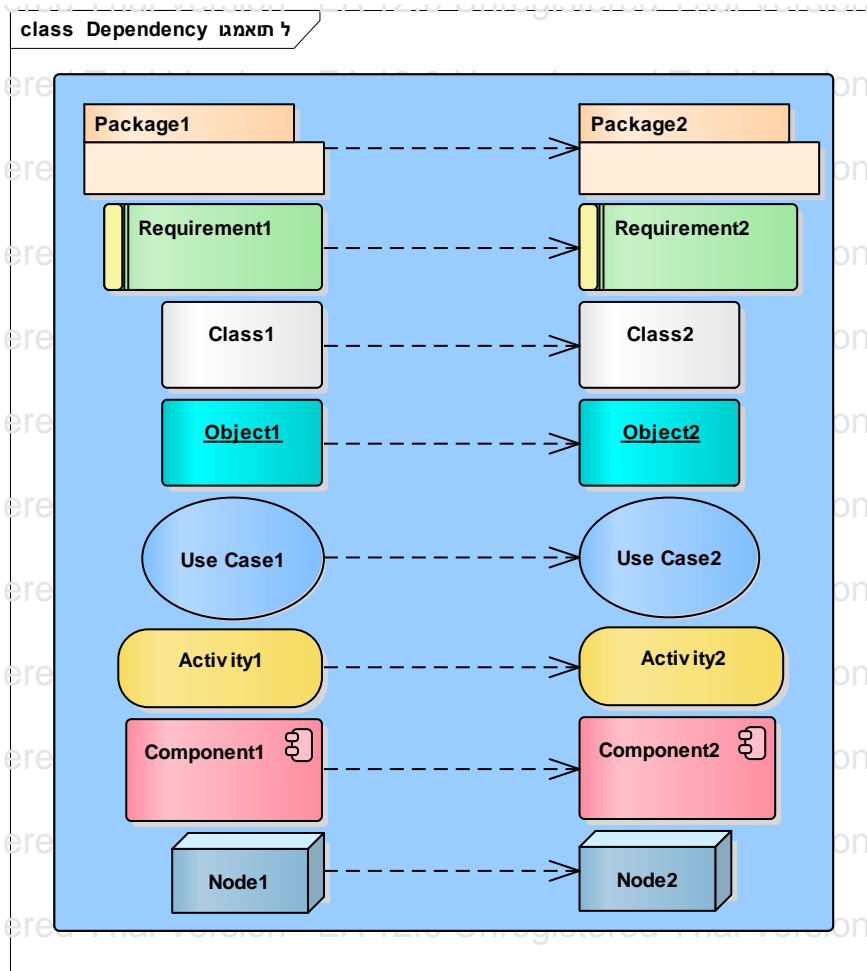
2. קשרים משותפים (Common)

בכל Tool Box יש רשימה תקנית של 5 סוגי קשרים שרלוונטיים לכל סוגי האלמנטים



Dependency 2.1

תלות מכוונת: זהו הקשר הנפוץ ביותר בין שני אלמנטים דוגמאות:

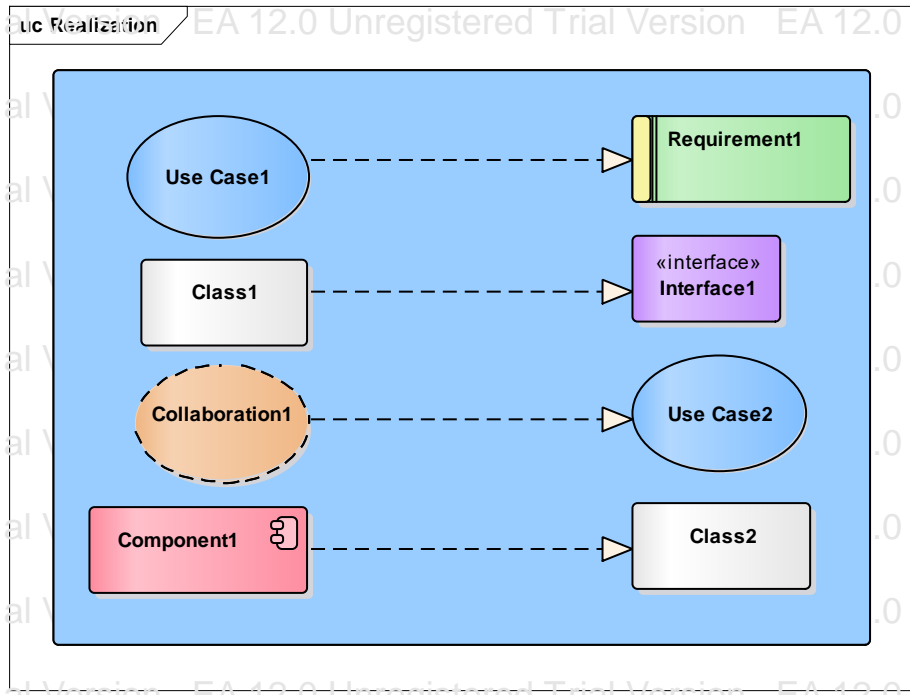


אלמנט A מצד שמאל תלוי/זקוק/דורש את אלמנט B מצד ימין.
 אלמנט A זקוק לאלמנט B לצורך הבנתו. ללא B לא ניתן להבין את המשמעות התחבירית המלאה של A או המבנה של A (תלוי בסוג התרשים).
 ניתן לומר שבין A ל B מתקיימים קשרי ספק לקוח. B הוא ספק A הוא לקוח.
 A אינו שלם ללא B
 אלמנט B אינו תלוי באלמנט A.

בשלב הראשונים של הניתוח / העיצוב אם קיים קשר כלשהו בין שני אלמנטים, ניתן להתחיל עם קשר מסוג Dependency ורק בשלבים מתקדמים יותר לפרט את המהות המדויקת של הקשר על ידי שימוש ב סטראוטיפים.

2.2-Realization-מימוש

דוגמאות:



האלמנט A (client) מצד שמאל מבצע מימוש (Realization) לאלמנט B מצד ימין.

A ו B הם בדרך כלל אלמנטים מסוגים שונים

המשמעות הינה בהתאם לקונטקסט של התרשים.

Use Case1 מממש את Requirement1, כלומר מפרט אותו לעומק

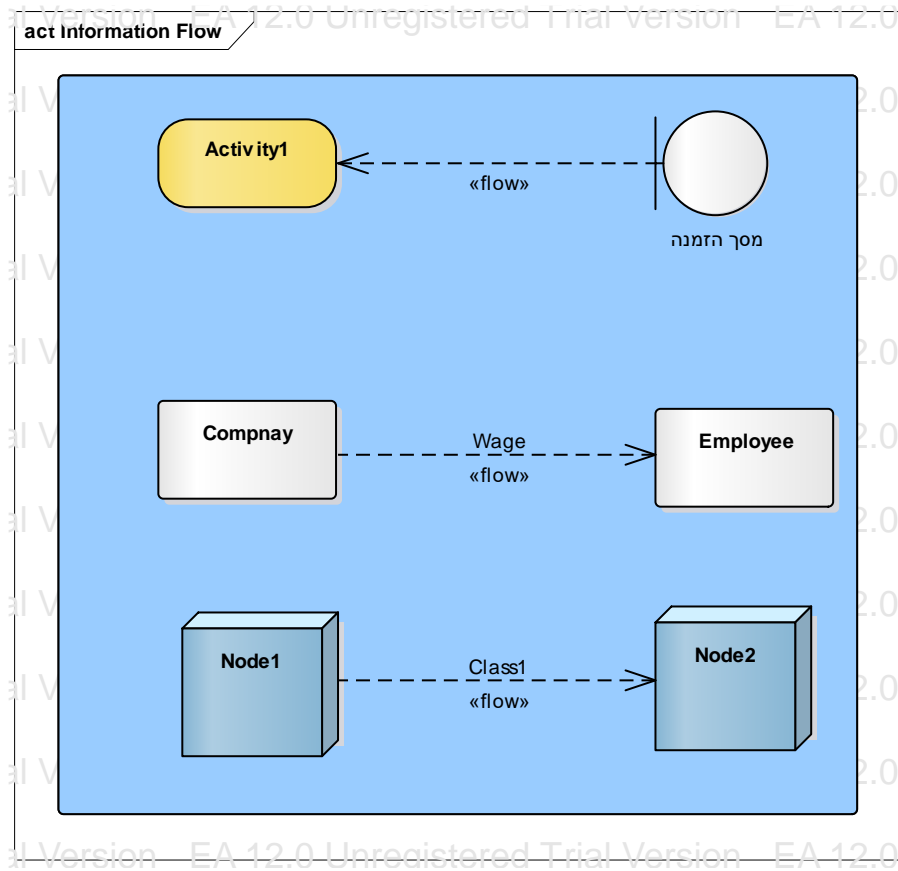
Class1 מממש את החוזה שמוגדר ב 'Interface1', כלומר כולל את פירוט ה Operations

שמצויינים ב Interface1

Collaboration1 מממש את UseCase2 על ידי הצגת התהליך באמצעות Sequence

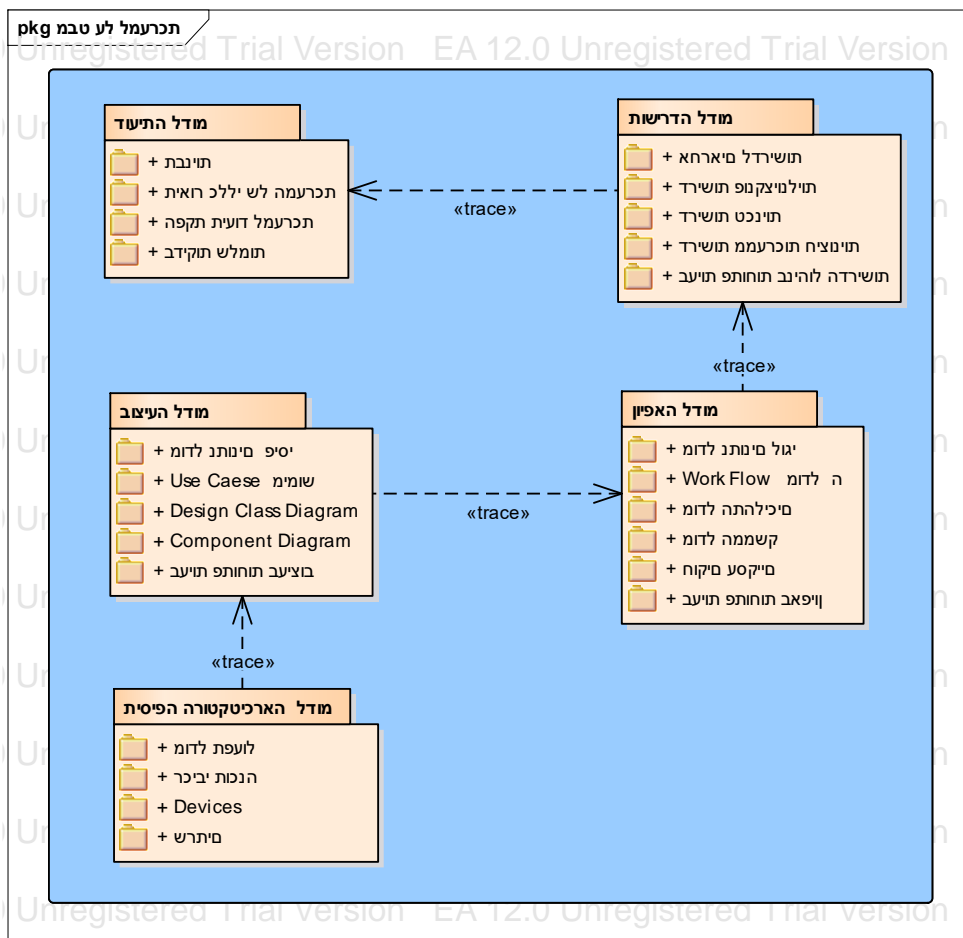
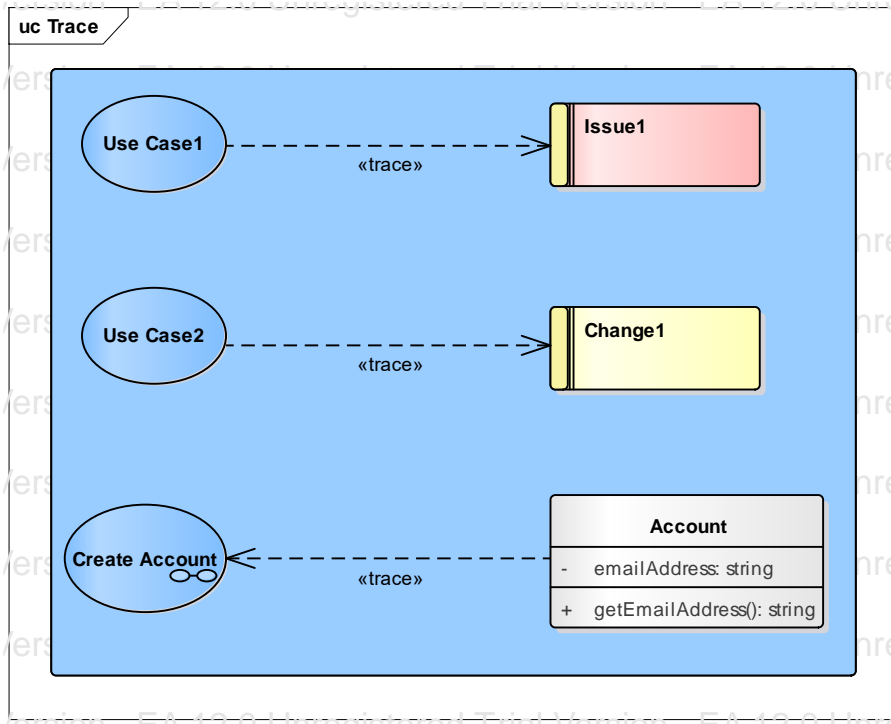
Diagram

קשר זה מייצג זרימה של Information Item בין שני אלמנטים. האינפורמציה זורמת בכיוון החץ. דוגמא:

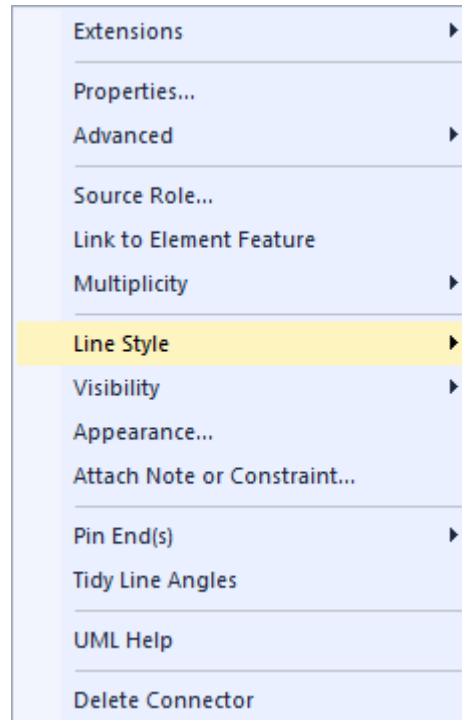


Trace 2.4

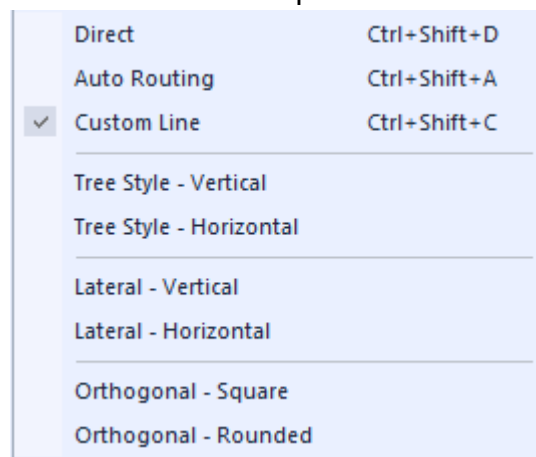
זוהי תלות היסטורית או תהליכית בין שני אלמנטים. אין חוקים ספציפיים למעבר בין האלמנטים. דוגמאות:



לקשר בין שני אלמנטים ניתן לקבוע סגנון רצוי על ידי ה Line Style



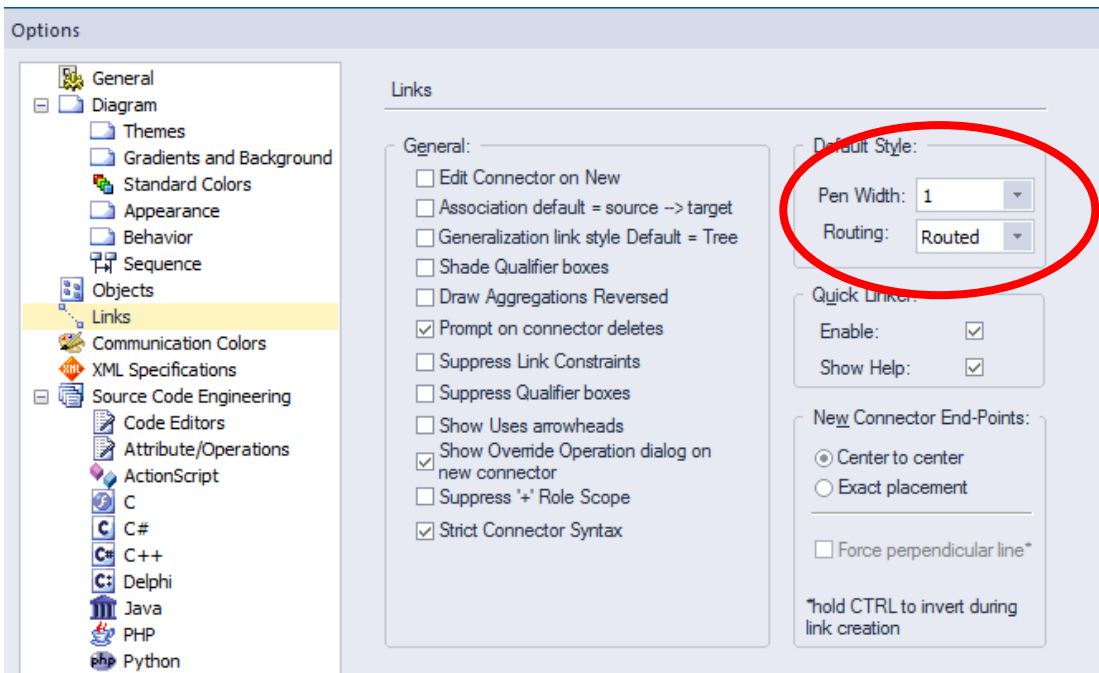
האפשרויות של הסגנון הינם:



ברירת המחדל של התוכנה הינה Custom: , כלומר המשתמש קובע את הסגנון. הסגנון המומלץ הינו קווים ישרי זווית.

כדאי לשנות את ברירת המחדל יש לבצע כדלקמן:

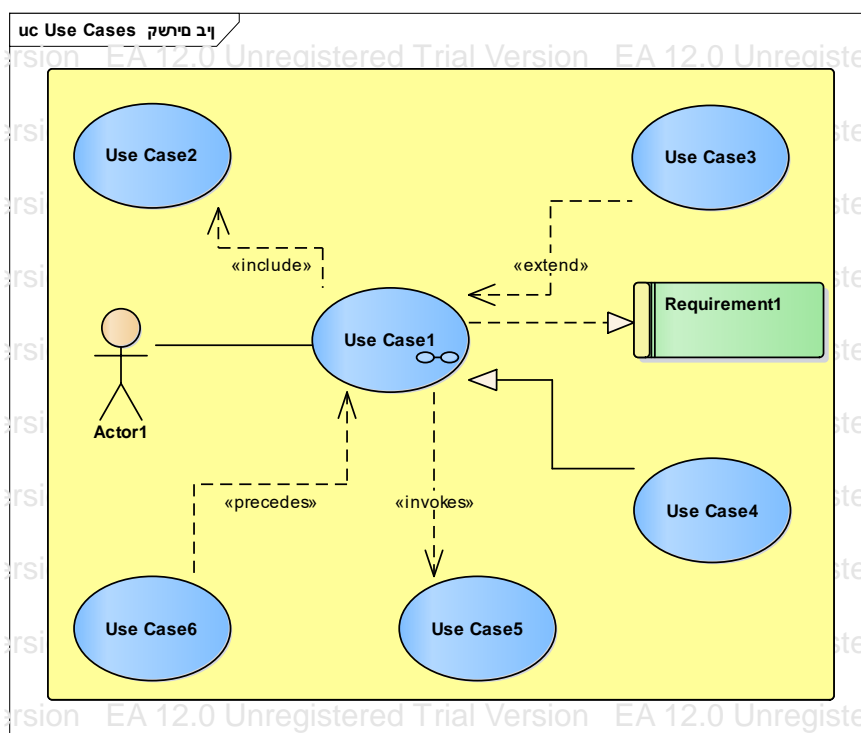
Tools→Options לחפש את Links ולאחר מכן ב default style לקבוע Routed



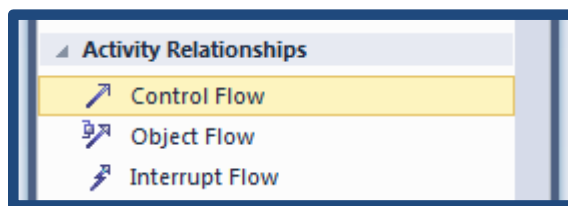
4. קשרים ייחודיים

לכל סוג של תרשים יש רשימה ייחודית של קשרים לדוגמא בתרשים מסוג Use Case, הרשימה כוללת 8 קשרים



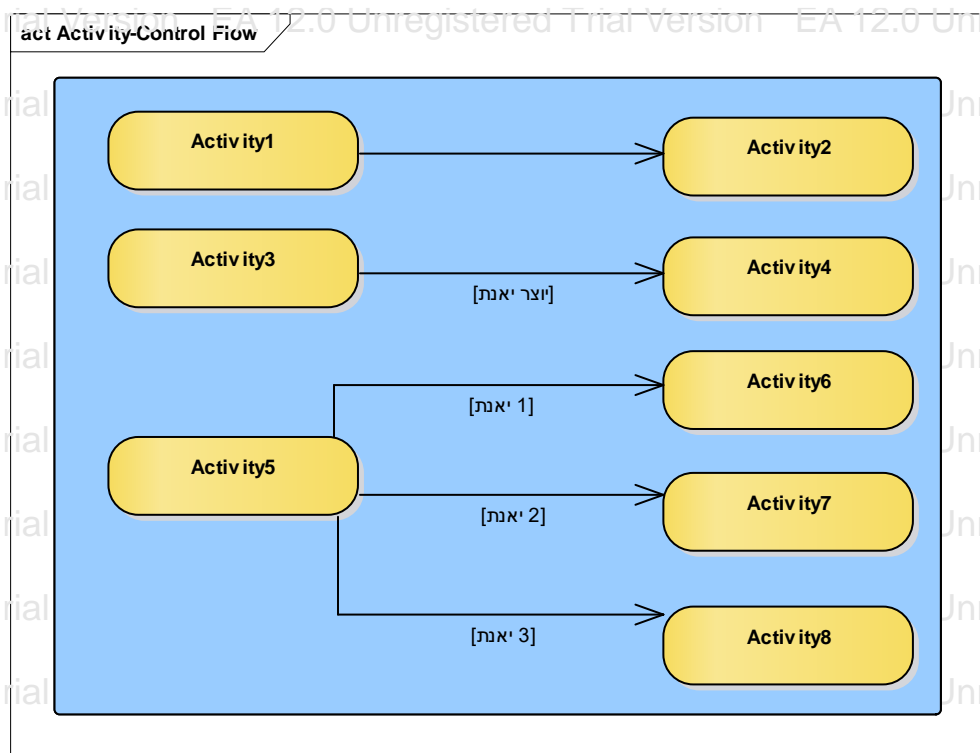


אלמנט א	אלמנט ב	סוג הקשר	הסבר
Actor1	Use Case1	Associate	Actor1 מבצע אינטראקציה עם Use Case1. הקשר הוא למעשה דו צדדי, כיון שברב המקרים השחקן מנהל דו שיח עם התהליך באמצעות מלוי שדות במסך או הקשה על פקדים שונים
Use Case 1	Use Case2	Include	Use Case1 מפעיל תמיד את Use Case2. למעשה Use Case2 היה צריך להיות חלק אינטגרלי של Use Case1, אבל הוא מוצג בנפרד כיון שהוא יכול לשרת UC נוספים או ש UC1 גדול מידי והוחלט לפצל אותו.
Use Case3	Use Case1	Extend	Use Case3 מרחיב את Use Case1. תת התהליך Use Case3 מופעל מתוך Use Case1 רק במקרים מסויימים. יש לשים לב! כיוון הקשר של Include הפוך מזה של Extend
Use Case 1	Requirement1	Realization	Use Case1 מממש את דרישה Requirement1. כלומר, דרישה 1 מוצגת ברמת על ואילו UC1 מפרט את הדרשיה לעומק, בדרך כלל באמצעות Activity Diagram
Use Case4	Use Case1	Generalization	Use Case4 יורש מ Use Case1. המשמעות היא ש Use Case4 הוא סוג של Use Case1 עם מספר הבדלים קטן. כאשר מפרטים את Use Case4 צריך למעשה לפרט רק את ההבדלים
Use Case1	Use Case5	Invokes	Use Case1 מפעיל/מעורר בשלב מסוים את Use Case5. Use Case1 ממשיך לבצע את הפעילויות שלו מבלי לצפות לתשובה מ Use Case5
Use Case6	Use Case1	precedes	Use Case6 חייב להתבצע לפני Use Case1



להלן מספר דוגמאות של Control Flow

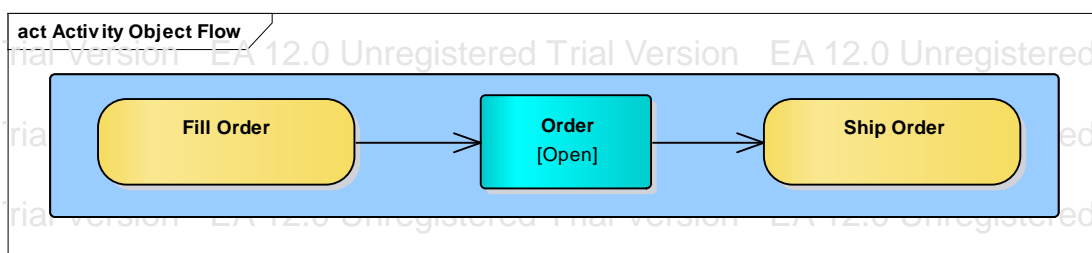
4.2.1



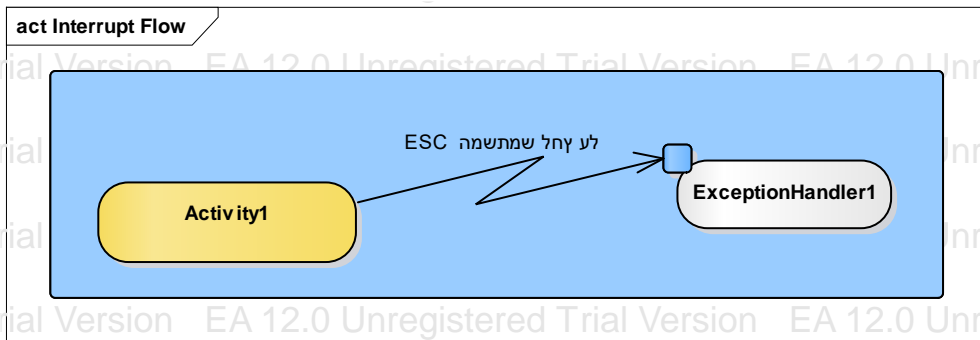
קשר מסוג Control Flow מציין סיום פעילות אחת ומעבר לפעילות הבאה. אם המעבר לפעילות הבאה תלוי בתנאי יש לפרט את התנאי ב Properties של הקשר, בעלה ה Constraints בשדה ה Guard.

להלן דוגמא של Object Flow

4.2.2

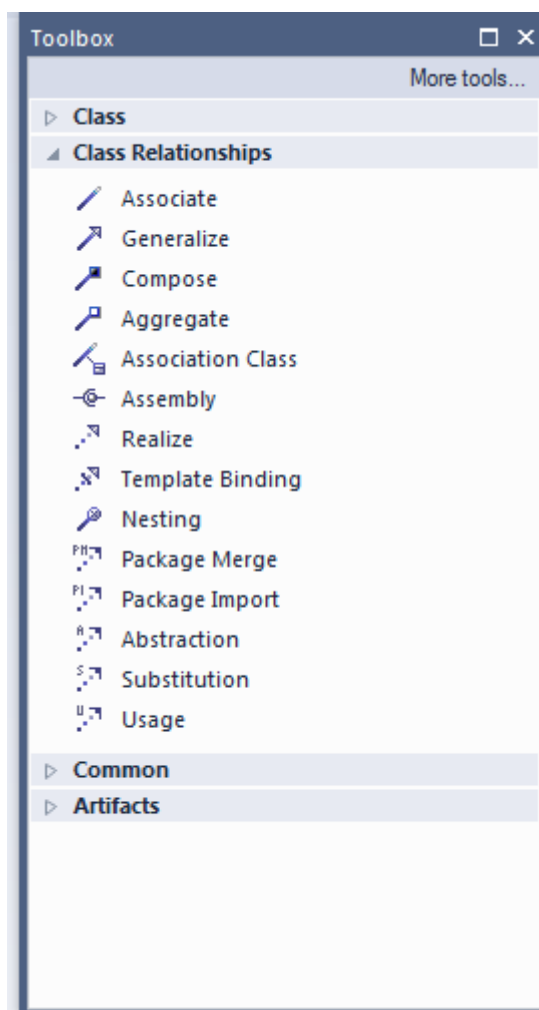


הפעילות בצד שמאל מעבירה אובייקט/מספר אובייקטים לפעילות בצד ימין. בנוסף לשם האובייקט ניתן גם לציין את הסטטוס שלו

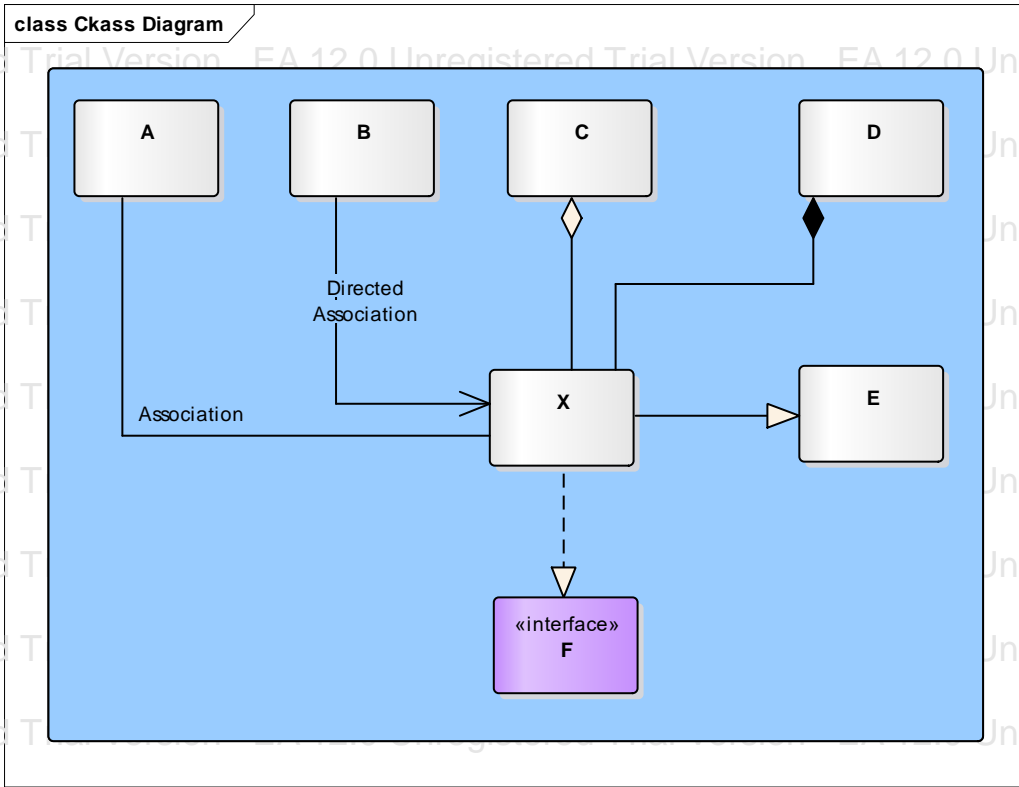


אם במהלך ביצוע Activity1 קרה משהו חריג צפוי, אבל נדיר ונדרשת פעילות מיוחדת ניתן לציין זאת על ידי קשר מסוג Interrupt Flow לפעילות מסוג Exception Handler

4.3 קשרים מיוחדים בתרשים מסוג Class Diagram



4.3.1 הקשרים הנפוצים

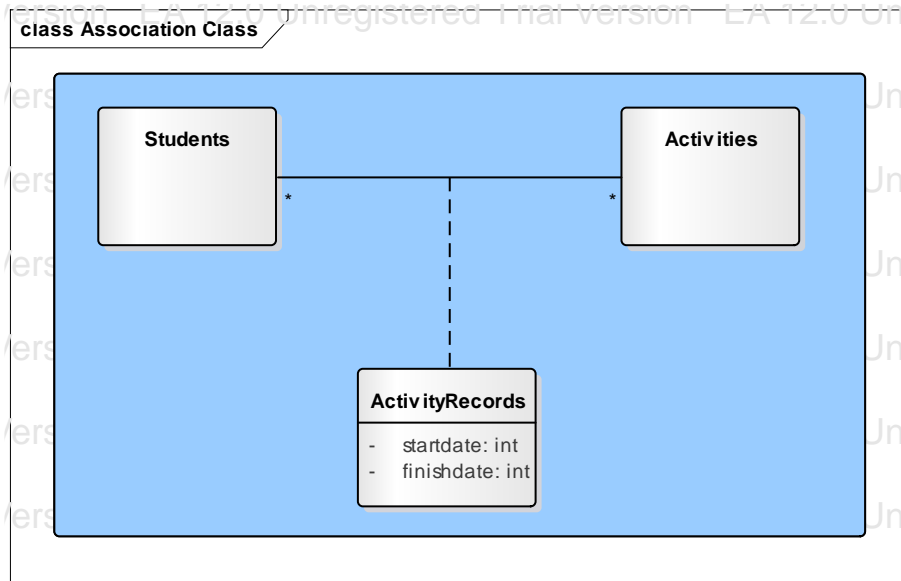


הסבר	סוג הקשר	אלמנט ב	אלמנט א
מחלקה A מכירה את מחלקה X מחלקה X מכירה את מחלקה A המשמעות המעשית X מוגדר ב Attributes של A מוגדר ב Attributes של X	Association	A	X
B מכיר את X X אינו מכיר את B	Directed Association	X	B
מחלקה C כוללת את מחלקה X בצורה שיתופית, כלומר X יכולה להיות כלולה גם במחלקות אחרות. הביטול של C אינו מבטל את קיום X	Aggregation	X	C
<div data-bbox="87 1429 758 1646" data-label="Diagram"> <pre> classDiagram class Car class Wheel Car o-- Wheel </pre> </div>			
מחלקה D כוללת את מחלקה X באופן בלעדי הביטול של D מוחק גם את X	Composition	X	D
<div data-bbox="71 1736 758 1937" data-label="Diagram"> <pre> classDiagram class Company class Department Company *-- Department </pre> </div>			

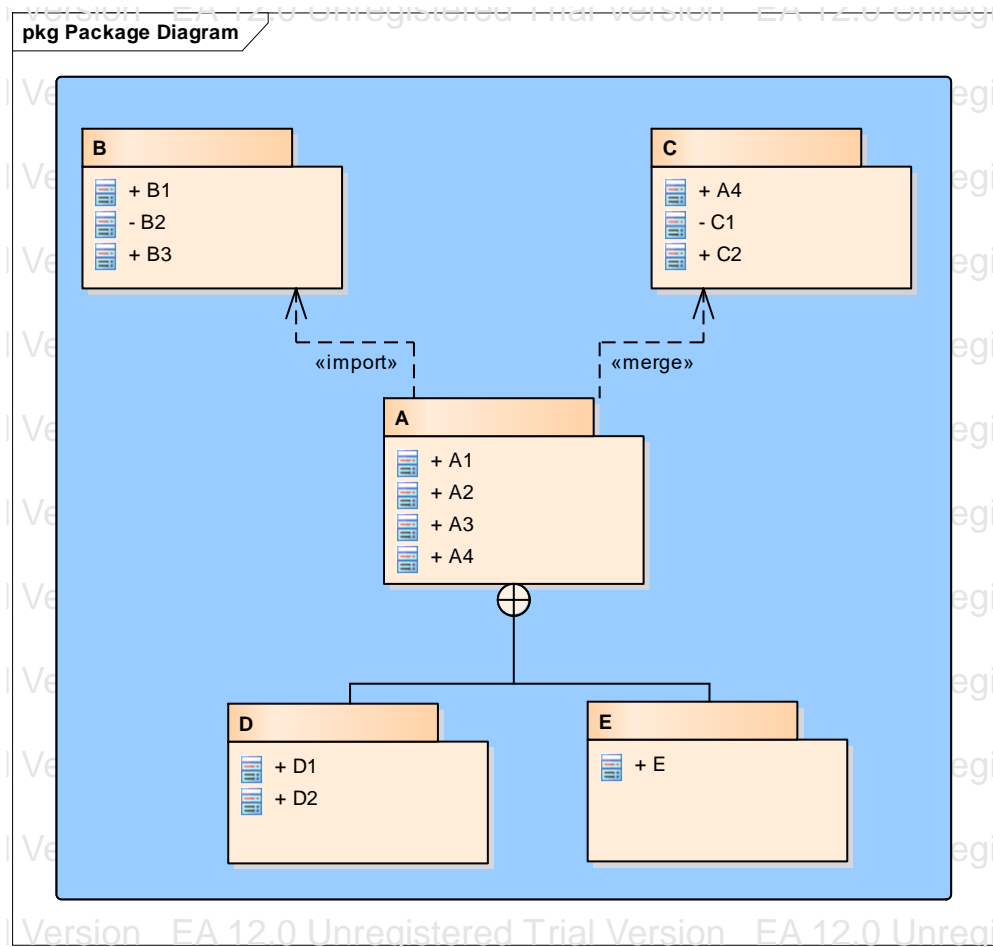
מחלקה X יורשת ממחלקה D הירושה כוללת: Attributes, Operations של E וכן את כל הקשרים	Generalization	E	X
מחלקה X מממשת את מחלקה F מחלקה F חייבת להיות מסוג Interface ל F עשויים להיות מספר מימושים	Realization	F	X

Association Class

4.3.2



4.4 הקשרים המיוחדים ב Package Diagram



אלמנט א	אלמנט ב	סוג הקשר	הסבר
A	B	Import	כל האלמנטים של B מיובאים לתוך A למעט האלמנטים שהם Private כלומר A: A1,A2,A3,A4,B1,B3
A	C	merge	A ימזג לתוכו את כל האלמנטים של C שאינם מוגדרים כ Private ויש להם שם זהה. במקרה זה A כולל: A1,A2,A3,A4, כאשר A4 ב A יכול גם את כל מרכיבי A4 מתוך C
A	D,E	nesting	A כולל את כל האלמנטים של D וגם של E