

דיאגרמת Activity

[הקדמה לדיאגרמת Activity](#)

[השימוש ב – Decision Node](#)

[השימוש ב – Merge Node](#)

[השימוש ב – Fork Node](#)

[השימוש ב – Join Node](#)

[דיאגרמה שכוללת לולאה](#)

[Preconditions – ו Postconditions](#)

[מתן ביטוי לזרימה של אובייקטים](#)

[השימוש ב – Connectors](#)

[Tokens – ו Weight](#)

[שימוש ב – Parameter Nodes](#)

[שימוש ב – Objects Nodes](#)

[השימוש ב – Pins](#)

[מתן ביטוי לטיפול בשגיאות](#)

[חלוקת הדיאגרמה ל – Partitions](#)

הקדמה לדיאגרמת Activity

באמצעות דיאגרמת activity ניתן לתת ביטוי לפעילויות שמתרחשות במערכת. כל פעילות מיוצגת באמצעות מלבן שפינותיו מעוגלות ושכפינותו השמאלית העליונה מצוין שמה.

בדוגמא הבאה מיוצג ה-activity ששמו Approve Credit Card.



בתוך גבולות המלבן אשר מייצג activity מסוים ניתן לצייר את הפעולות הבודדות ששייכות אליו. כל פעולה בודדת מיוצגת באמצעות אותו סימול גראפי שמשמש לייצוג activity רגיל.



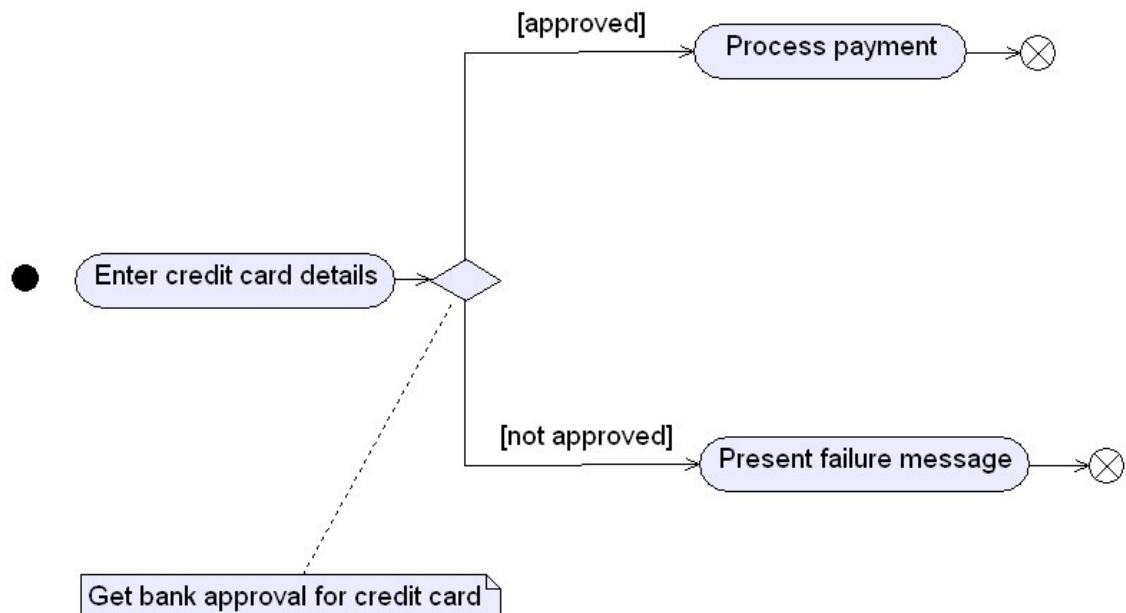
כל דיאגרמת activity מתחילה ב- initial node שמצויר באמצעות עיגול שחור. כל דיאגרמת activity מסתיימת ב-final node שמצויר באמצעות עיגול שחור בתוך טבעת שחורה. דיאגרמת activity נותנת ביטוי לסדרה של התרחשויות אשר קרוייה גם בשם activity flow.



השימוש ב – Decision Node

באמצעות ה-Decision Node ניתן לתת ביטוי לכך שזרימת הפעילויות מתפצלת למספר אפשרויות על פי ערכו של ביטוי בוליאני נתון. את ה-Decision Node מציירים באמצעות מעוין ריק. את הביטוי הבוליאני ניתן לרשום גם באמצעות תיאור טקסט פשוט וניתן לרשום אותו או בתוך note שיחובר בקו ל – Decision Node או בתוך ה – Decision Node (המעוין הריק) עצמו.

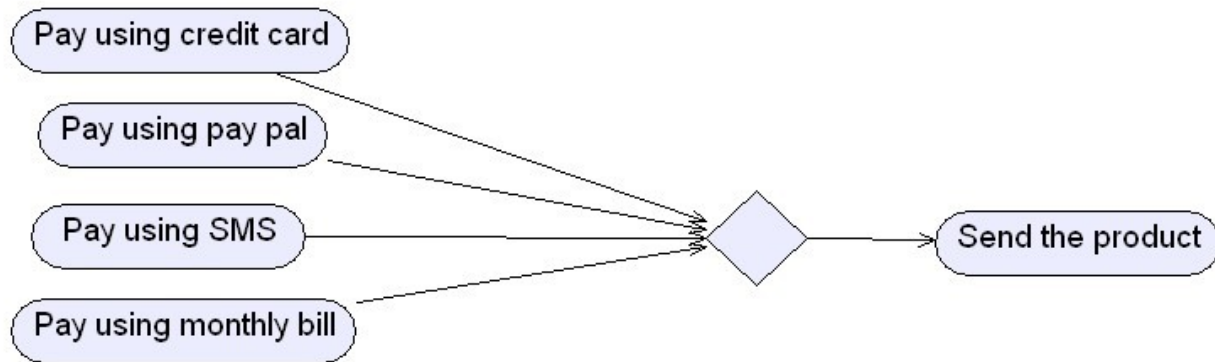
בדוגמא הבאה מוצג activity diagram אשר כולל decision node אשר מתבסס על התנאי 'Get Bank Approval'. אם ה-approval מגיע אז התשלום יבוצע ואם הוא לא יגיע אז תוצג הודעת שגיאה.



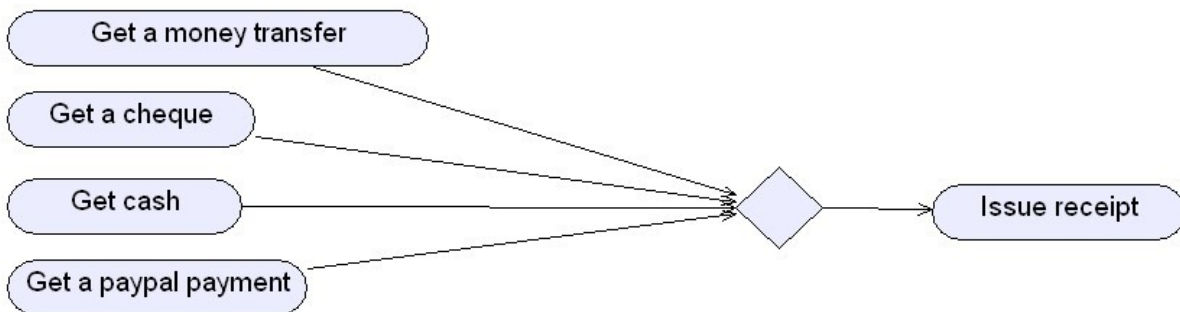
השימוש ב – Merge Node

באמצעות merge node, אשר בדומה ל-decision node מצויר באמצעות מעויין ריק, ניתן לתת ביטוי לכך שמספר activities flows מתאחדים ל – activity flow אחד.

בדוגמא הבאה בין אם התשלום בוצע באמצעות כרטיס אשראי, SMS, pay pal, או באמצעות תשלום הטלפון החודשי הפעילות הבאה שתתרחש תהיה שליחת המוצר עצמו.



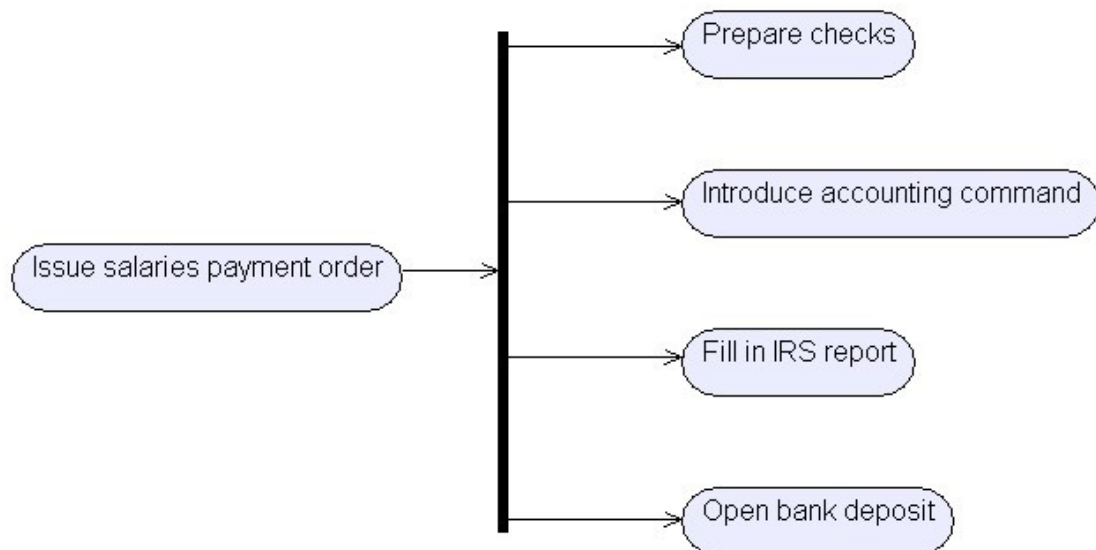
בדוגמא הבאה בין אם התשלום הגיע באמצעות העברה בנקאית, המחאה, מזומן או paypal הפעילות הבאה שתתבצע תהיה הוצאת קבלה.



השימוש ב – Fork Node

באמצעות ה-fork node ניתן לתת ביטוי להתפצלות של activity flow מסויים למספר activity flows אשר מתבצעים במקביל. כל activity flow שיגיע ל-fork node ישוכפל למספר activity flows כמספר הזרועות שה-fork node כולל.

בדוגמא הבאה ה-issue salaries payment order הוא activity שברגע שהוא מסתיים ה-flow מתפצל לארבעה flows אשר מתבצעים במקביל: flow אחד כולל את הכנת השייקים, flow שני כולל את מתן הפקודה החשבונאית בספרים, flow שלישי כולל מילוי של טופס של מס הכנסה ו-flow רביעי אחראי לפתיחת הפיקדון המתאים בבנק (לצורך תשלום המשכורות).

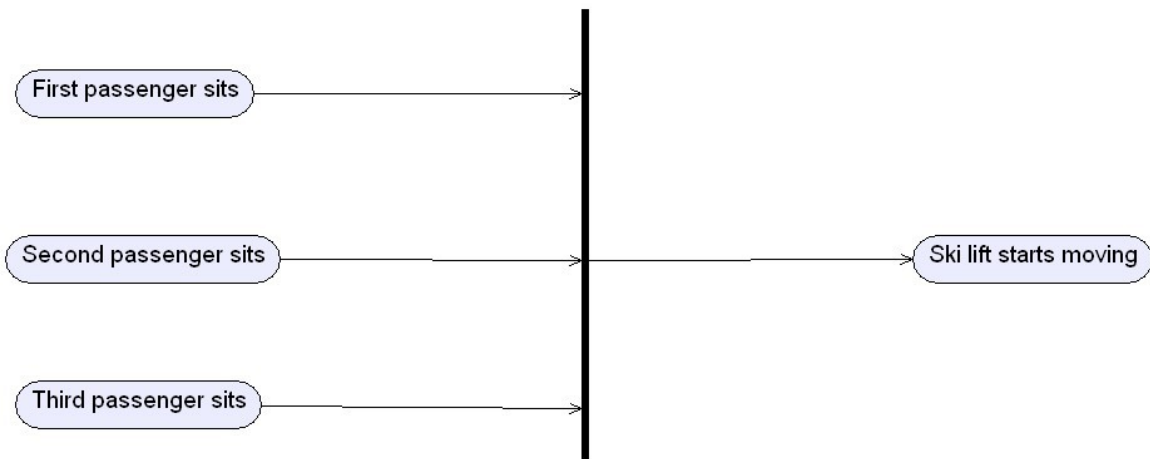


השימוש ב – Join Node

באמצעות join node ניתן לתת ביטוי לכך ש- activity flows שונים מתאחדים באופן מסונכרן אחד עם השני ל-activity flow אחד באופן שה-activity flow שאליו כולם מתאחדים מתחיל בפעולתו רק לאחר שכל ה-activity flows אשר הגיעו אל ה- join node סיימו את פעולתם.

הסימול הגרפי של join node הוא אותו סימול גרפי שמשמש ל-fork node.

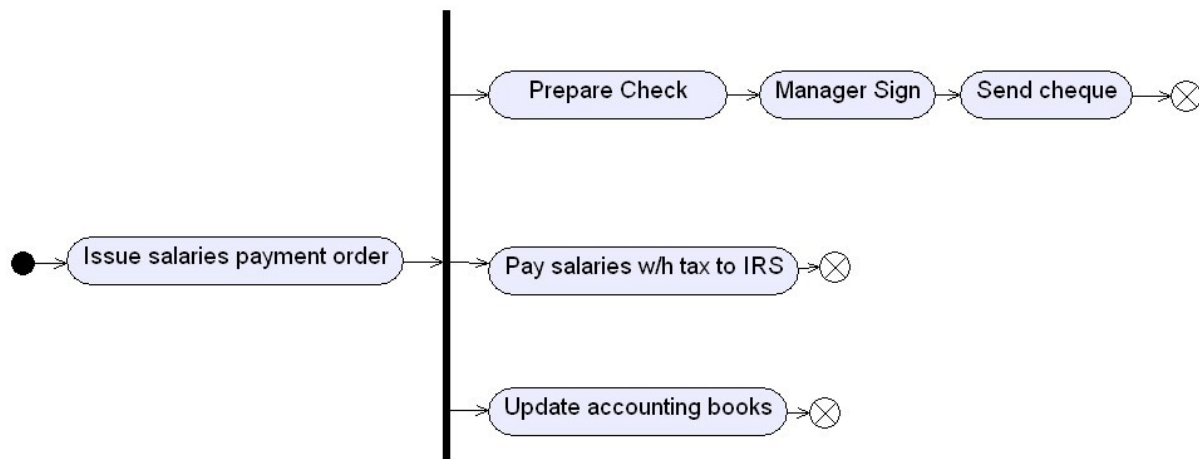
בדוגמא הבאה ניתן ביטוי לעובדה שמעלית הסקי מתחילה בתנועתה רק לאחר שכל שלולת הנוסעים התיישבו.



ניתן להוסיף בסמיכות ל-join node תנאי לוגי כדי לתת ביטוי לכך שהמשך הפעילות יקרה רק אם התנאי הלוגי הוא אמת. בדוגמא הבאה ניתן ביטוי לכך שמעלית הסקי מתחילה בתנועתה רק לאחר ששלושת הנוסעים התיישבו ובתנאי שחגורות הבטיחות של כל אחד מהם הופעלו בהצלחה.



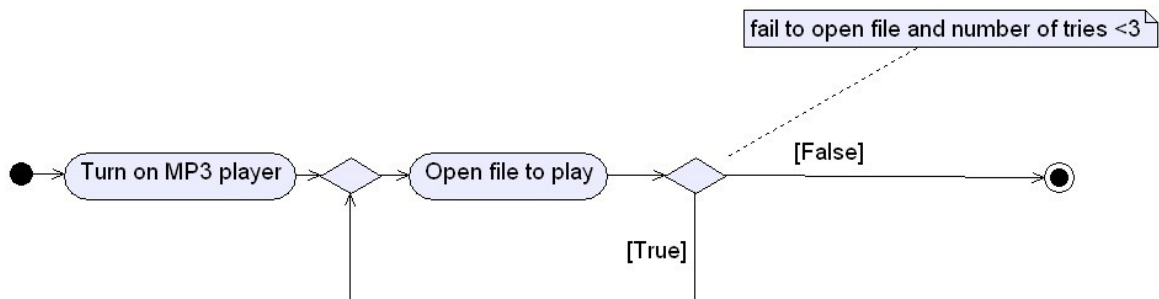
באמצעות ה – flow final node ניתן לתת ביטוי לכך ש-activity flow מסויים הסתיים ללא כל קשר ל-activity diagram עצמה אשר ממשיכה בפעולתה. הסימול הגרפי של ה-flow final node כולל את הסימן X בתוך עיגול ריק. הדוגמא הבאה כוללת שימוש ב-flow final node כדי לתת ביטוי לסיימם של ה-activity flows השונים בנפרד מהאחרים.



דיאגרמה שכוללת לולאה

באמצעות מעוין ריק אשר מייצג תנאי ניתן לייצור תרשים activities אשר כולל בתוכו לולאה / לולאות.

הדוגמא הבאה מתארת תרחיש שבו נעשים ניסיונות חוזרים (לכל היותר שלושה) לפתוח קובץ לצורך השמעתו בנגן .MP3



Postcondition – | Precondition

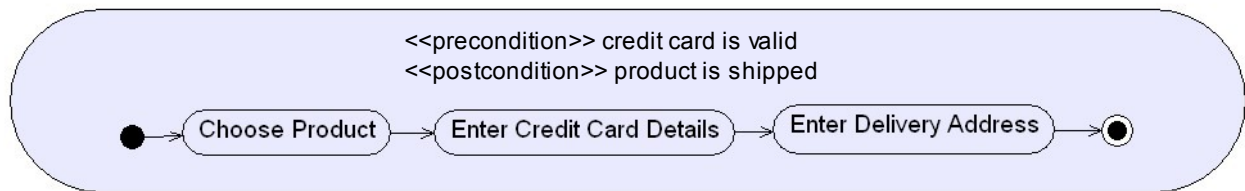
כל activity יכול לכלול precondition (תנאי שחייב להתקיים לפני שהיא מתחילה) ו-postcondition (תנאי שחייב להתקיים לאחר שהיא מסתיימת).

תנאים אלו מקובל להוסיף אל התרשים שמתאר activity מסויים באמצעות ה-stereotypes הבאים:

<<precondition>>

<<postcondition>>

הדוגמא הבאה כוללת את ה-activity ששמו: 'Buying On The Web' בצירוף precondition ו-postcondition באופן הבא: לפני ש-'Buying On The Web' מתחיל כרטיס האשראי חייב להיות מאושר. לאחר שהוא מסתיים ניתן לומר שהמוצר תמיד יישלח לייעדו.

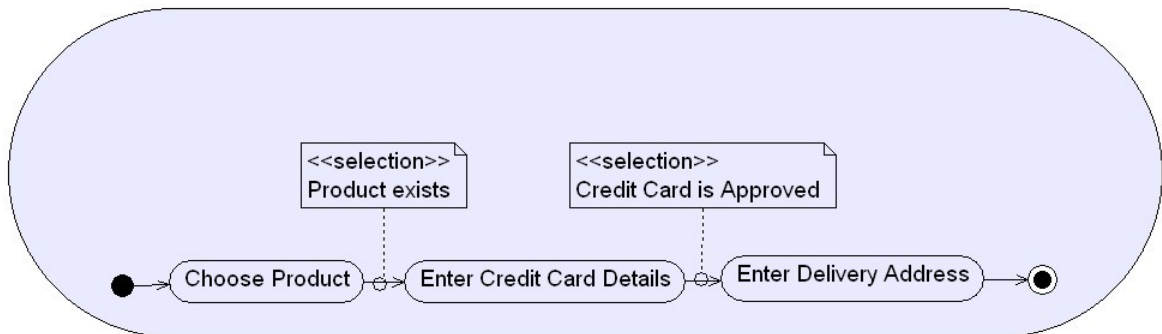


לחיצים שמחברים את ה-activities השונים מקובל לקרוא בשם activity edges והם מייצגים את כיוון הזרימה וההתרחשות מ-activity אחד לשני.

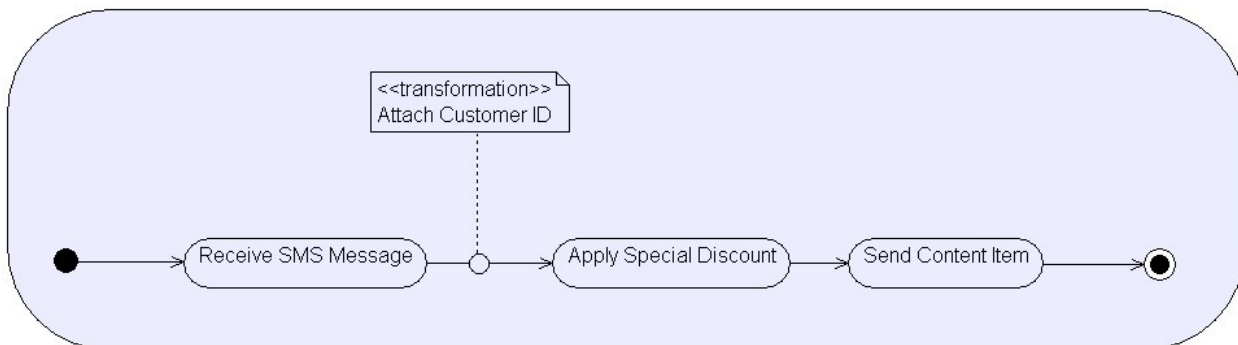
מתן ביטוי לזרימה של אובייקטים

בכל תרשים activity ניתן לדמיין את זרימתם של אובייקטים שונים דרך ה-activities השונים שמופיעים בתרשים.

באמצעות note אשר כולל בתוכו פירוט של תנאי בצירוף '<<selection>>' ניתן לתת ביטוי לכך שכל אובייקט שעובר דרך ה-edge שאליו ה-note מתייחס ימשיך בדרכו רק אם הוא מקיים את התנאי שכתוב בתוך אותו note. בדוגמא הבאה המעבר מ-'Choose Product' ל-'Enter Credit Card Details' מתרחש רק אם המוצר קיים במלאי. כמו כן, המעבר מ-'Enter Credit Card Details' ל-'Enter Delivery Address' מתרחש רק אם כרטיס האשראי מאושר.

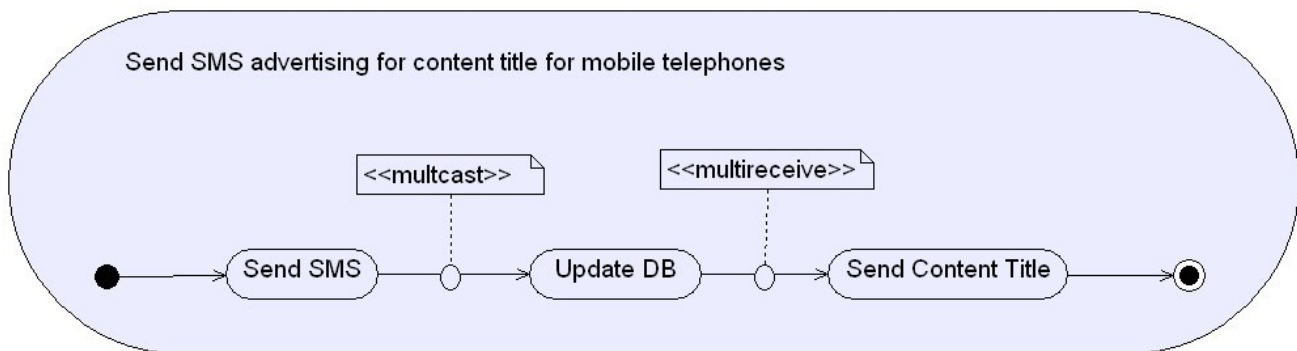


באמצעות note אשר כולל בתוכו תיאור של שינוי שיש לבצע באובייקטים שזורמים במערכת המתוארת בצירוף '<<transformation>>' ניתן לתאר את ביצועו של השינוי האמור בכל אובייקט אשר זורם דרך ה-edge שאליו ה-note מתייחס. בדוגמא הבאה לכל הודעת SMS שמועברת מ-'Receive SMS Message' ל-'Apply Special Discount' מצורף פריט מידע נוסף (ה-customer ID).



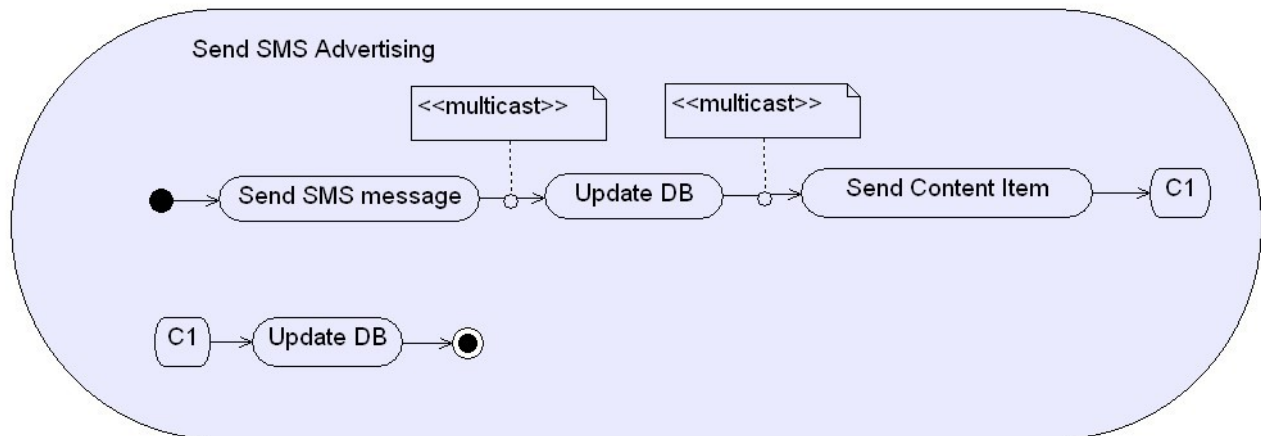
באמצעות note אשר כולל בתוכו <<multicast>> ושמשוייך ל-edge מסויים ניתן לתת ביטוי לכך שכל אובייקט אשר זורם דרך אותו edge ישוכפל במספר גדול של עותקים. באמצעות note אשר כולל בתוכו <<multirecieve>> ושמשוייך ל-edge מסויים ניתן לתת ביטוי לכך שכל האובייקטים אשר זורמים דרך אותו edge יאוחדו מבחינת התרשים לאובייקט אחד וההתרחשות שתתואר ביחס אליו היא ההתרחשות שתהיה בפועל לכל אחד מהאובייקטים שאוחדו.

בדוגמא הבאה כל הודעת SMS אשר עוברת מ-'Send SMS' ל-'Update DB' משוכפלת למספר רב של עותקים ובכך ניתן, למעשה, ביטוי לשליחתה של אותה הודעה אשר הוכנה על ידי ה-activity ששמו 'Send SMS' במספר רב של עותקים לאנשים רבים. באופן דומה, התרשים נותן ביטוי לקבלתן של הודעות טקסט רבות שמהוות בקשות בודדות של משתמשים לקבל אליהם את ה-content title. ההתרחשות שמתוארת באמצעות ה-'Send Content Title' מתייחסת למעשה לכל אחת מההודעות שהתקבלו.



השימוש ב - Connectors

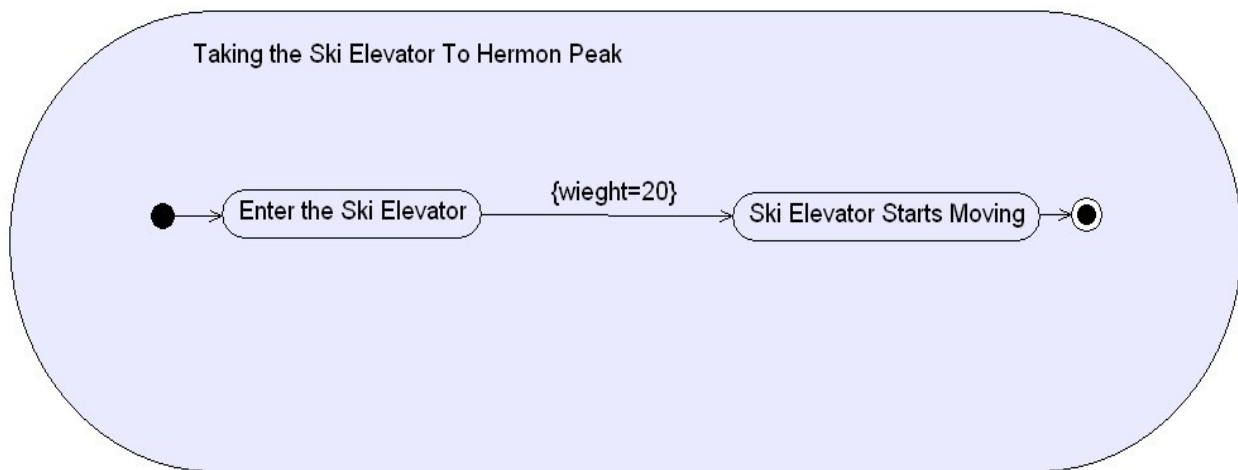
כאשר תרשים activity-ה גדול ואיננו יכולים לכלול אותו בעמוד אחד ניתן לפצלו ולחבר את החלקים השונים באמצעות connectors. כל connector מצויר באמצעות עיגול שבתוכו רשום שם מזהה שבחרנו עבורו.



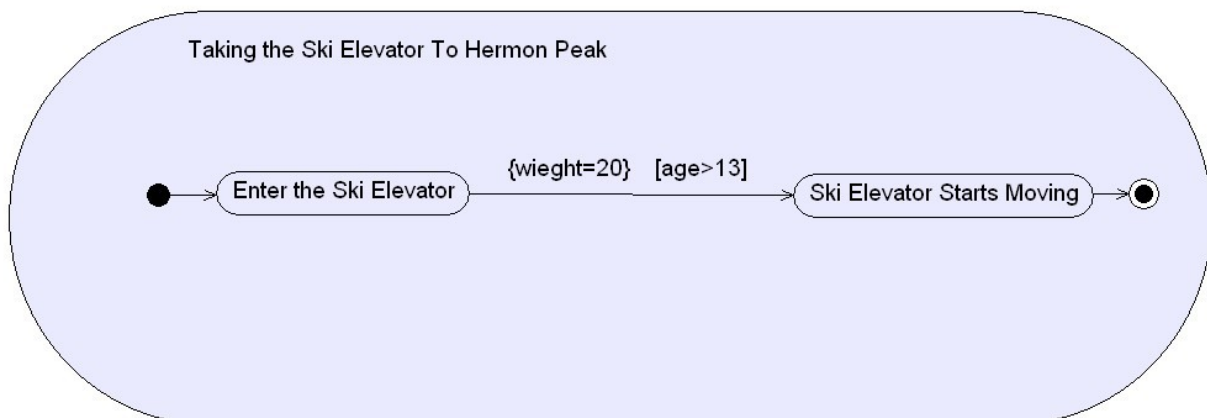
Weight ו- Tokens

את זרימת ההתרחשות דרך ה-activities השונים ניתן לדמיין כזרימה של יחידות קטנות בשם Tokens. בדרך זו, באמצעות weight ניתן לתת ביטוי לכך שהזרימה האמור מ-activity מסויים לאחר תתרחש רק אם מספר ה-Tokens שהצטברו הגיע לערך (weight) מסויים שקבענו.

בדוגמא הבאה המעבר מ-'Enter The Ski Elevator' ל-'Ski Elevator Starts Moving' מתקיים רק אם המספר המצטבר של האנשים (ה-Tokens) שנכנסו למעלית הגיע ל-20.



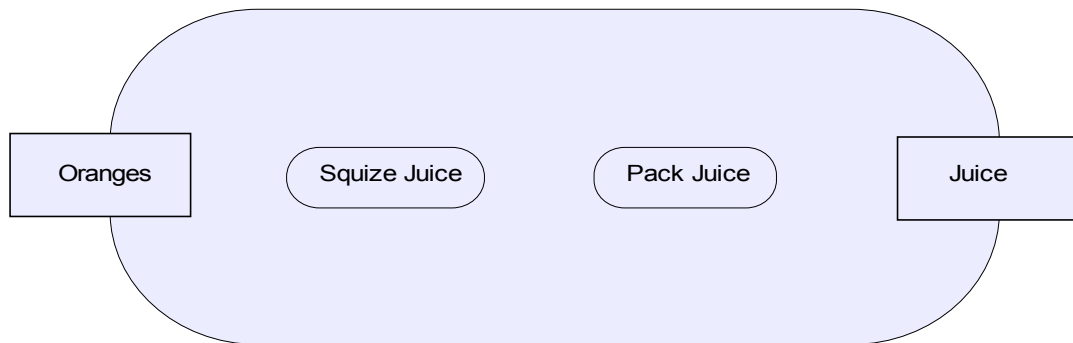
בסמיכות לכל edge ניתן להוסיף Guard Condition כדי לתת ביטוי לכך שהמשך ההתרחשות יקרה רק אם התנאי המצויין מתקיים. בדוגמא הבאה ניתן ביטוי לכך שלא די שמעלית הסקי תכלול כבר 20 אנשים לפחות גילו של כל אחד מהם חייב להיות לפחות 13.



שימוש ב – Parameter Nodes

באמצעות ה-parameter node ניתן לתת ביטוי לפרמטרים שמתקבלים מ-activity מסויים ונשלחים ל-activity אחר. את ה-parameter node מתארים באמצעות מלבן פשוט שבתוכו אנו רושמים את שמו (או את התיאור שלו). את המלבן מציירים על גבי קו המסגרת של ה-activity שאליו הוא מתייחס. את ה-parameter node ניתן לחבר לכל אחד מה-actions הקרובים אליו. בדרך זו, ניתן לתת ביטוי לפרמטר שמתקבל וניתן לתת ביטוי לפרמטר שנשלח הלאה ל-activity הבא.

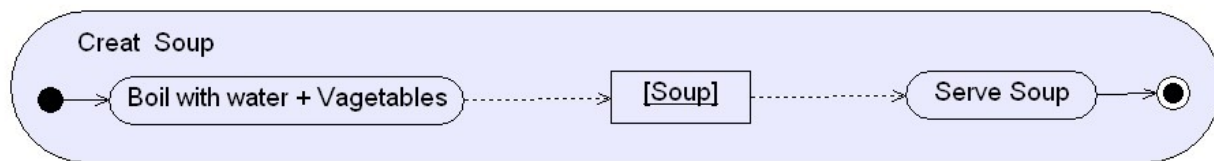
בדוגמא הבאה מופיעים שני parameter nodes. האחד בשם Oranges אשר מייצג פרמטר שמכיל ערך שה-activity מקבל והשני בשם Juice אשר מייצג פרמטר שמכיל ערך שה-activity מעביר הלאה ל-activity הבא.



שימוש ב – Object Nodes

באופן דומה לייצוגם של פרמטרים באמצעות ה-Parameter Nodes ניתן גם לתת ביטוי לאובייקטים שנוצרים מהמחלקות השונות ושבים לידי ביטוי בהתרחשות שהתרחשו שלנו מתאר.

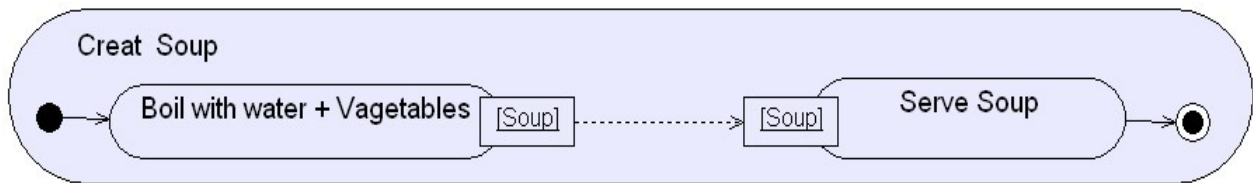
בדוגמא הבאה מתואר ה-activity ששמו 'Create Soup' אשר כולל בתוכו שני actions נפרדים. האחד בשם 'Boil with Water + Vegetables' מסתיים ניתן לומר שנוצר אובייקט ממחלקה בשם 'Soup'. ניתן לתת לכך ביטוי באמצעות ייצוג של אותו אובייקט כמקובל ב-Object Diagram (החל מ-UML 2.X ניתן לשלב יחדיו דיאגרמות מסוגים שונים). בדוגמא זו ניתן לראות שילוב של activity diagram עם object diagram.



השימוש ב - Pins

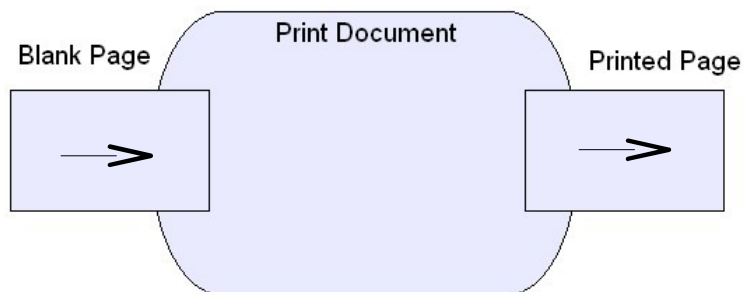
ה - Pin הוא Object Node מיוחד אשר מייצג אובייקט שמשמש כ-input וכ-output מ/ל action מסויים. הייצוג הגרפי של Pin זהה לייצוג הגרפי של Object Node רגיל רק במידות יותר קטנות. כמו כן, מקובל לצייר את ה-Pin כחלק מה-action שאליו הוא מתייחס.

הדוגמא הבאה מציגה את ה-pin ששמו Soup כמי שמייצג אובייקט אשר מהווה output של ה-action ששמו 'Boil with Water + Vegetables' ומהווה input של ה-action ששמו 'Serve Soup'.



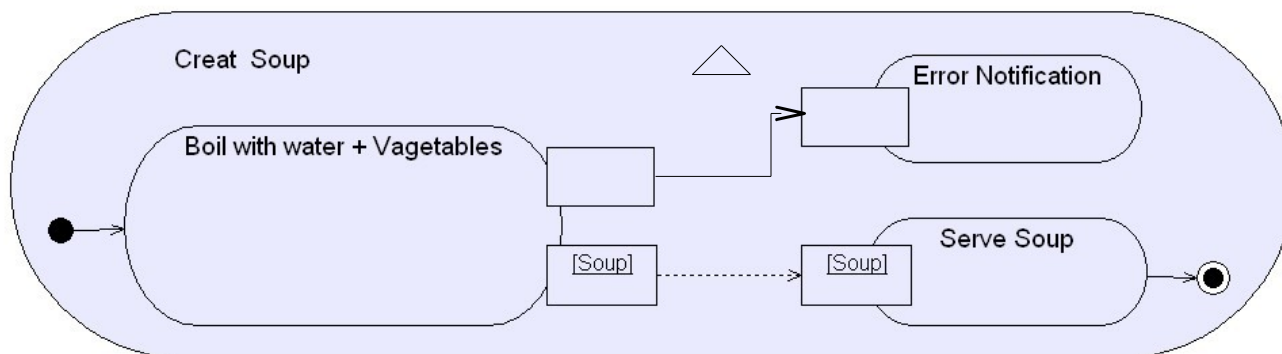
ניתן להוסיף אל תוך גבולות המלבן שמייצג pin חיצים אשר מעידים על היותו של ה-pin אובייקט שמהווה input או אובייקט שמהווה output.

הדוגמא הבאה מציגה שימוש בשני pins ששמם Printed Page ו-Blank Page אשר מייצגים שני אובייקטים. הראשון מהווה output והשני מהווה input.

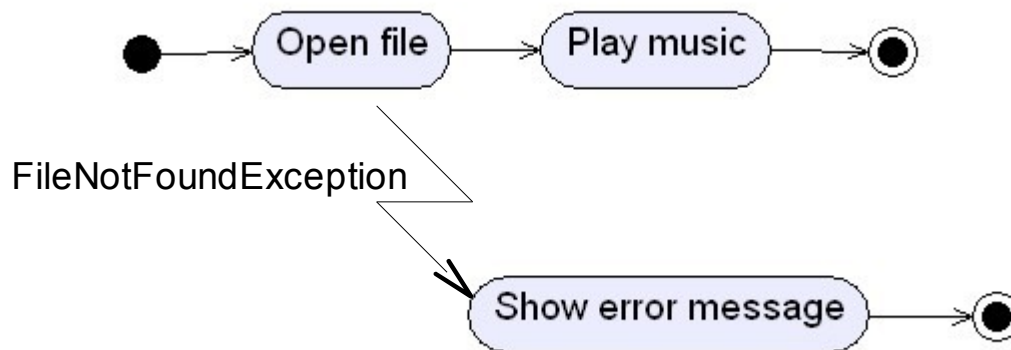


מתן ביטוי לטיפול בשגיאות

כדי לתת ביטוי ל-exception שנזרק יש להשתמש ב-exception pin. הציור הגרפי שמשמש לייצוג exception pin הוא אותו ציור גרפי שמשמש לייצוג pin רגיל בצירוף משולש אשר מצביע על כיוון ה-exception שנזרק.



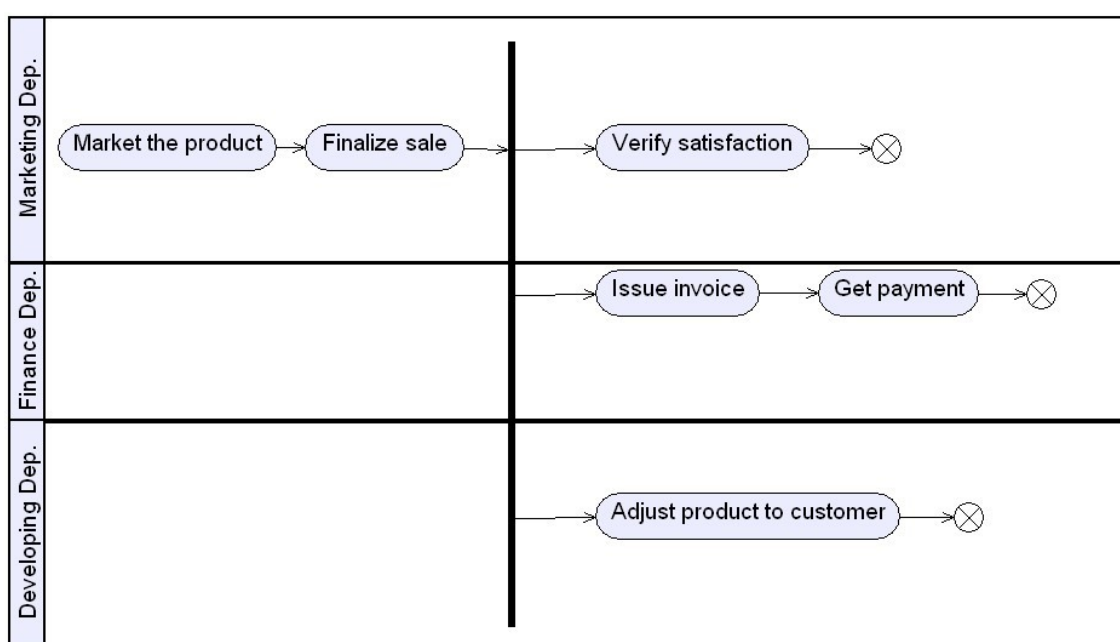
דרך נוספת לתיאור exception שנזרק כוללת שימוש בסימן חץ בצורת ברק. הדוגמא הבא מציגה זאת.



חלוקת הדיאגרמה ל - Partitions

את דיאגרמת ה-activities ניתן לחלק למסלולים (partitions או בשםם האחר: swim lanes). החלוקה יכולה להתבצע על סמך קריטריונים שונים.

כך למשל, ניתן לחלק את הדיאגרמה למסלולים שונים על סמך האחריות לפעילויות השונות שיש ליחידות העיסוקיות השונות באירגון. הדוגמא הבאה מציגה זאת.



ניתן גם לבצע חלוקה למסלולים (partitions) בשני מימדים ובכך לתת ביטוי למאפיינים נוספים של המערכת.

הדוגמא הבאה מציגה חלוקה למסלולים בשני מימדים. מימד אחד נותן ביטוי לאחריות שיש למחלקות השונות באירגון. המימד השני נותן ביטוי למיקום הגיאוגרפי של כל אחת מה-activities שמתרחשים.

