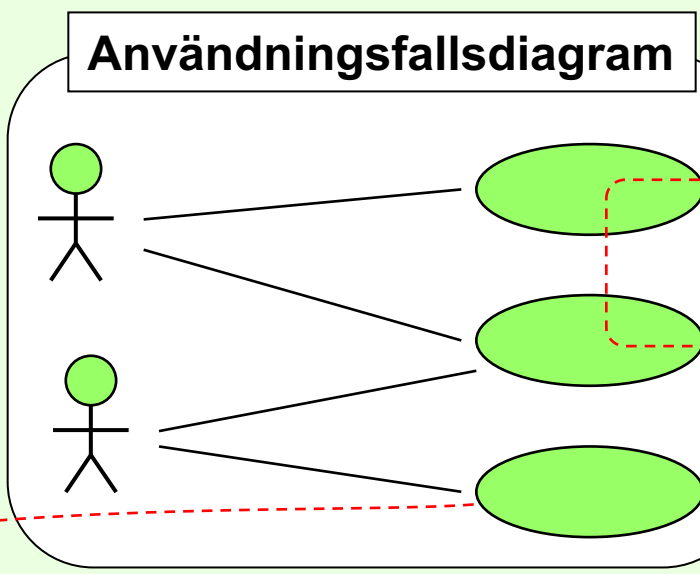


# Modelldriven utveckling enligt RUP®

## Användningsfallsmodell

En **Användningsfallsmodell** är en specifikation av funktionaliteten i ett system

Ett **Användningsfallsdiagram** visar aktörer och användningsfall och hur de kommunicerar med varandra.

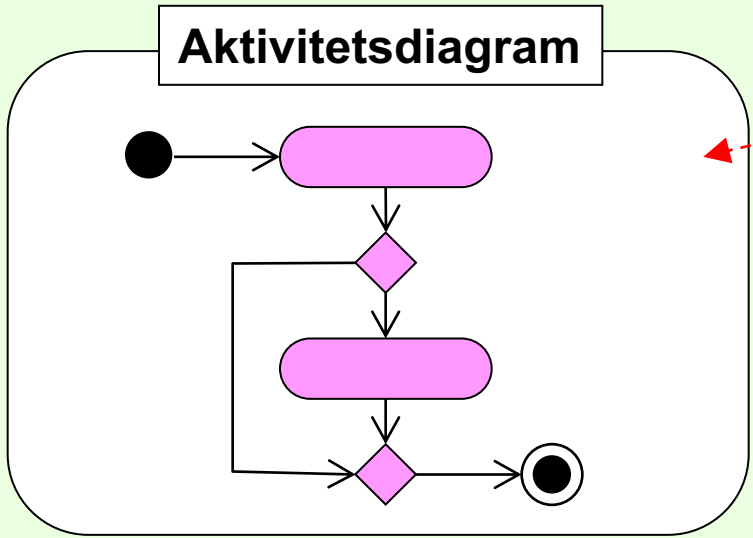


Ett **Användningsfall** specificerar ett sätt att använda systemet som ger observerbar nytta för en aktör.

En **Aktör** är någon som använder systemet. En aktör kan vara ett människa eller ett annat system.

**Analysklasser** identifieras genom att läsa igen specifikationen för varje användningsfall

Ett **Aktivitetsdiagram** kan användas för att specificera flödet i ett enskilt användningsfall.



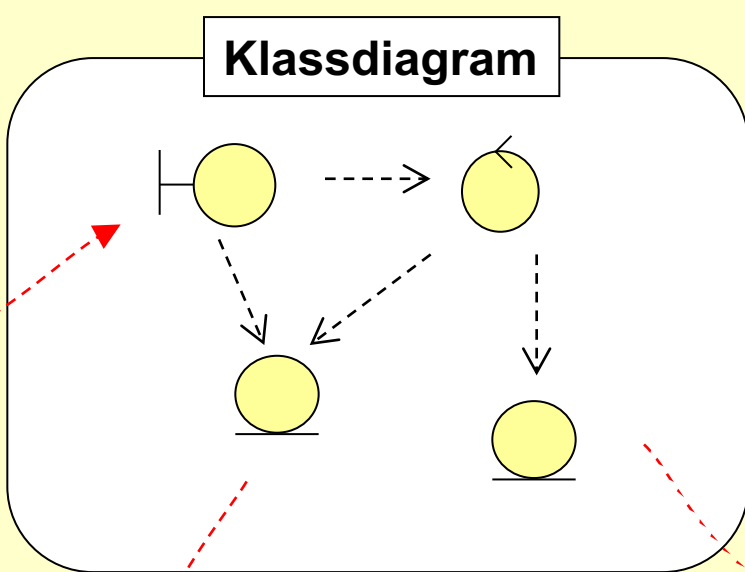
Varje icke-trivialt användningsfall i användningsfallsmodellen ska ha en **Användningsfallsrealisering** i **Analysmodellen**

Varje icke-trivialt användningsfall i användningsfallsmodellen ska ha en **Användningsfallsrealisering** i **Designmodellen**

## Analysmodell

En **Analysmodell** är en abstrakt, implementationsoberoende utformning av systemet

Ett **Klassdiagram** visar analysklasser, deras information, ansvar och relationer mellan varandra.

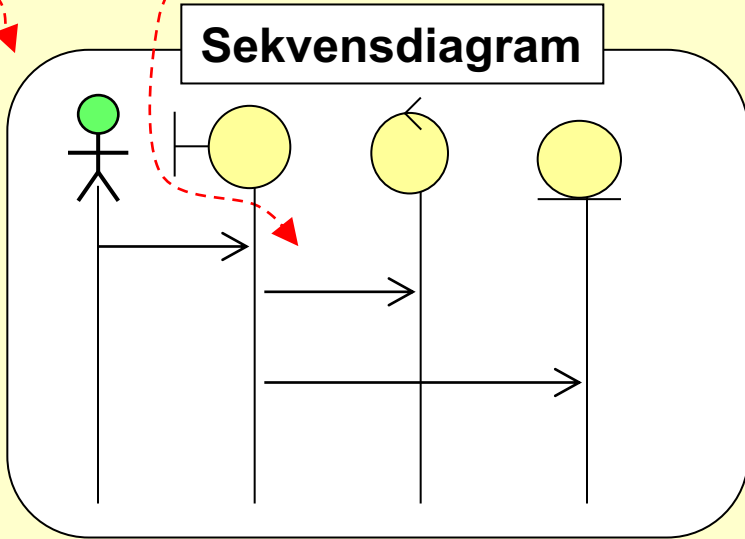


En **Analysklass** är en implementationsoberoende abstraktion av en företeelse i systemet.

Ett ansvar för en analysklass representeras av en **Operation** i UML

Information som hör till en analysklass representeras av ett **Attribut** i UML.

Ett **Sekvensdiagram** visar hur instanser av analysklasserna samarbetar för att realisera ett specifikt flöde genom ett användningsfall.



En **Användningsfallsrealisering** består av ett antal sekvensdiagram som visar hur ett användningsfall realiseras. En användningsfallsrealisering representeras av ett **Samarbete** i UML.

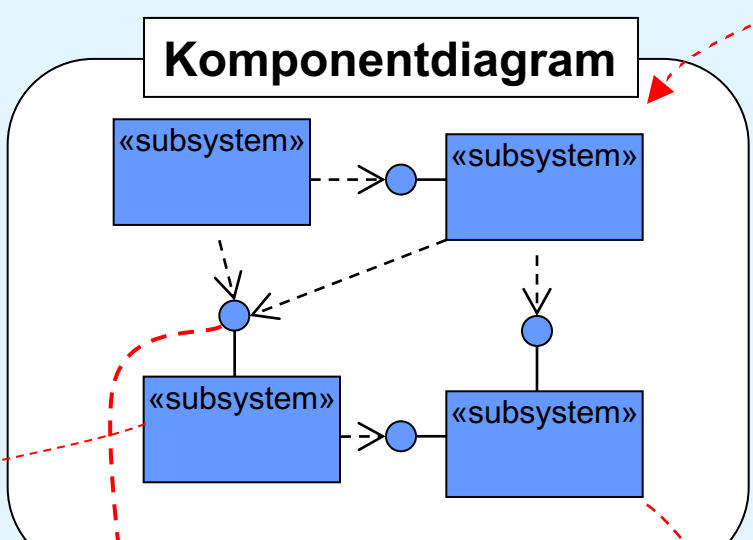
**Delsystem** identifieras genom att gruppera analysklasser som logiskt hör ihop

**Gränssnitt** identifieras genom att gruppera ansvar på analysklasserna som logiskt hör ihop

## Designmodell (systemnivå)

**Systemnivån** av **Designmodellen** är en högnivå nedbrytning av systemet i logiska delsystem och mellanliggande gränssnitt. Denna modell är vanligtvis lagerindeldad

Ett **Komponentdiagram** visar delsystem och hur de tillhandahåller och utnyttjar mellanliggande gränssnitt.



Ett **Delsystem** är en mindre del av ett system med tydliga gränssnitt mot omkringliggande delsystem.

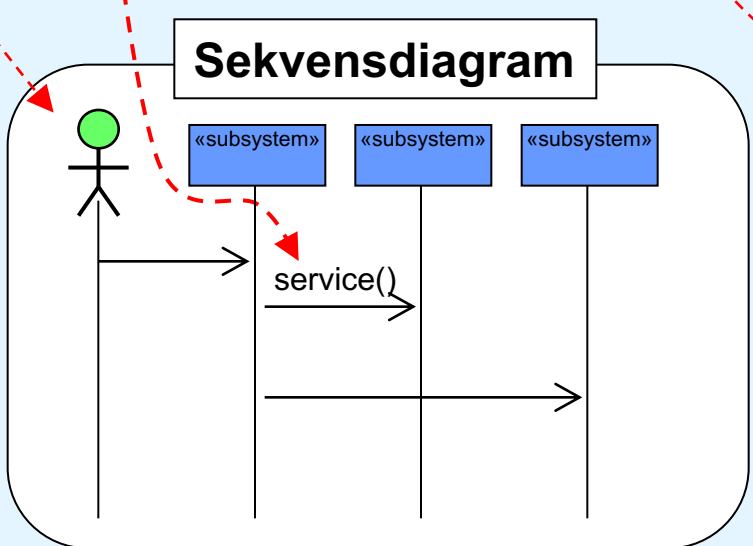
Ett delsystem representeras av en **Komponent** i UML

Ett **Gränssnitt** är en specifikation av ett antal tjänster.

En **Tjänst** representeras av en **Operation** i UML

Det ska finnas en **Delsystemnivå** av **Designmodellen** för varje egenutvecklat delsystem i systemet

Ett **Sekvensdiagram** visar hur delsystemen samarbetar för att realisera ett specifikt flöde genom ett användningsfall.



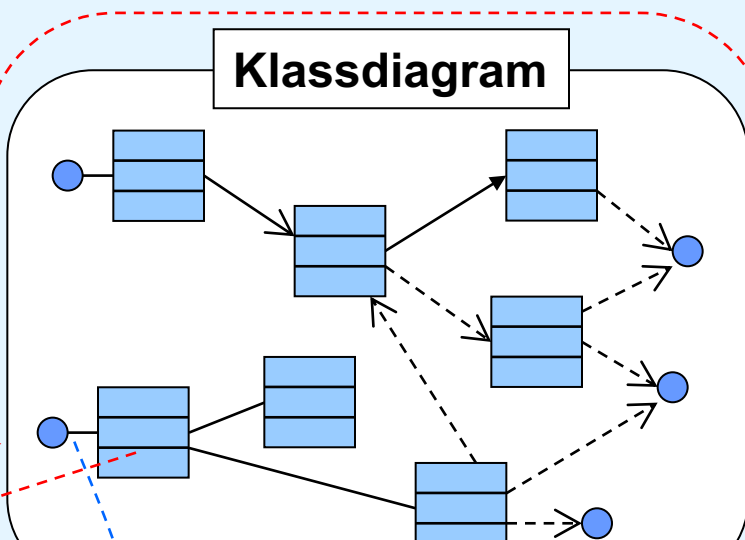
En **Användningsfallsrealisering** består av ett antal sekvensdiagram som visar hur ett användningsfall realiseras. En användningsfallsrealisering representeras av ett **Samarbete** i UML.

En **Artefakt** kan **Manifestera** ett eller flera delsystem

## Designmodell (delsystemnivå)

**Delsystemnivån** av **Designmodellen** är en konkret logisk utformning av ett specifikt delsystem, där hänsyn tagits till hur delsystemet ska implementeras.

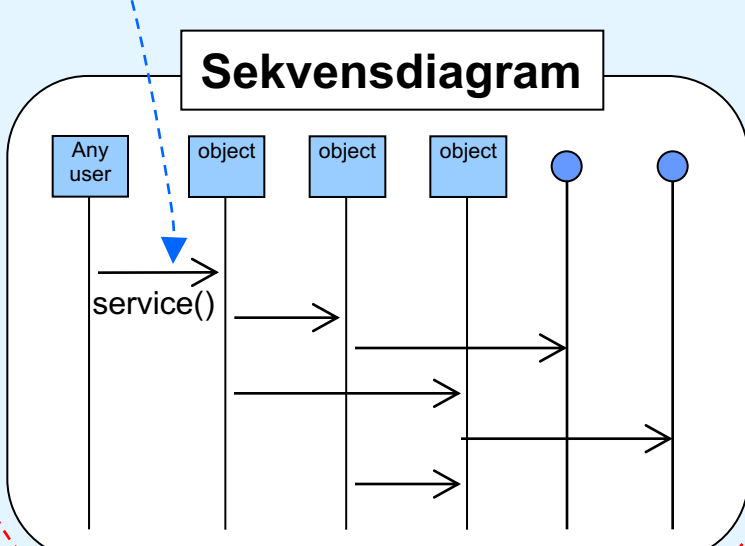
Ett **Klassdiagram** visar designklasser och hur de använder sig av och implementerar publika gränssnitt.



En **Designklass** är en abstraktion av en företeelse i systemet. Dessa motsvarar ofta klasserna i ett objektorienterat programmeringsspråk.

**Attribut** används för att visa informationsbehov och **Operationer** används för att visa funktionalitet i klassen.

Ett **Sekvensdiagram** visar hur instanser av designklasserna samarbetar för att realisera en specifik tjänst i ett publikt gränssnitt.



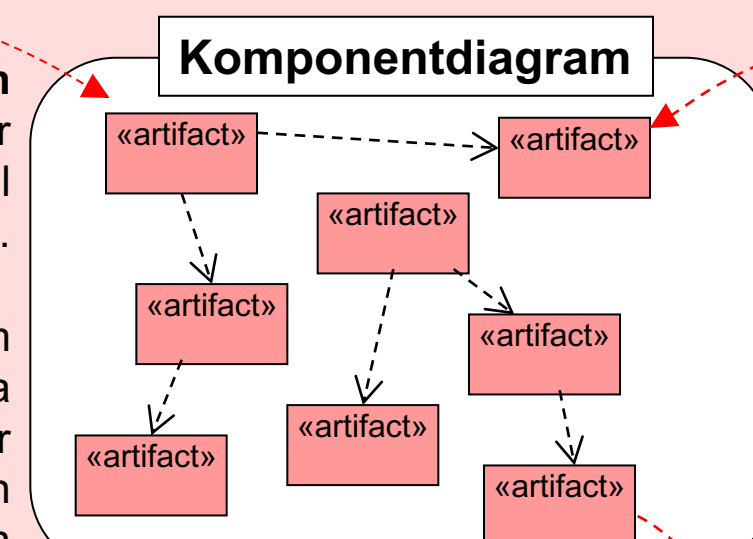
En **Användningsfallsrealisering** består av ett antal sekvensdiagram som visar hur ett användningsfall realiseras. En användningsfallsrealisering representeras av ett **Samarbete** i UML.

En **Artefakt** kan **Manifestera** en eller flera designklasser

## Implementationsmodell

En **Implementationsmodell** visar vilka artefakter som systemet består av

Ett **Komponentdiagram** visar exekverbara artefakter och deras beroenden till varandra vid körning.



Komponentdiagram används också för att visa hur exekverbara artefakter manifesterar delsystem och designklasser från designmodellen.

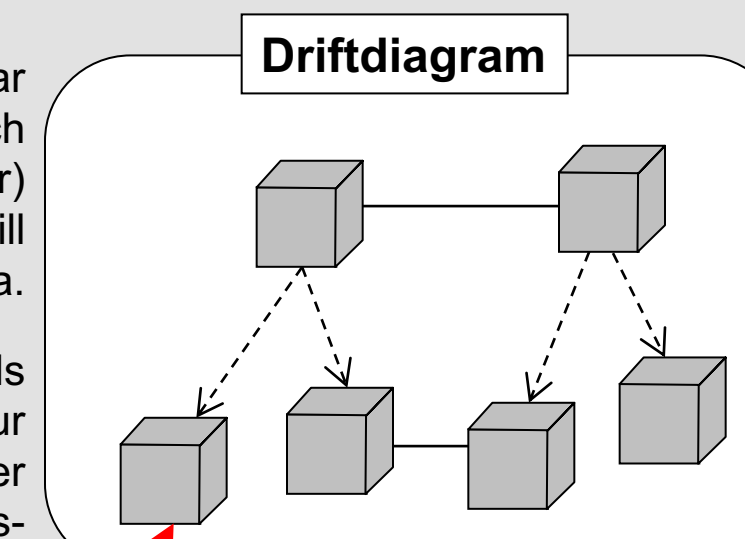
En **Exekverbar Artefakt** är en körbar fil i systemet. Det kan vara en.exe-file, .jar-fil eller annat beroende på implementationstekniken.

En hårdvaruenhet eller exekveringsomgivning kan **Driftsätta** en eller flera exekverbara artefakter

## Driftmodell

En **Driftmodell** visar hur ett system är driftsatt i driftmiljön

En **Driftdiagram** visar noder (hårdvaruenheter och exekveringsomgivningar) och hur de är relaterade till varandra.



Ett driftdiagram används också för att visa hur exekverbara artefakter från Implementationsmodellen är driftsatt på de olika noderna.

En **Exekveringsomgivning** är en variant av nod som representerar en programvara som används för att exekvera en artefakt på en hårdvaruenhet.

En **Hårdvaruenhet** är en variant av nod, som representerar en fysisk dator eller annan hårdvara i ett system.

## Generella UML koncept

### Modell

- En modell är ett avbildning av ett system utifrån ett specifikt perspektiv och på en specifik abstraktionsnivå.
- En modell består av ett antal element, relationer och diagram.
- Exempel på modeller: *Användningsfallsmodell, Designmodell, Driftmodell*

### Diagram

- Ett diagram är en visualisering av en delmängd av en modell.
- Ett diagram visar element och deras relationer.
- Exempel på diagram: *Användningsfallsdiagram, Sekvensdiagram, Driftdiagram*

### Element

- Ett element används för att representera något logiskt eller fysiskt koncept i eller runt ett system
- Element ägs av andra element eller av själva modell och kan visas i ett eller flera diagram
- Exempel på element: *Aktör, Komponent, Nod, Gränssnitt, Samarbete*

### Relation

- En relation är en koppling mellan två eller fler element som uttrycker något strukturell eller beteendemässig relation mellan de koncept som elementen representerar
- Relationer ägs av element i modellen och kan visas i ett eller flera diagram
- Exempel på relationer: *Beroende, Association, Arv, Manifestering*

## Vanliga UML-begrepp på svenska

### Svenska

Användningsfall  
Användningsfallsrealisering  
Artefakt  
Delsystem  
Driftsätta  
Exekveringsomgivning  
Gränssnitt  
Hårdvaruenhet  
Manifestera  
Samarbete

### Engelska

Use-Case  
Use-Case Realization (*variant av Collaboration*)  
Artifact  
Subsystem (*variant av Component*)  
Deploy  
Execution Environment (*variant av Node*)  
Interface  
Device (*variant av Node*)  
Manifest  
Collaboration