

Dibk / FDV Bygg – UML-modellering

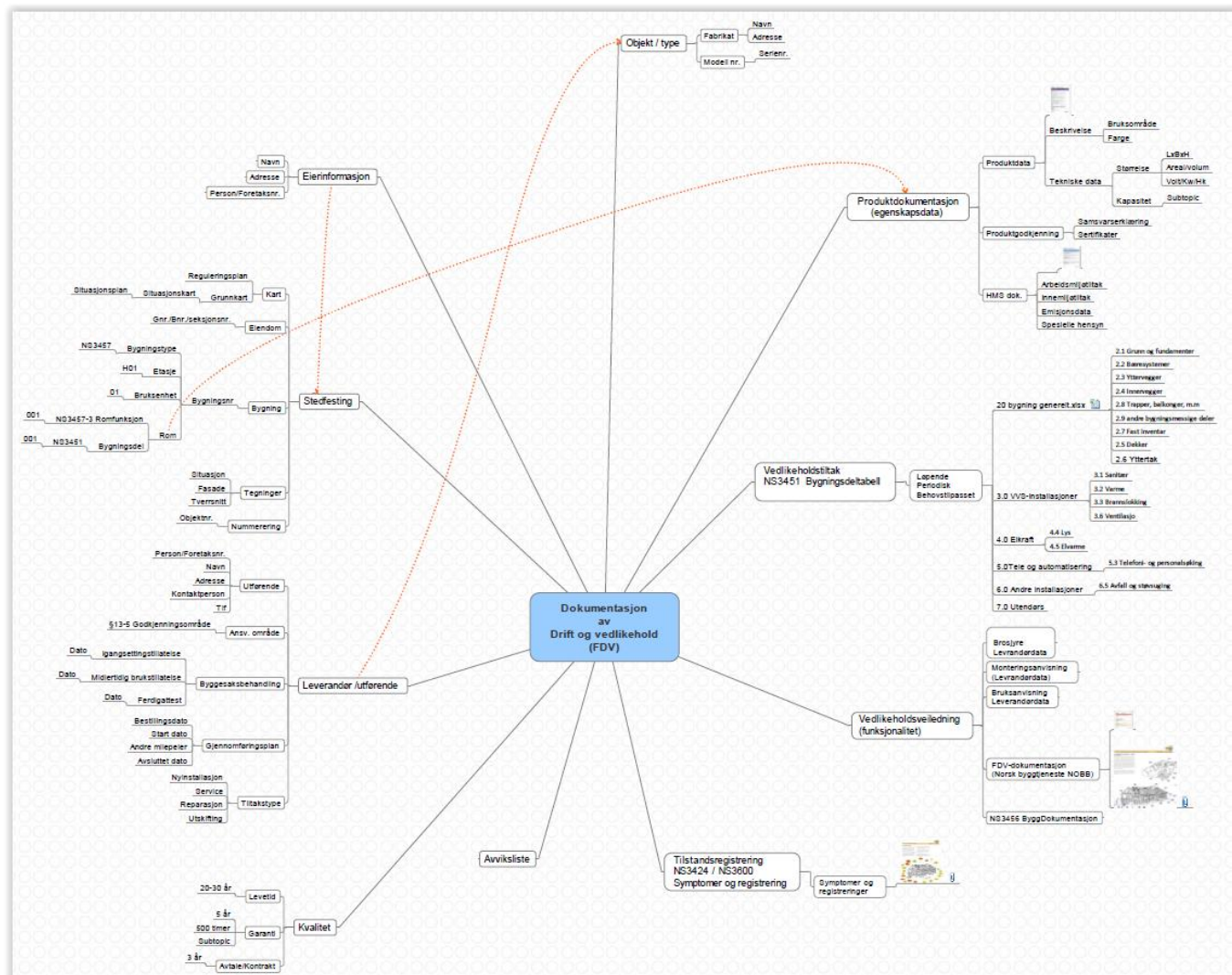
Dato	2014-11-03
Fra	Erling Onstein, Arkitektum AS
Til	Dibk v/ Frode Horjen

Innhold

1	Hensikten	2
1.1	Utgangspunktet for modelleringen.....	2
1.2	Hensikt med UML-modellen	2
1.3	Status på modell-arbeidet.....	3
1.4	Hva er ikke gjort: Vurdering av påkrevde og frivillige deler.....	3
1.5	Hvordan sette seg inn i UML-modellen.....	3
1.6	Hovedutfordringene	3
2	Forklaring på modellen	5
2.1	FDV_Bygning : senteret i modellen	5
2.2	Oppdeling av bygninger i strukturelementer.....	6
2.3	Hvordan knytte informasjon til «byggnings-strukturen»	7
3	Brukseksempel	9
3.1	Eksempel 1 Hente informasjon fra matrikkelen.....	9
3.2	Eksempel 2 Opp-pussing av bad.....	9
3.3	Eksempel 3 Legge inn bygningstegninger	10
3.4	Kontroll av elektrisk opplegg.....	11
4	Bygg_FDV.....	12
4.1	Byggeklosser - forklaringsdiagrammer.....	12
4.2	Full dokumentasjon av UML-modellen.....	17
5	Intro til UML.....	18

1 Hensikten

1.1 Utgangspunktet for modelleringen



Figur 1 Utgangspunktet for UML-modelleringen - tankekart

UML-modelleringen av FDV-dokumentasjonen har tatt utgangspunkt i tankekart presentert på prosjektmøte 30.sept 2014.

1.2 Hensikt med UML-modellen

En UML-modell er en konseptuell modell. Det betyr at det er konseptene, denne gangen i FDV-dokumentasjon, som skal beskrives. De ulike delene av FDV-dokumentasjonen skal identifiseres, og det skal beskrives hvordan delene «henger sammen». UML-modellen skal, sagt på en annen måte, skape en felles forståelse for informasjonsinnholdet i FDV-dokumentasjon av bygninger.

Ut fra den konseptuelle UML-modellen kan det lages ulike implementasjoner/datasystemer for bruk. UML-modellen gir føringer for hvordan implementasjoner skal gjøres. Føringene bør være så klare at informasjon enkelt kan utveksles mellom de ulike implementasjonene.

Modellen foreslår en metode for strukturering av tilgjengelig informasjon. Den sier igjen ting om hvordan informasjonen skal presenteres til ulike brukere, og dermed heller ikke hvilke deler av informasjonen som skal skjules for brukere som antas ikke å ha behov/ikke forstå betydningen.

1.3 Status på modell-arbeidet

Innsatsen så langt i modellarbeidet har vært å identifisere flest mulig av delene av en FDV-dokumentasjon for en bygning.

Informasjon om standarder og andre kilder er tatt fra åpne sider på internett, og ellers fra åpne kilder Arkitektum har tilgang til.

1.4 Hva er ikke gjort: Vurdering av påkrevde og frivillige deler

UML har gode muligheter for å bestemme hvilke deler av en modell som er påkrevd (må følges av alle som skal forholde seg til modellen) og hvilke deler som er frivillige. Dette vises i modelldokumentasjon som multiplisitet:

- 0..* : kan minimum forekomme 0 ganger (dvs frivillig) og kan maksimalt forekomme uendelig antall ganger (*)
- 0..1 : min 0, maks 1 gang.
- 1..* : min 1 (dvs påkrevd), maks uendelig
- 1 : min 1, maks 1, dvs en og bare en gang

Men i denne versjonen av dokumentet er IKKE disse forholdene gjennomarbeidet.

Det vil være nyttig å gå gjennom disse multiplisitets-angivelsene når innholdet er mer fullstendig.

1.5 Hvordan sette seg inn i UML-modellen

Dokumentasjonen av UML-modellen er delt i to dokumenter:

- Hoveddokumentet (den du leser i nå). Denne har også en «lyn-intro» til å lese UML-dokumentasjon (se kap 5)
- Vedlegg med full UML-dokumentasjon, kan fås ved henvendelse til Fredrik Horjen eller Erling Onstein (erling@arkitektum.no)

Denne delingen er gjort fordi det antatt at fleste leserne av rapporten vil klare seg med innholdet i hoved-dokumentet.

I tillegg til de to tekst-dokumentene finnes en prosjekt-fil for UML-editoren Enterprise Architect. Denne inneholder den fullstendige modellen. De som ser for seg å videreutvikle modellen, kanskje i retning implementasjon, kan ta utgangspunkt i denne prosjektfila. Den kan fås fra Fredrik Horjen eller Erling Onstein.

1.6 Hovedutfordringene

Etter å ha diskutert den foreslåtte modellen i flere møter, med ulik representasjon, synes hovedutfordringene å være.

- Logisk oppdeling av en bygning. I modellen er det nå benyttet tre delvis overlappende oppdelinger
 - basert på BIM/IFC SpatialStructureElement-prinsippet, der bygningen blir delt opp i etasjer og etasjene deles opp i rom
 - bruksenhetsoppdeling, slik det gjøres i matrikkelen
 - bygningsdel, etter bygningsdel-tabellen i NS3451, siffer-nivå 1 og 2

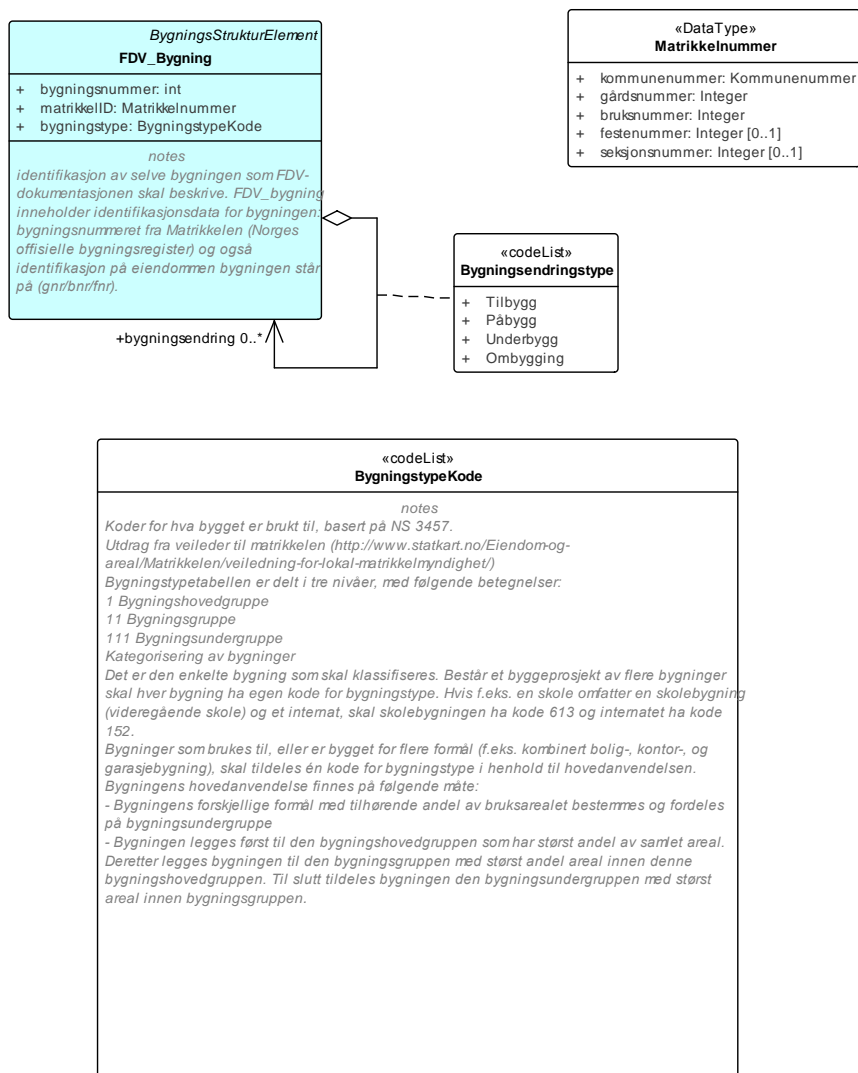
Hensikten med oppdelingen er å ha logiske enheter å knytte dokumentasjon til. Det videre arbeid må avklare om disse tre oppdelings-«prinsippene» kan leve «side om side».

- Oppdelingen av dokumentasjonen. Dokumentasjonen som finnes/utarbeides om bygninger og arbeid som utføres i bygninger, samles i dag i dokument. Dokumentene kan gjelde alt som er relevant for en bolig (f.eks. en takst), et tiltak (f.eks. opp-pussing av bad), eller en

bygningselement (f.eks. varmepumpa). Denne oppdelingen av dokumentasjon basert på hva som er naturlig når dokumentasjonen blir produsert, kan ved senere tids bruk bli en stor utfordring. Eksempel på brukerbehov som kan ha fordel av annen dokumentasjonsoppdeling, er å lage vedlikeholdsinformasjon om «hele bygningen». Da vil en måtte gå gjennom mange dokumenter knytta til både **FDV_Enheter**, **Tiltak (FDV_Oppdrag)**, **Bygningsdeler** og **Bygningselementer**, og plukke ut det som er relevant.

2 Forklaring på modellen

2.1 FDV_Bygning : senteret i modellen



Figur 2 FDV_Bygning

FDV_Bygning er den sentrale objekttypen i modellen. All annen informasjon er knyttet til denne. FDV_Bygning er identifisert med tre egenskaper:

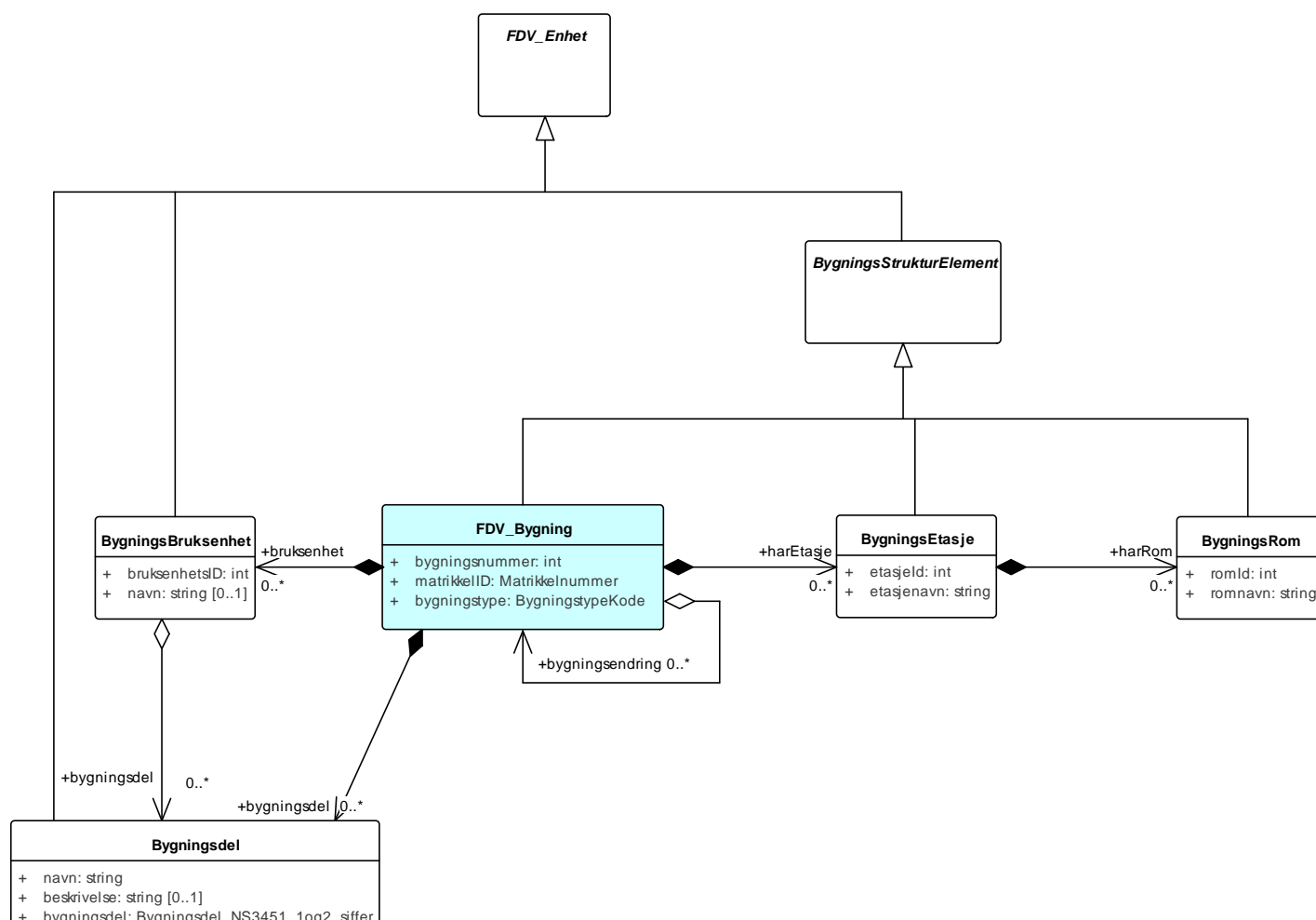
- bygningsnummer: ID-en som bygningen er identifisert med i Matrikkelen
- matrikkelID: knytting til gnr/bnr/fnr
- bygningstype: klassifiseringen av bygningen etter NS 3457

UML-kunnskap:

- **Om modell-nivå.**
 - En UML-modell beskriver «**type-nivået**» av informasjonsbeskrivelser. Det betyr at den identifiserer klasser av forekomster, og forteller hva som er felles for alle forekomster av denne klassen.
 - Et datasett er på **instans/forekomst-nivået**, nivået «under» type-nivået. Når en legger inn en forekomst i et datasett, må en bruke «type-malen» for å se hvordan forekomsten skal være, og tilordne forekomsten egenskapsverdier for hver egenskapstype i «type-nivået»
 - En systemutvikler / database-tilrettelegger vil også se på UML-modellen for å se hvilke egenskaper og assosiasjoner som kan forekomme for hvert modell-element, og tilrettelegge databasen slik at dette er mulig.

- Ved å lage en streng modell, kan en sikre at alle som bruker modellen gjør arbeidet svært likt. En mer åpen modell vil flere «frihetsgrader» til brukerne.
- En **klasse** i UML framstilles i **UML klassesdiagram** som firkant med inntil 3 «etasjer». Den øverste inneholder alltid klassenavnet, nest øverst finnes egenskapene/attributtene og tredje øverst finnes operasjonene. Operasjoner er lite brukt. I tillegg kan en i et klassesdiagram «tvinge fram» tilhørende forklaringer.
- En klasse kan brukes til å representere flere ulike modell-elementer:
 - **objekttyper**: «noe» som har en egen identitet, og som ikke er avhengig av andre elementer for å «leve»
 - **datatype**: en samling egenskaper som naturlig hører sammen. Tilsvarende i stor grad det BIM kaller «property set». I UML merkes slike ofte med **stereotypen** <<dataType>>. Stereotypen vises i klassesdiagrammene foran klassenavnene
 - **kodelister**: Dette er samling av kodeverdier. En velger normalt en kodeverdi fra kodelista for å beskrive en klasse. Merkes med stereotypen <<codeList>>.
- Klasser kan ha **assosiasjoner** til andre klasser, og også med seg selv. Dette vises med «streker» mellom klassene. Strekene kan ha ulike symbol i hver ende, og kan ha tilhørende egenskaper. I klassesdiagrammet over vises det en «selv-assosiasjon» som sier en bygning kan ha (assosiasjon med «diamant-ende») bygningsendring.

2.2 Oppdeling av bygninger i strukturelementer



Figur 3 FDV_Enhet

I BIM/IFC er en bygning logisk oppdelt i flere av det IFC kaller SpatialStructureElements. Dette er vist i Figur 3. Det finnes altså tre typer bygningsstrukturelementer: FDV_Bygning, BygningsEtasje og BygningsRom.

I enkelte sammenhenger er det også formålstjenlig å dele opp en bygning i bruksenheter. En

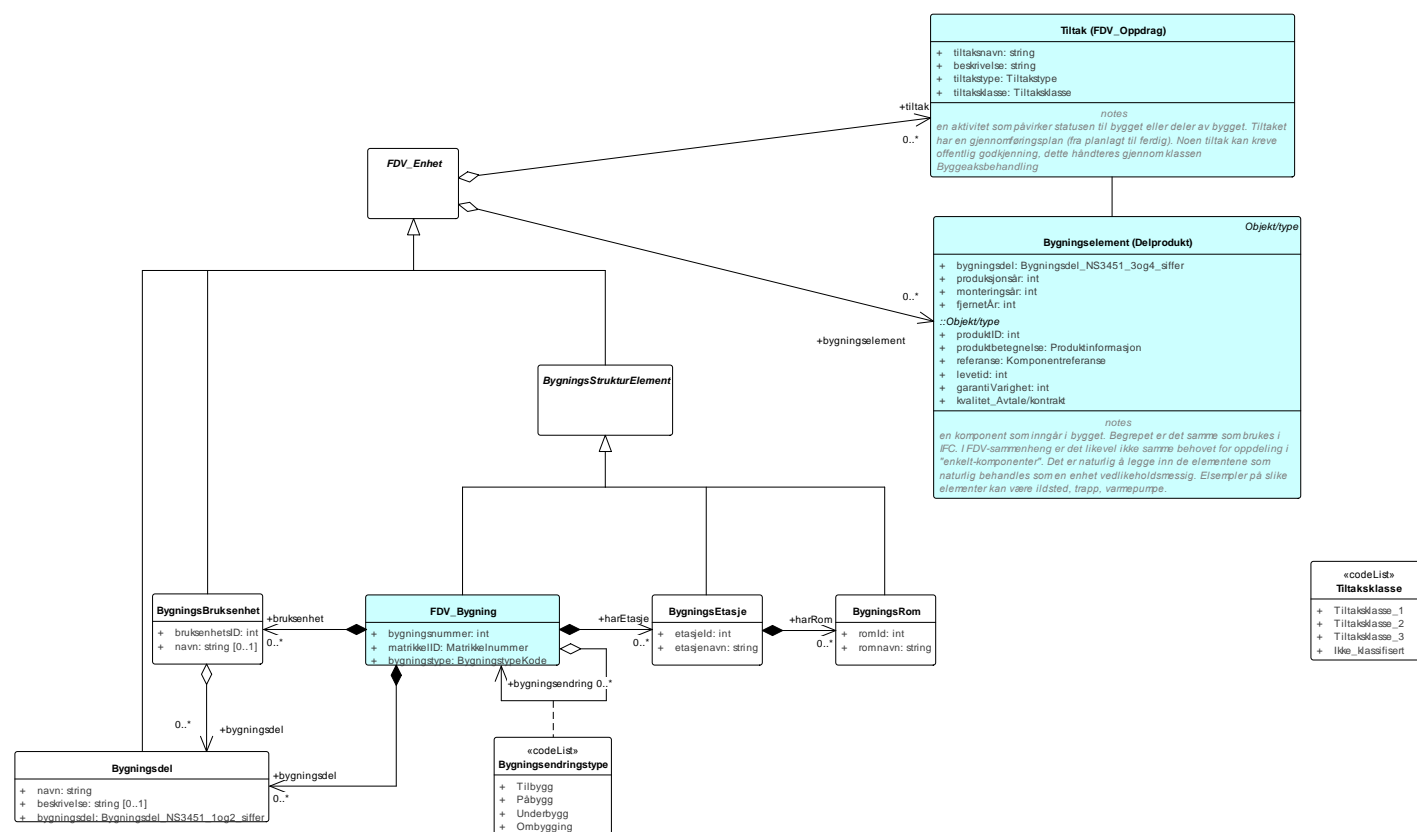
bruksenheter kan være en egen leilighet eller lignende. Bruksenheter kan til en viss grad omsettes for seg selv, og vil i mange tilfeller ha egen «FDV-mappe».

For å håndtere de ulike slags enheter som FDV-dokumentasjon kan knyttes til, er det opprettet en abstrakt objekttype «FDV_Enhet». Denne skal ikke forekomme i noe datasett, men er kun en felles overbygning.

Dette gir mulighet for å kunne knytte FDV-dokumentasjon ikke bare til hele bygningen, men også til hvert bygningsstrukturelement, og til bygningsenheter.

Ved opp-pussing av badet, kan en registrere badet som forekomst av Bygningsrom, og logisk knytte informasjon til det ene rommet.

2.3 Hvordan knytte informasjon til «bygningstrukturen»



Figur 4 Tiltak og Bygningselement

To sentrale objekttyper i modellen er Tiltak og Bygningselement.

- **Tiltak** er tatt inn fra plan- og bygningsloven, og beskriver der oftest fysiske tiltak som forandrer bl.a. bygninger.
NB! Det er ikke uten videre problemfritt å bruke samme betegnelse (Tiltak) i FDV-sammenheng som i plan- og bygningslova. De to typene tiltak har noe til felles, men er ikke identiske. Derfor er det her benyttet betegnelsen «Tiltak (FDV_Oppdrag)»
- **Bygningselement** er et begrep tatt fra BIM/IFC. Det er en samlebetegnelse på alle komponentene en setter sammen, og som til sammen utgjør en bygning.

Et tiltak er knytta inn til bygningsstrukturen, i figuren over til objekttypen FDV_Enhet.

Et tiltak kan føre til at nye bygningskomponenter blir «montert». Noe av disse er det rimelig å kunne legge inn i FDV-basen som forekomster av Bygningselement.

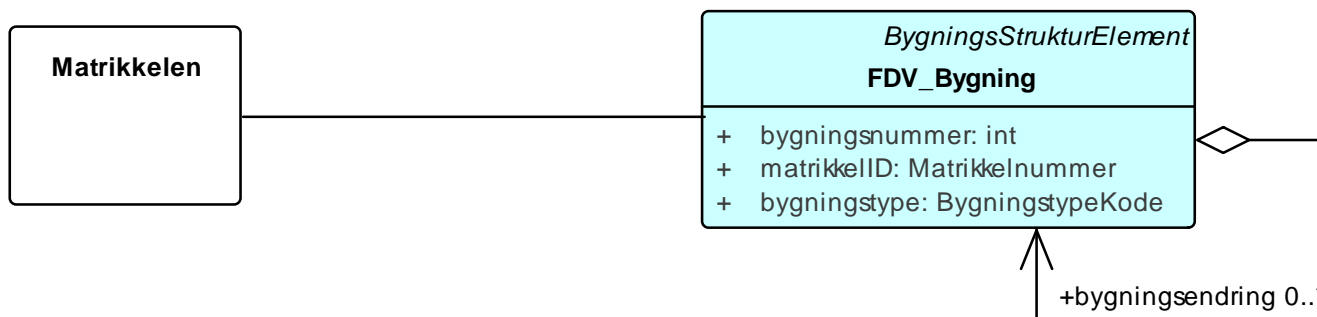
EN FDV_Bygning har også i modellen en «selvassosiasjon». Det betyr at en FDV_Bygning kan bestå av andre FDV_Bygning-forekomster. De vil da være knytta sammen via rollen «bygningssendring». En bygningssendring kan modelleres som en FDV_Bygning og knyttes til «hoved-bygningen». I den tilknyttede kodelista er det oppgitt de 4 endringstypene som finnes i Matrikkelen.

Merknad: behovet for håndtering av bygningssendringer ble diskutert i møte 24.10. Det ble vist til begrep brukt i matrikkelen (underbygg, tilbygg, påbygg,...). Usikkert om det er behov for denne kompliseringen av modellen nå.

3 Brukseksempel

For å forklare hvordan modellen kan brukes, er det nyttig med noen eksempler:

3.1 Eksempel 1 Hente informasjon fra matrikkelen.



Figur 5 Eksempel 1 Matrikkelinformasjon

Dette kan gjøres automatisk, og modellen har plass for å lagre det som kommer.

Modellen sier nå ingen ting om hva slags informasjon som skal hentes fra Matrikkelen og hva som skal gjøres med denne informasjonen. Men modellen sier at det er en knytting som kan utnyttes.

Modellen inneholder også assosiasjoner til andre eksterne registre, som kommunalt planregister og Det offentlige kartgrunnet. Dette er gjort på tilsvarende måte som for matrikkel-koblingen.

Eksterne informasjonskilder kan også være byggesaksdokumenter fra kommunens saksbehandling, eller tilsynsrapporter etter offentlige tilsyn. Slik dokumentasjon kan være naturlig å knytte til **FDV_Bygning**. Det kan også være naturlig å knytte det til **Tiltak (FDV_Oppdrag)** eller til **Bygningsdel**.

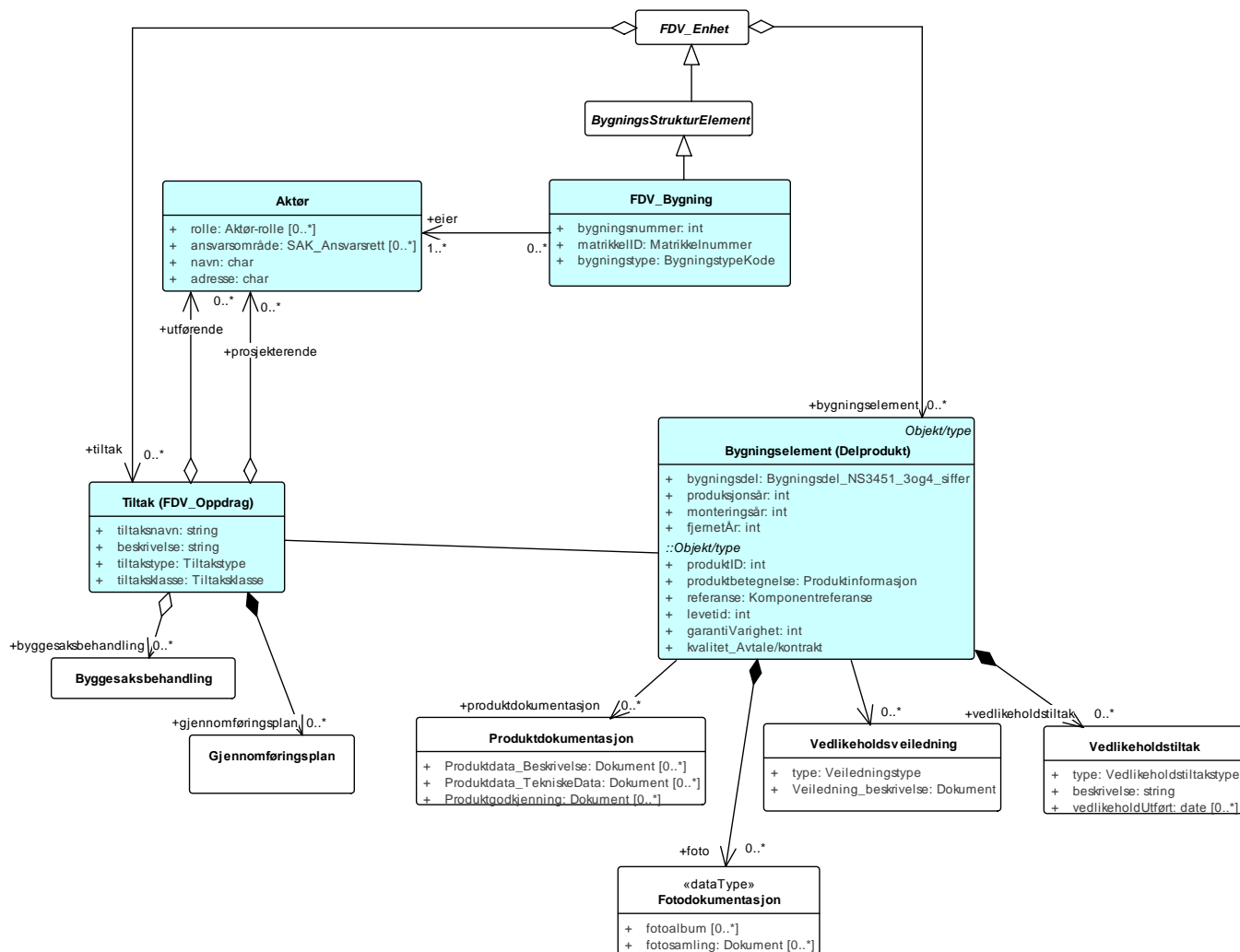
3.2 Eksempel 2 Opp-pussing av bad

Badet i bygningen er pusset opp, og opp-pussingen skal dokumenteres. Selve opp-pussingen beskrives som et eget **Tiltak (FDV_Oppdrag)** tilknyttet bygningen.

I tilknytningen til tiltaket, kan det legges inn opplysninger om **byggesaksbehandlingen** (i alle fall tillatelses-vedtaket). Det kan også legges inn informasjon om **gjennomføringsplanen**.

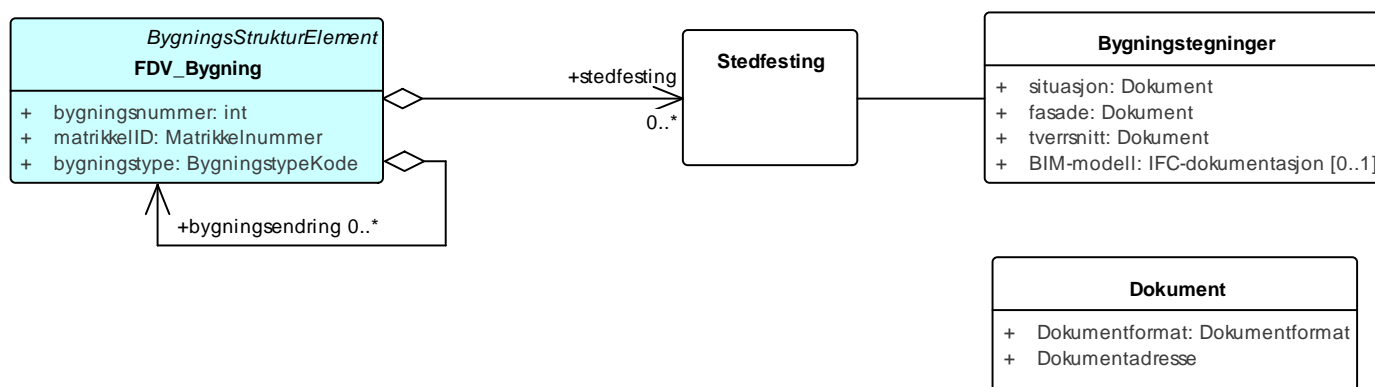
Som del av arbeidet er det installert en ny varmtvannsbereder. Denne legges inn som nytt **bygningselement**. Det kan legges inn mange slags opplysninger om bygningselementer, bl.a. **produkt dokumentasjon, fotodokumentasjon, vedlikeholdsveiledning** og **vedlikeholdstiltak**. Den kan også ha «kvalitetsopplysninger» om garantier og levetidsinformasjon. Aktørene involvert i tiltaket kan angis med **prosjekterende** og **utførende**, enten som foretak eller som fysisk person.

Det er i modellen ikke gjort noe for å skille på drift, vedlikehold og utskifting. Et slikt skille vil i noen tilfeller være nyttig, og bør vurderes tatt inn.



Figur 6 Eksempel 2 Opp-pussing av bad

3.3 Eksempel 3 Legge inn bygningstegninger

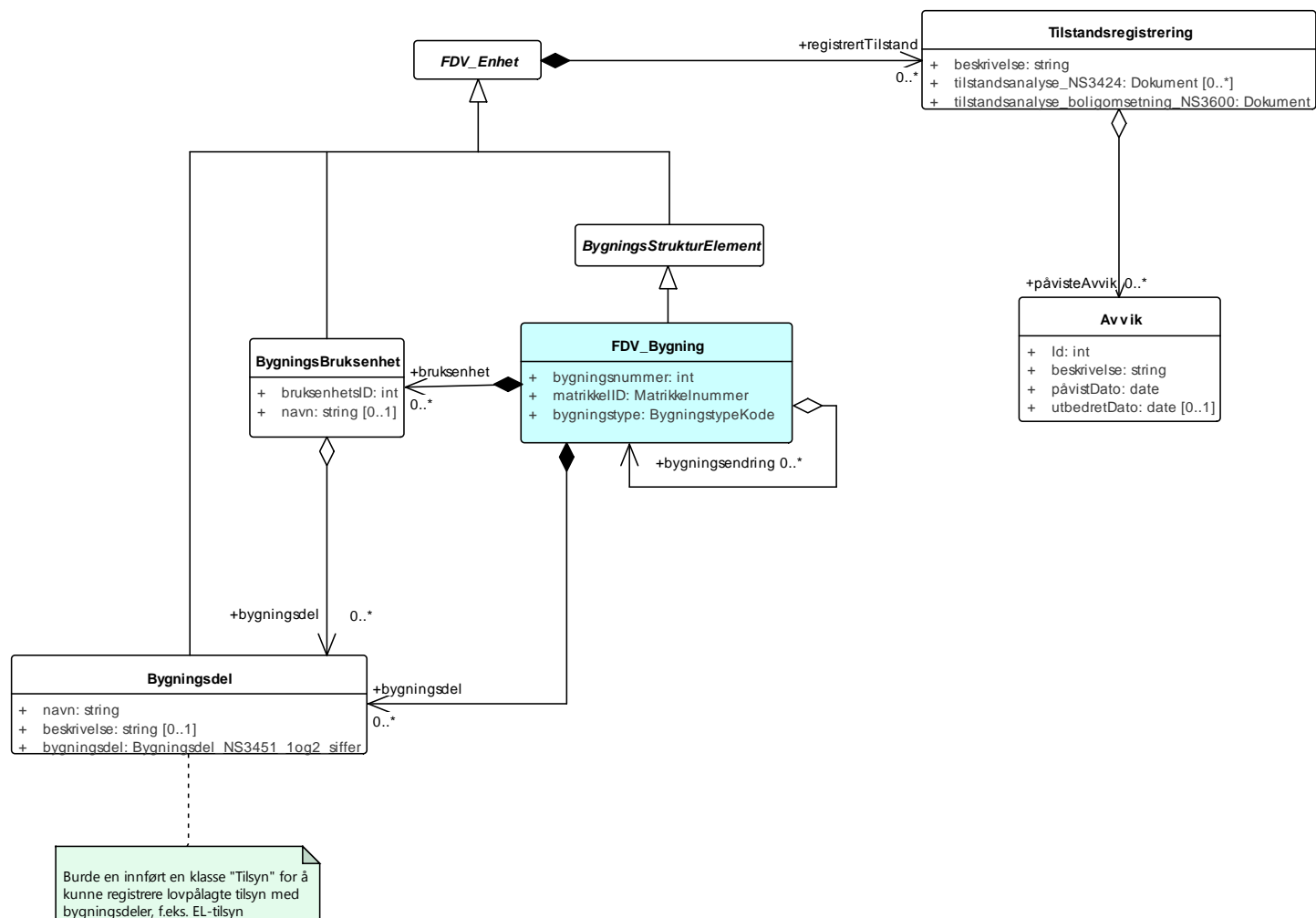


Figur 7 Eksempel 3 Bygningstegninger

Det er «dukket opp» bygningstegninger som eieren ønsker å legge inn. Disse gjøres om til PDF-filer og knyttes via «stedfesting» til bygningen. Tilsvarende kan gjøres med BIM-modeller.

NB! I en BIM/IFC-modell vil hver enkelt strukturelement og bygningselement ha sin egen identifikasjon. Det betyr at all FDV-dokumentasjon kan knyttes til rett plass i BIM-modellen. Det er ikke gjort forsøk på formell modellering av dette i denne versjonen av FDV/UML-modellen.

3.4 Kontroll av elektrisk opplegg



Figur 8 Eksempel 4 Kontroll av elektrisk opplegg

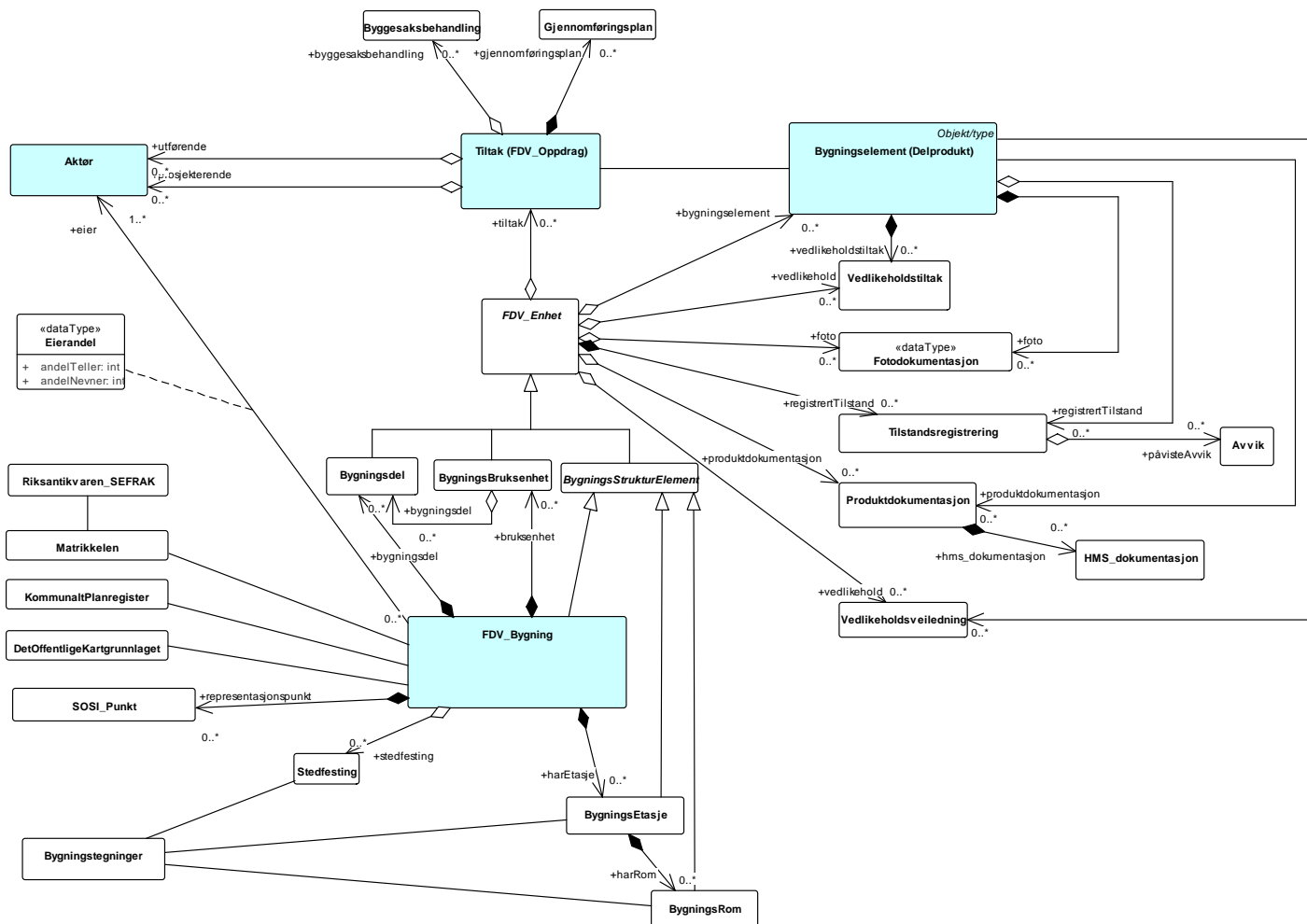
El-tilsynet har vært på besøk og funnet tre avvik som må utbedres.

NB! Modellen har ikke noen logisk oppdeling i «fag». En slik oppdeling ville kunne gjort at en rapport/avviksliste/godkjenning fra El-tilsynet kunne knyttes logisk til «EL-delen av bygningen», Dette kan en også gjøre ved å legge til en kodeliste med «fag» på avvikslista.

4 Bygg_FDV

Klassediagrammet FDV_Oversiktsmodell er satt opp for å gi mest mulig oversikt over modellen. Fargekodingen er ikke del av den formelle UML-syntaksen, kun lagt på for å gjøre det enklere å skille ulike komponenter:

- Gul farge: Komponenter "kopiert" fra andre standarder
- Lys blå farge: Hovedkomponentene i modellen
- Lys grønn: Huskelapper/spørsmål som bør "løses opp i"



Figur 9 FDV_Oversiktsmodell

4.1 Byggekluser - forklaringsdiagrammer

Diagrammene under viser hver sitt utsnitt av modellen.

1. Hovedkomponenter

Dette viser de fire hovedkomponentene bygning, bygningselement, tiltak og aktør. Disse vises med blå bakgrunn på klasse-symbolet.

2. Stedfesting

Viser ulike metoder å stedfeste bygningen på, og også ulike måter å knytte bygningen til utfyllende stedfesta informasjon i matrikkelen, kommunale arealplanregister og DOK (Det kommunale kartgrunnlaget). Det viser også hvordan bygningstegninger kan håndteres, enten ved egne "fasade/tverrsnitt-filer" eller ved hjelp av en mer detaljert BIM-modell.

3. FDV_Bygning

viser hvordan en bygning er modellert. Dette er kraftig "inspirert" av IFC. I IFC finnes BygningsStrukturElementer, som enten er Bygning, BygningsEtasje eller BygningsRom. Disse henger logisk sammen, som vist på figuren. Til hvert strukturelement kan det knyttes egne Bygningselementer.

4. Bygningselement

Bygningselement er en viktig del av modellen. Det er her mesteparten av FDV-dokumentasjonen knyttes til:

- Kvalitet: Sier noe om garantier og levetid
- Produktdokumentasjon er produsentens dokumentasjon av elementet
- Vedlikeholdsdokumentasjon er dokumentasjon av hvordan produktet skal vedlikeholdes
- Vedlikeholdstiltak viser hva som bør gjøres av vedlikehold.

5. FDV-aktør

Diagrammet viser hvordan alle personer og foretak som har et forhold til en FDV-bygning modelleres. Hvilken rolle de har og hvilken godkjenning de ulike aktørene har, går fram av tilknyttede kodelister.

6. SAK Ansvarsrett

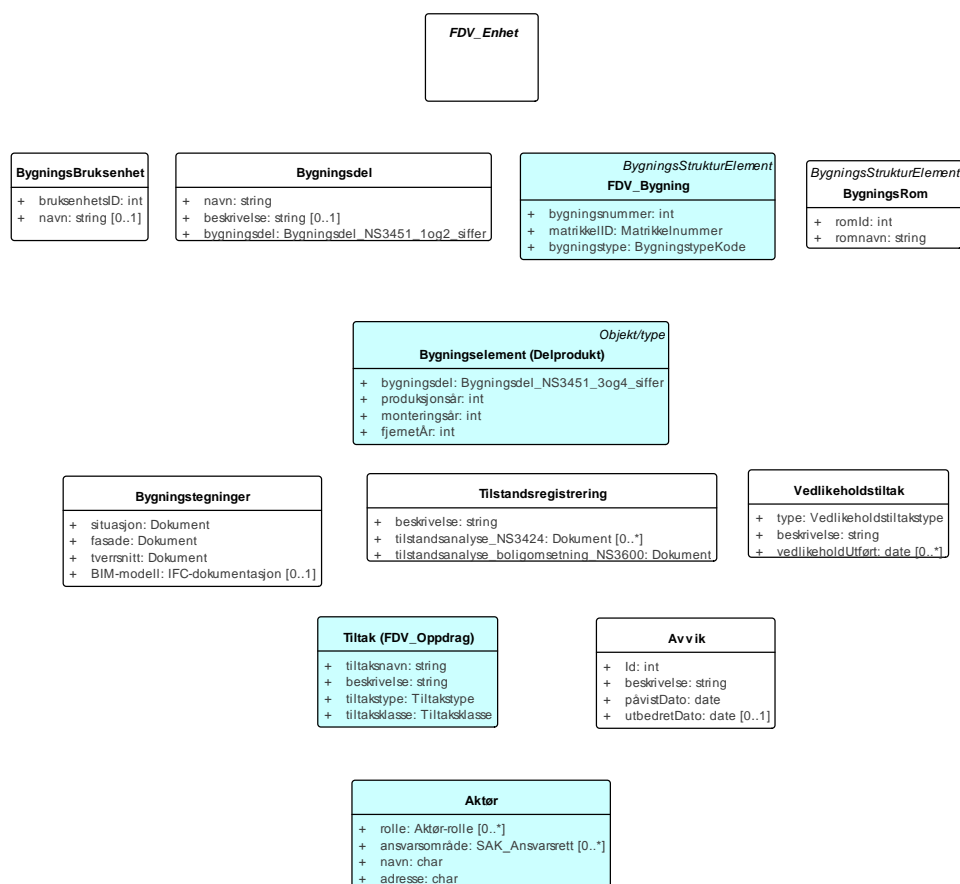
Kodelistene for avsvarsrett er tatt fra SAK 10 §|13-5. Der er ansvarsretten delt i 5 ulike typer, nummerert fra 1 til 5.

7 Tilstandsregistrering

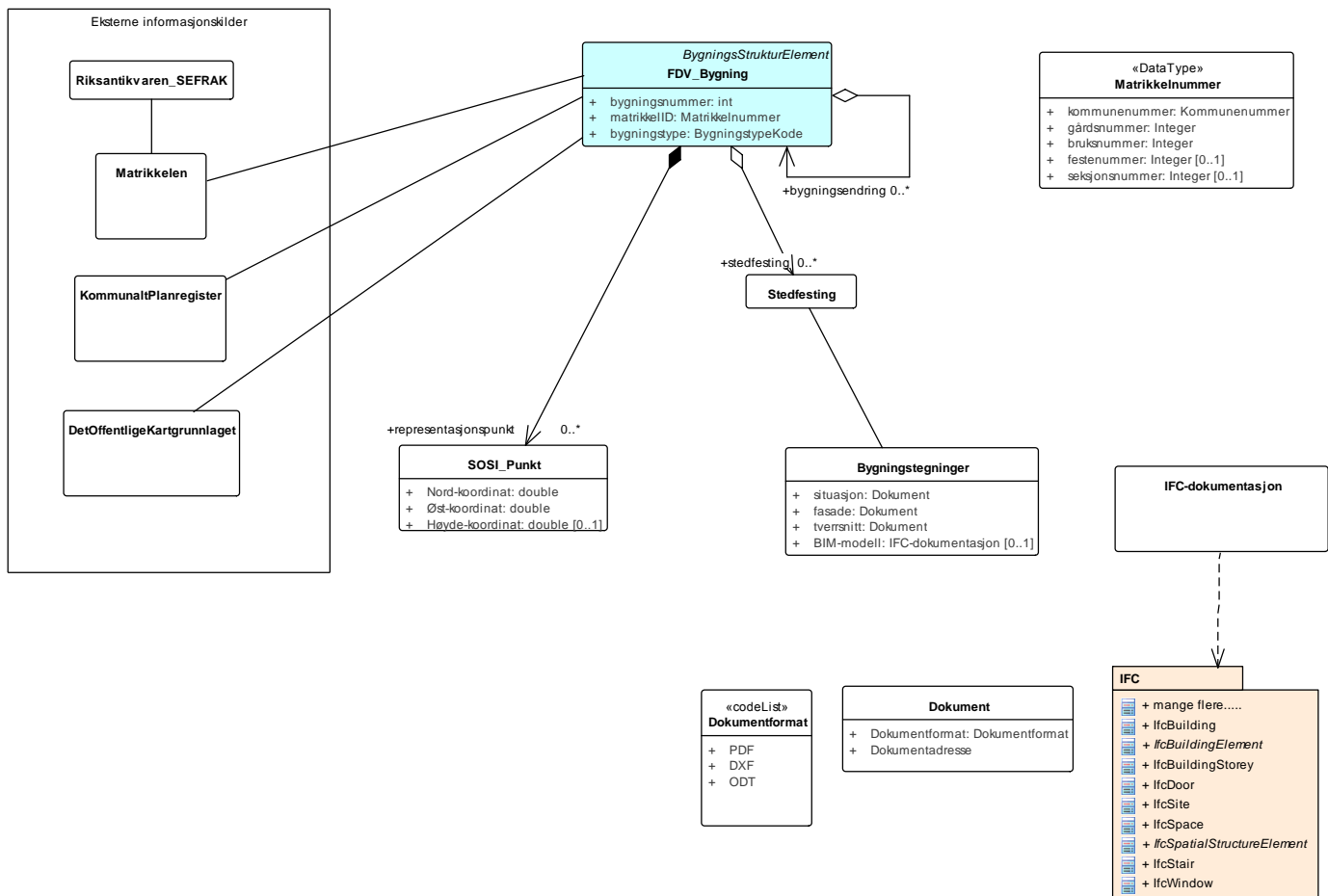
NB! Forstod ikke av tankekartet hvordan dette skal modelleres

8. Tiltak

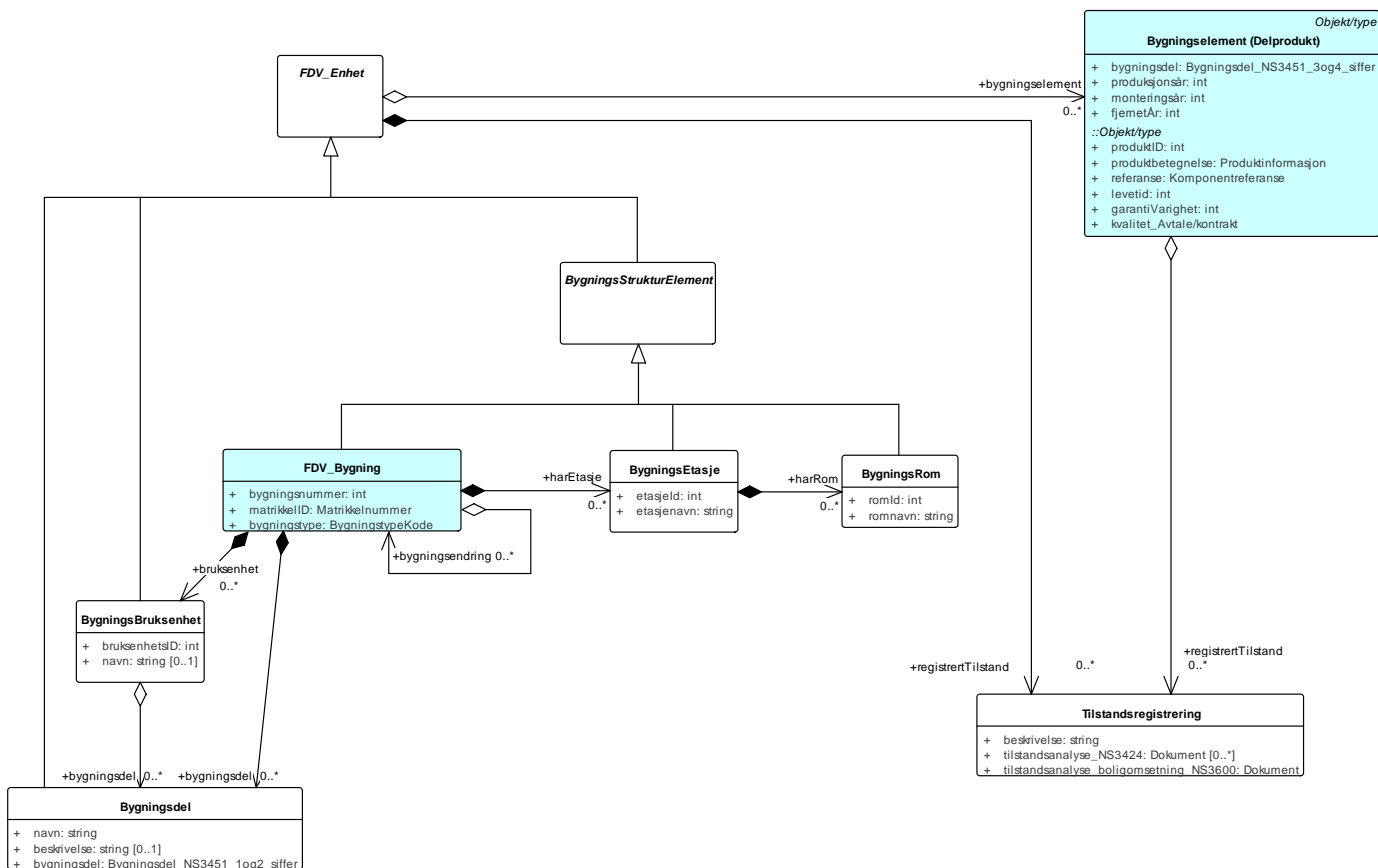
Diagrammet viser informasjonen om tiltakene som gjennomføres for å endre statusen på en bygning.



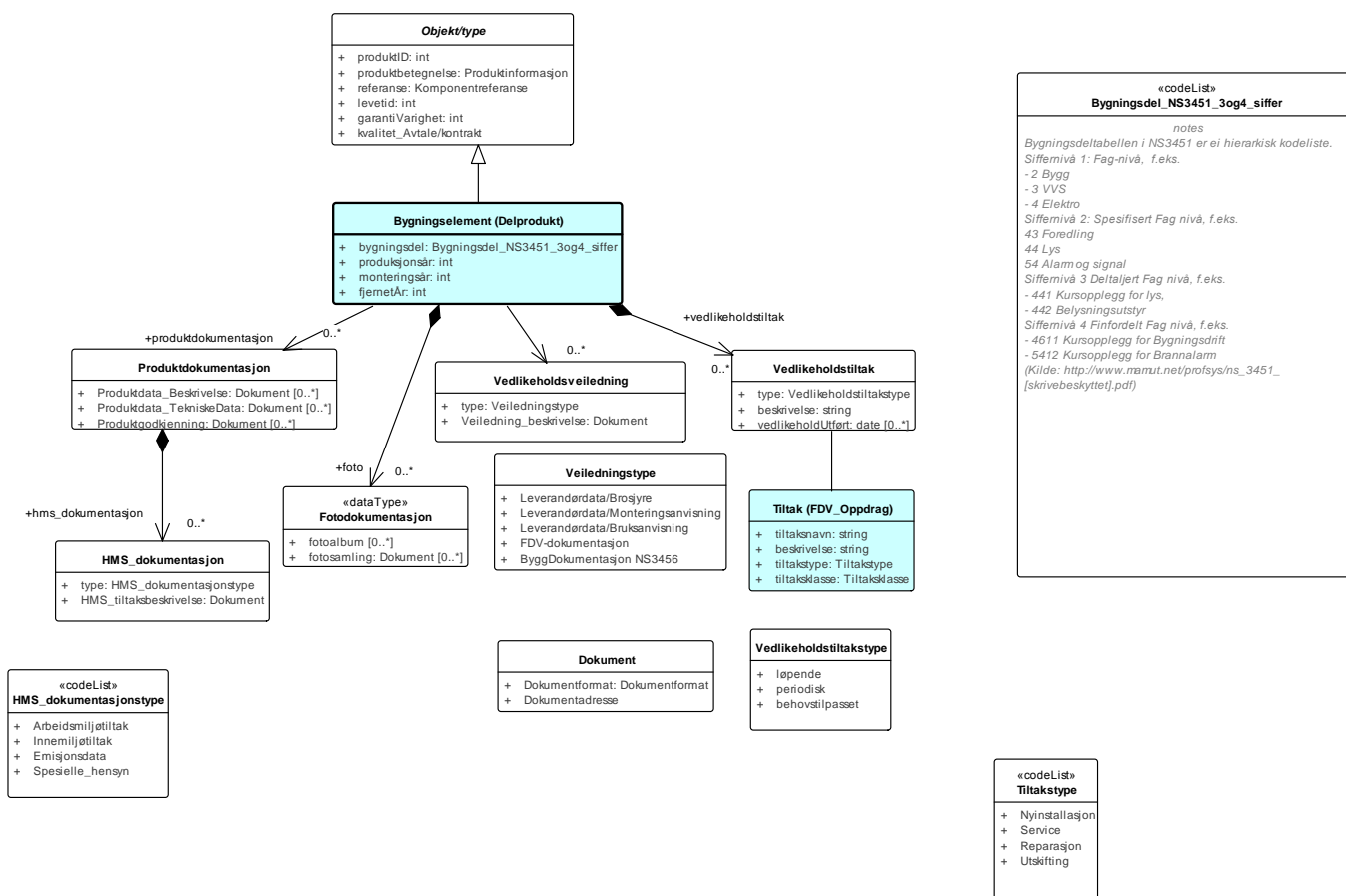
Figur 10 Hovedkomponenter



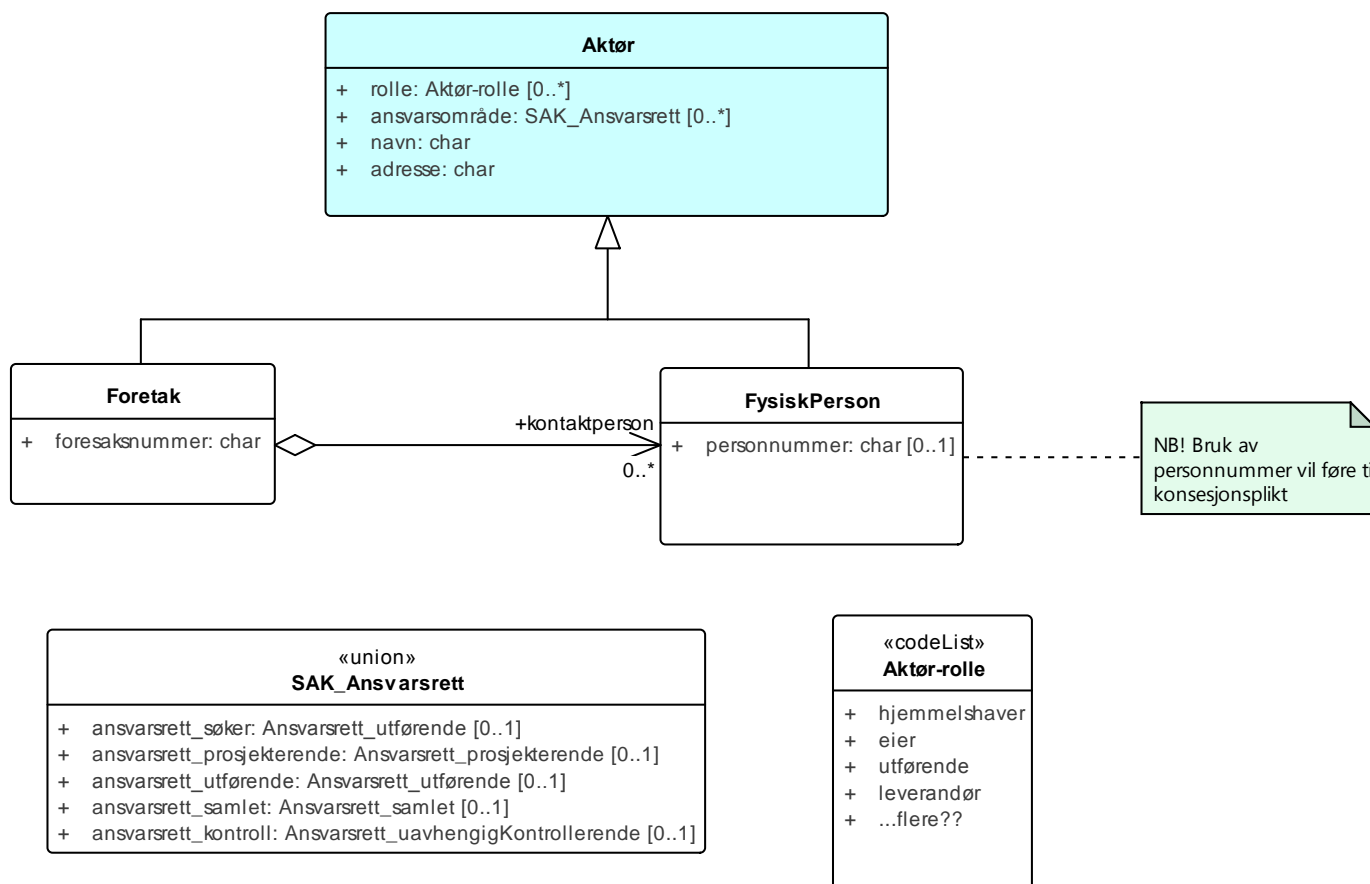
Figur 11 Stedfesting



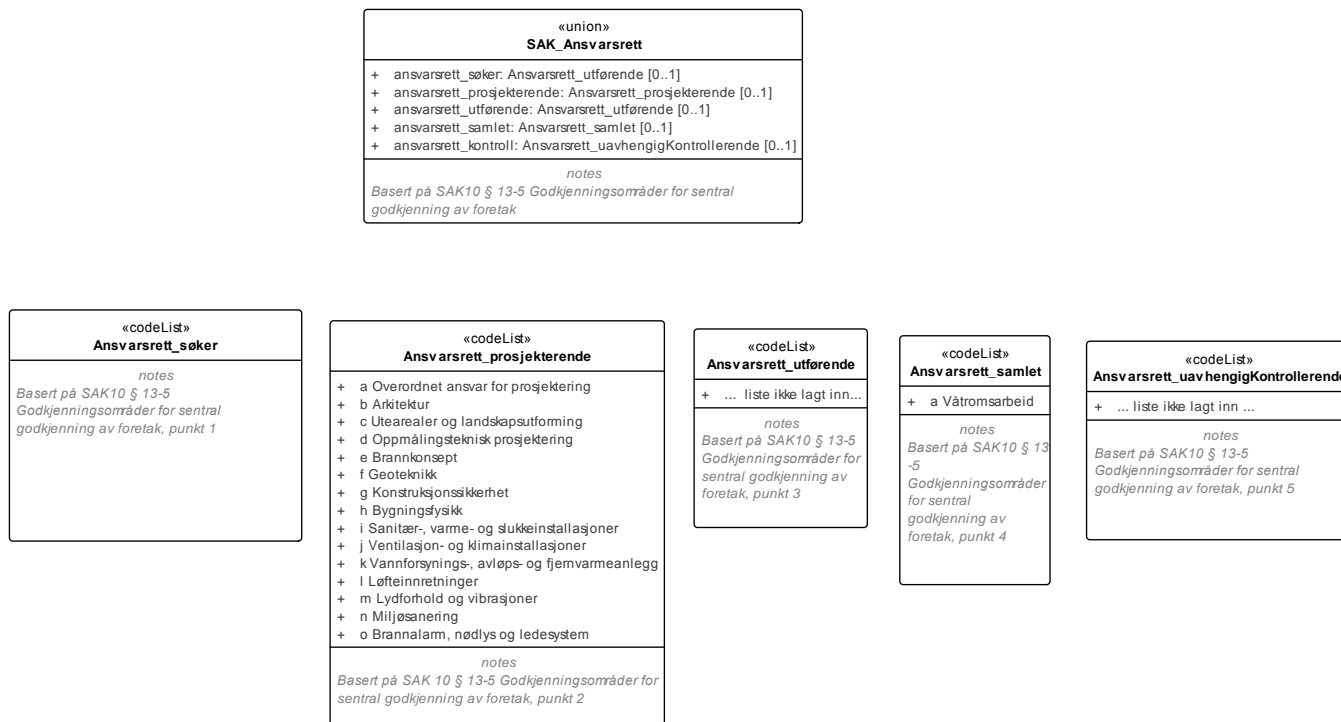
Figur 12 FDV_Bygning



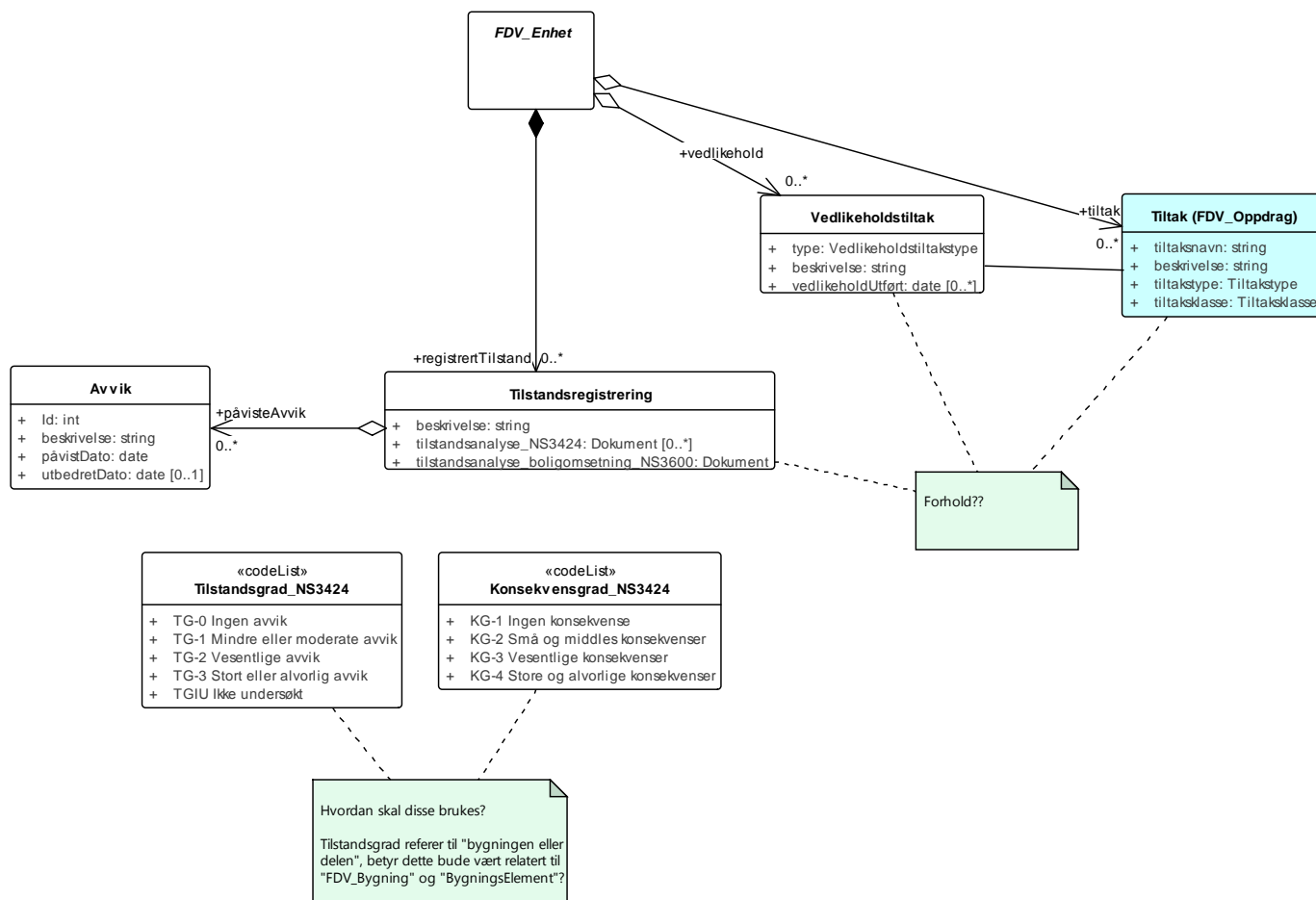
Figur 13 Bygningselement



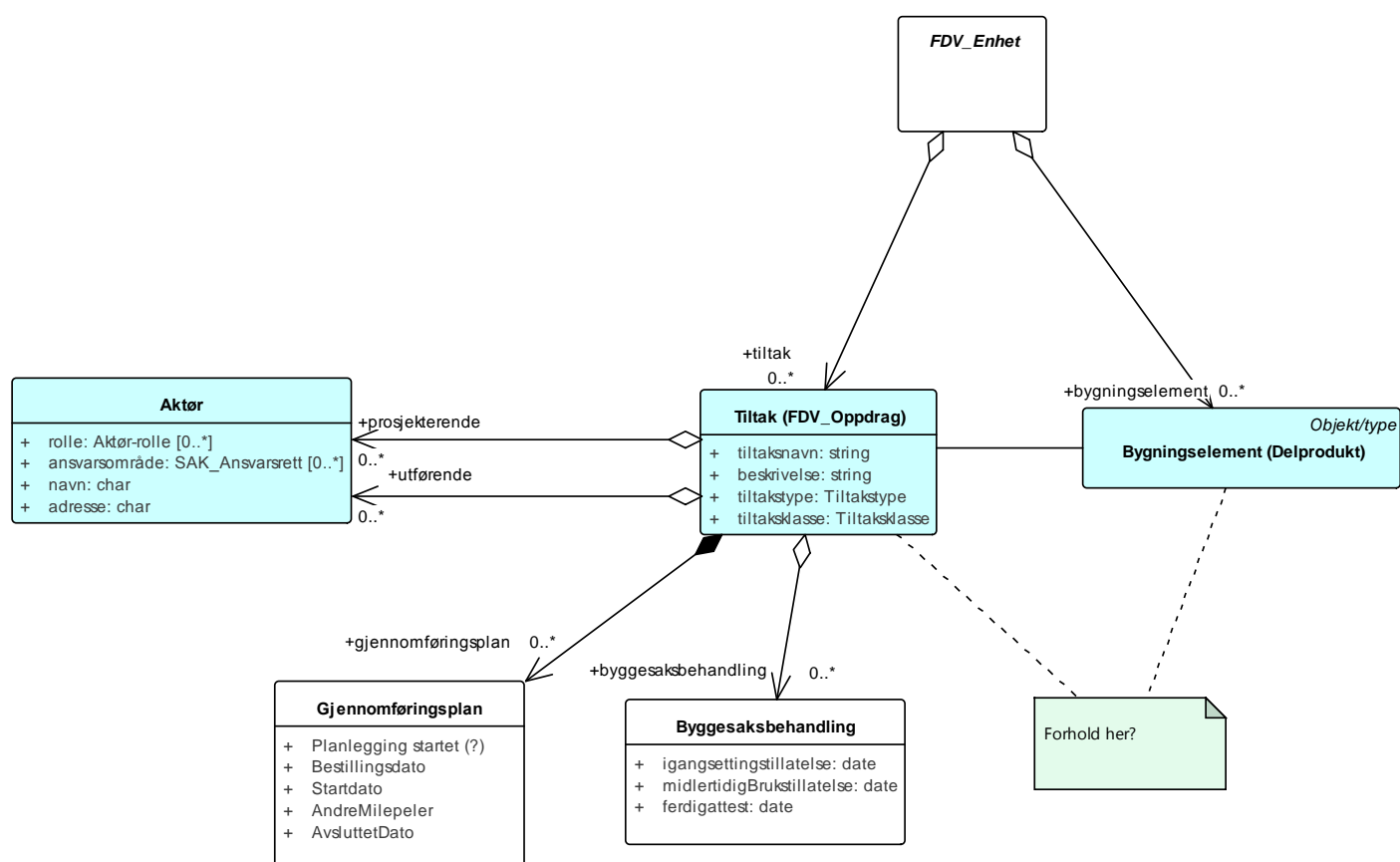
Figur 14 FDV_aktør



Figur 15 SAK_Ansvarett



Figur 16 Tilstandsregistrering



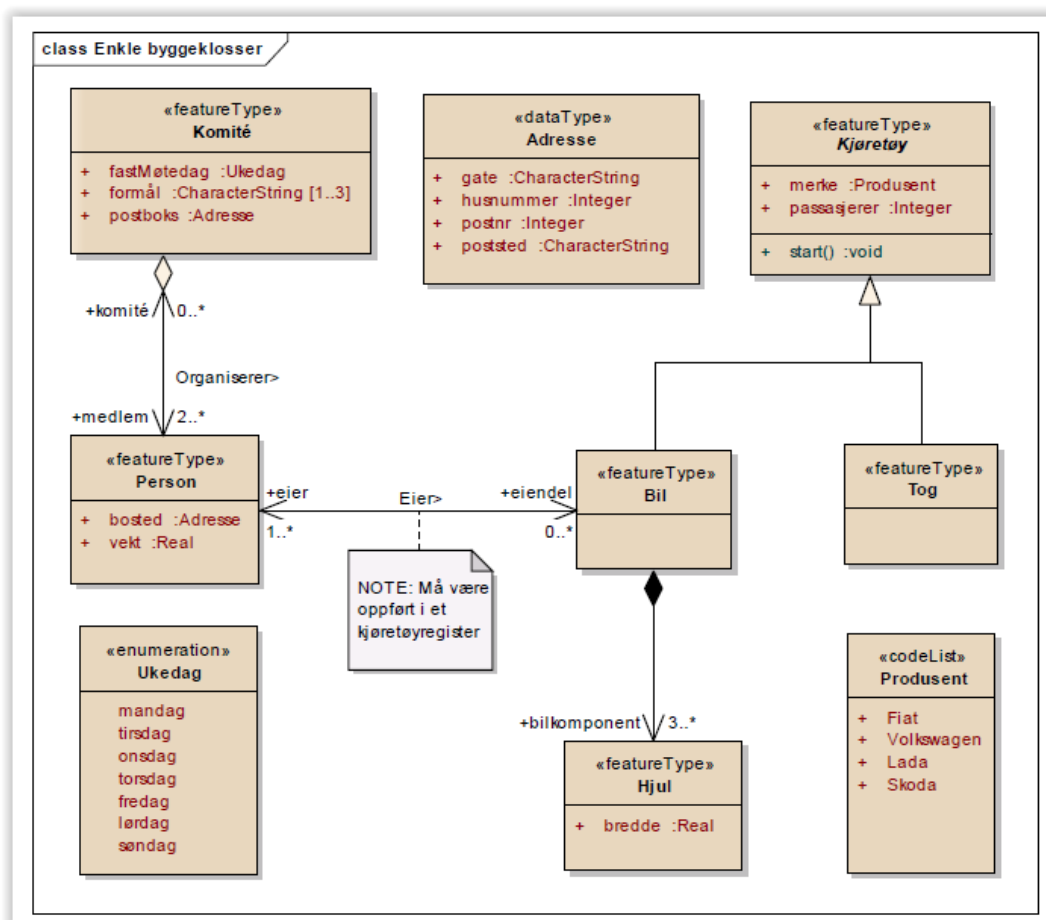
Figur 17 Tiltak

4.2 Full dokumentasjon av UML-modellen

For full dokumentasjon vises til eget, frittstående vedlegg til rapporten, se kap 1.5

5 Intro til UML

Dette er en kort innføring i de mest sentrale begrepene i UML (Unified Modelling Language)



Figur 18 UML-prinsipper (Kilde: www.kartverket.no)

Hovedbegrep	Tilknyttta begrep	Forklaring/eksempel
Klasse	Klassenavn	Eksempel: Tog, Person
	Attributter	Eksempel: bosted og vekt på klasse Person
	Operasjoner	Eksempel: start på klasse Kjøretøy
	Stereotyper	Eksempel: <<featureType>>, <<codeList>>
	Abstrakte	Med klassenavn i kursiv skrift. Klasser som ikke skal forekomme i et datasett, men som er nyttige for å beskrive fellesegenskaper til «underklasser»
Assosiasjoner	Generalisering	«er en»-assosiasjon. Eksempel: Tog er et kjøretøy
	Komposisjon	«består av», vises med åpen diamant i assosiasjonsenden. Eksempel: Personer er del av en Komite
	Aggregering	«
	Rolleravn	
Multiplisitet		Hvor mange ganger en egenskap kan forekomme for hver forekomst, eller (for assosiasjoner) hvor mange forekomster av en klasse som kan assosieres. Angis med to verdier: min og maks. Eksempel: <ul style="list-style-type: none"> 0..* : min 0, maks uendelig 1..* : min 1, maks uendelig (dvs påkrevd)

		<p>egenskap)</p> <ul style="list-style-type: none">• 0..1: min 0, maks 1 (dvs frivillig egenskap)• 1: min 1, max 1: Skal alltid finnes en og bare en gang,
--	--	---

For mer forklaring på UML-begreper, se <http://www.uml-diagrams.org/class-reference.html>