

Modellering med UML som virkemiddel for økt interoperabilitet – anvendelse innen støtte til anskaffelse av nye kampfly

Morten Østbø

Forsvarets forskningsinstitutt/Norwegian Defence Research Establishment (FFI)

21. august 2007

FFI-rapport 2007/01958

1026I

ISBN 978-82-464-1238-2

Emneord

Interoperabilitet

Kampfly

Modellering

Godkjent av

Arne Petter Bartholsen

Prosjektleder

Johnny Bardal

Avdelingssjef

Sammendrag

Transformasjonen av Forsvaret foregår i en stadig mer kompleks og multinasjonal sammenheng. Ønsket om integrasjon i de operative systemene fører med seg et behov for integrasjon mellom et stort antall prosjekter og fagmiljøer som støtter fremskaffelser og konseptutvikling. Interoperable prosjekter og fagmiljøer er en forutsetning for vellykkede hurtige endringsprosesser i et "system av systemer".

Denne rapporten diskuterer hvordan systematisk modellering av operasjoner, systemer og relaterte prosjektaktiviteter ved hjelp av et grafisk modelleringsspråk kan bedre interoperabilitet. Grafiske modeller som er lagret i databaser gir effektiv og lettvinntilgang til informasjon. De gir hurtig oversikt og detaljinnsikt, og sporbarhet fra toppnivået (doktrine) ned til tekniske detaljer og krav. Håndtering av kompleksitet og opprettholdelse av konsistens i endringsprosesser lettes betydelig ved bruk av helhetlige modeller sammenlignet med tallrike tekstdokumenter.

Modeller bør være standardiserte i form, strukturerte og lett tilgjengelige for best å støtte samarbeid og kommunikasjon. Det eksisterer en rekke ulike grafiske modelleringsspråk og rammeverk som spesifiserer innholdet i modeller. UML (Unified Modeling Language) er bedømt å være det mest relevante språket for bruk i vårt prosjekt, sammen med NAF (NATO Architecture Framework).

Modellering med UML iht NAF er et nytt felt. UML og NAF overlater en hel del valg til modellereren. Denne valgfriheten reduserer potensielt kompatibiliteten til ulike modeller. Bruk av UML til å produsere modeller iht NAF krever dessuten en lengre opplæringsstid enn de etablerte arbeidsverktøyene. En UML profil for NAF og en kortfattet veiledning bør derfor utvikles og gjøres tilgjengelig så snart som mulig. Et felles modellbibliotek bør etableres. Eierskap til modeller bør plasseres hos organisasjoner, ikke hos enkeltprosjekter som er tidsavgrenset.

Denne rapporten dokumenterer arbeidet med å etablere en modell for kampflysystemet ved bruk av UML iht NAF. Rapporten inneholder en rekke eksempeldiagrammer fra modellen. Modellen fokuserer foreløpig på nærstøtteoperasjoner, og skal støtte prosessen med anskaffelse av nye kampfly. Hensikten er å bidra til å innføre metodikk som skal fremme interoperabilitet i Forsvaret.

Erfaringene fra arbeidet er at den beskrevne metodikken er relativt lett implementert og har stort potensiale. De store gevinstene for forsvarsrelaterte miljøer vil bare komme når et stort antall aktører slutter seg til satsingen på modellering.

English summary

Transformation of the Norwegian Defence Force is taking place in an increasingly complex and multinational context. Successful transformation requires changes in the way projects communicate, coordinate and make information available. Interoperable systems and concepts require a multitude of development projects and acquisitions processes to be interoperable.

This report discusses how modeling operations and related project activities using a graphical modeling language may improve interoperability. Graphical models that are developed and stored in a database provide efficient and easy access to information. They also provide traceability from the top operational/political level to the technical detail level. Managing complexity and maintaining consistency during change is made easier through the use of graphical models rather than numerous and voluminous text documents.

Models should be standardized, structured and available to best support cooperation and coordination. Several standardised graphical modeling languages exist, as well as a number of frameworks that guide the structure of models. The Unified Modeling Language (UML) has been deemed the most relevant language for the work here described, along with the NATO Architecture Framework (NAF).

Modeling with UML according to the NAF is a new field, and thus immature. The real benefits to the defence community as a whole will only be realized once most of the community has agreed to implement a given modeling method, integrated it in their work processes and agreed upon the detailed ways in which to implement it. The UML language and the NAF leave a great deal of choices up to the modeller, thus limiting the compatibility of the models they produce. In addition, using the UML to produce NAF architecture products requires a longer learning period than many other common office tools. A UML profile and a simple "how to" guide" must be developed and made available. Common model repositories must be established, and ownership of models must be allocated to organisations, not individual projects which are time-limited.

This report documents an effort to apply the Unified Modeling Language and the NATO Architecture framework in support of the forthcoming combat aircraft acquisition in Norway. Systems and activities related to "close air support operations" (CAS) have been modelled as a test case to explore the possibilities posed by systematic and disciplined modeling. The report contains example diagrams from the model.

The main conclusion of the work is that modeling systems, operations and project activities with the UML according to the NAF is a reasonably easily implemented and useful tool. Once a sufficient number of actors within the defence community have embraced modeling as an everyday tool, the potential for positive impact on interoperability seems great.

Innhold

1	Innledning	7
2	Målgruppen	7
3	Målsetting	8
4	Systemarkitektur og interoperabilitet	8
4.1	Definisjonen og nytten av systemarkitektur	8
4.2	Behovet for problemstrukturering	10
4.3	Informasjonstilgjengelighet	11
4.4	Nye muligheter i samarbeidsform	12
5	Arkitekturmodeller	13
6	Arkitekturrammeverk	14
6.1	NAF – NATO Architecture Framework	15
7	Modelleringspråk	18
7.1	UML – Unified Modeling Language	19
7.1.1	UML symboler og diagrammer	19
8	Metamodeller og profiler	23
8.1	Metamodeller	23
8.2	Profiler	24
9	Modelleringsverktøy	25
9.1	Enterprise Architect	26
10	Kampflymodellen	27
10.1	Ambisjonsnivå	28
10.2	Valg av rammeverk, modelleringspråk og programverktøy	29
10.3	Modellstrukturen	30
10.4	To UML-tekniske problemstillinger	31
10.4.1	Modellering av mennesker i kampflysystemet	31
10.4.2	Representasjon av kampflyet i UML	32
10.5	Metamodell og UML-profil for kampflymodellen	33
10.6	Eksempeldiagrammer fra modellen	36
10.6.1	NAV-1	36
10.6.2	NAV-2	37

10.6.3	NOV-1	37
10.6.4	NOV-2	38
10.6.5	NOV-4	40
10.6.6	NOV-5	40
10.6.7	NSV-4	44
10.6.8	NSV-5	46
10.6.9	NSV-8	47
10.6.10	NTV-1	48
10.6.11	Kravdiagrammer	48
11	Modellering i systemteknikk ("Systems Engineering")	50
11.1	Fra dokumentsentriske til modellsentriske prosesser	52
11.2	Arkitekturmodellering som støtte til PRINSIX	53
11.3	Ansvar og roller	54
11.4	Felles modellbibliotek	54
11.5	Begynnerterskelen	55
11.6	Gradering	56
11.7	Felles forum for systemarkitektur i Norge	56
11.8	Implementasjonsplan og opplæring	57
	Forkortelser	58
	Referanser	59

1 Innledning

Effektivt militært samvirke og integrasjon er et sentralt tema i sammenheng med den pågående transformasjonen av Forsvaret. Temaet er relevant i forbindelse med både militære operasjoner og anskaffelsesprosesser. Den sterke vektleggingen av samvirke og integrasjon er motivert ut fra militærfaglige, politiske og økonomiske hensyn.

Økt effektivitet i samvirke og høyere grad av integrasjon er ensbetydende med styrket ”interoperabilitet”. Oppnåelse av styrket interoperabilitet krever, blant annet, fokusering på hvordan et stort antall ulike prosjekter og fagmiljøer – sivile og militære – kommer frem til tekniske og operative løsninger gjennom samarbeid. Koordinerte endringsprosesser, gitt stor kompleksitet, er nøkkelen til interoperable systemer og prosesser.

Kampflyprosjektene i Forsvaret og på FFI har ambisjoner om å være en viktig drivkraft for interoperabilitet og for realisering av NbF. Som deloppfyllelse av denne ambisjonen søker kampflyprosjektet på FFI å fremme en større bevissthet rundt arbeidsmetoder og arbeidsprosesser, da disse har sterk innflytelse på interoperabilitet i de systemene som anskaffes.

Denne rapporten oppsummerer kampflyprosjektets innledende arbeid med systemarkitektur, inkludert modellering i UML (Unified Modeling Language). Modellering er tenkt å fremme interoperabilitet gjennom å effektivisere kommunikasjon mellom aktører som videreutvikler Forsvaret, samt lette strukturering av komplekse problemstillinger. Arbeidet er en konkret oppfølging av en generell behandling av temaet interoperabilitet og nye kampfly [1].

Arbeidet har blitt utført under FFI-prosjekt 1026, Nye kampfly – assistanse til Forsvaret II, delprosjekt 3, Teknologi og arkitektur. Dette delprosjektet har fokusert på luft-til-bakke-operasjoner, herunder spesifikt nærstøtteoperasjoner med kampfly (CAS – Close Air Support).

Denne rapporten består grovt sett av tre deler. Den første delen beskriver fagområdet systemarkitektur og relevante konsepter, dokumenter og verktøy, samt sammenhengen mellom arbeid med systemarkitektur og interoperabilitet. Den andre delen beskriver konkret innholdet i en UML-modell av deler av den eksisterende F-16-arkitekturen, resonnementer rundt valgene som er gjort vedrørende innholdet, metode og verktøy samt tanker rundt videreutviklingen av modellen. Del tre omtaler sammenhengen som denne arkitekturmodellen er tenkt å skulle inngå i.

2 Målgruppen

Målgruppen for denne rapporten er personell som arbeider med utvikling av interoperable systemløsninger for Forsvaret, og med koordinering av aktiviteter på et konkret teknisk nivå.

Det er samtidig ønskelig at beslutningstagere på et noe høyere nivå blir kjent med mulighetene som ligger i den beskrevne metodikken, for dermed å kunne bidra til å samordne og fremme en

hensiktsmessig bruk av modeller som koordinerings-, dokumentasjons- og kommunikasjonsverktøy innen forsvarssektoren.

3 Målsetting

Primærmålet med den nye arbeidsmetodikken som her beskrives er å bidra til effektivt samarbeid mellom aktørene som forbereder anskaffelse av nye kampfly. Dette er tenkt realisert gjennom bedret informasjonsdeling, problemstrukturering, oversikt og kommunikasjon. Modellen som er utviklet i prosjektet er også tenkt brukt som grunnlag for simulering i det nyetablerte simuleringslaboratoriet (Battle Lab) på FFI.

Arbeidet søker også å bidra til etableringen av en hensiktsmessig arbeidsmetodikk og koordinerte prosesser rundt videreutviklingen av Forsvaret som helhet.

Modellering av operasjoner, systemer og prosjektaktiviteter i henhold til arkitekturrammeverk er et umodent fagfelt. Arbeidet i kampflyprosjektet skal etablere kompetanse som vil gjøre det mulig å være en viktig bidragsyter til utformingen av felles løsninger for arkitekturmodellering.

4 Systemarkitektur og interoperabilitet

4.1 Definisjonen og nytten av systemarkitektur

Modellering med standardiserte modelleringsspråk som UML er en metodikk som anvendes innen fagfeltet "systemarkitektur". En tidlig begrepsavklaring er viktig når man snakker om systemarkitektur, da begrepet kan tolkes på flere måter. Tre betydninger er:

1. "Hvordan systemer er organisert, fungerer og brukes" ("slik virkeligheten er")
2. "Hvordan man lager systemer" (en arbeidsprosess)
3. "Beskrivelse av systemer" (tilsvarende begrepet "modell")

Et FFI-prosjekt har brukt følgende formulering:

"Systemarkitektur – kunsten å utforme komplekse tekniske systemer" (P889 FFI).

I IEEE standarden 1471 finner vi denne definisjonen på begrepet "Architecture":

"The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution"

"INCOSE Systems Engineering Handbook" [2] bruker en tilsvarende definisjon av "System Architecture":

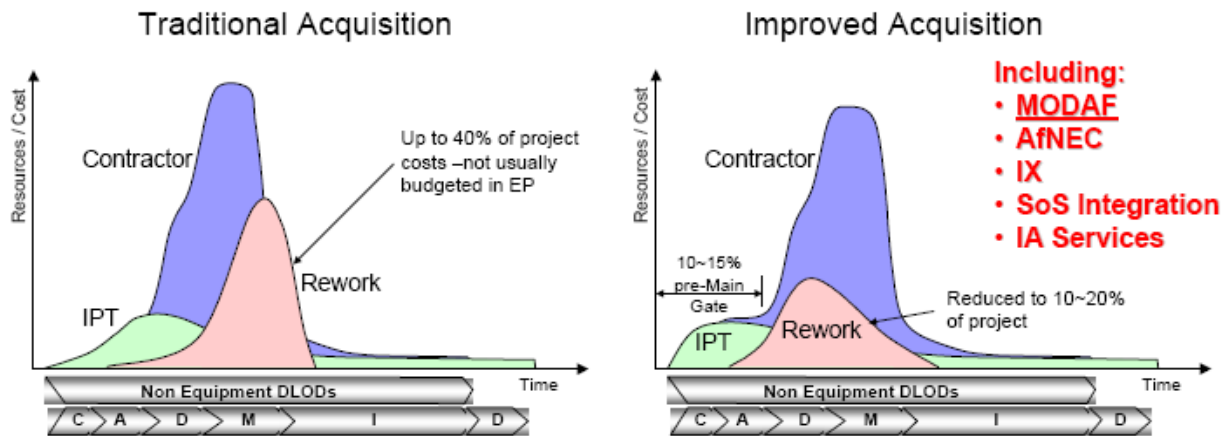
“The arrangement of elements and subsystems and the allocation of functions to them to meet system requirements”

Ordet ”systemarkitektur” kan altså bety både ”slik systemet er”, ”en arbeidsprosess” og ”en beskrivelse av systemet”. I mange tilfeller kan det altså være ”Systems Engineering” det er snakk om, mens andre ganger er det ”arkitekturen til et system” man mener når man sier ordet ”systemarkitektur”. Denne rapporten tar hovedsakelig for seg et konkret tiltak innenfor beskrivelser av systemers arkitektur. Dette tiltaket er ment å effektivisere arbeidsprosessen med å ta frem systemer (av systemer), med endelig hensikt å komme frem til systemer med ønsket grad av interoperabilitet.

For mange virker fagfeltet systemarkitektur diffust. Også nytten av å arbeide med metodikk for beskrivelser av systemer (altså å utarbeide arkitekturmodeller) er høyst uklare for de fleste som ikke har arbeidet direkte med slike oppgaver. Og det er foreløpig få som har hatt dette som arbeidsfelt.

Vanlig argumentasjon for vektlegging av systemarkitektur (i alle ordets betydninger) lyder at en høyere ressursbruk i prosjekters tidlige fase kan synliggjøre potensielle problemer før det er for sent eller dyrt å rette på dem. Samtidig kan man finne bedre løsninger tidlig. Argumentene som finnes i tekster og som kommer frem i samtaler er kvalitative. De er gjerne ”heuristiske”, dvs de bygger på erfaring og rimer som oftest godt med fornuften. Tanken er at tiltak som gir bedre oversikt, reduserer misforståelser, utelatelse mm må være av det gode. Imidlertid er kvantitative og konkrete eksempler på hvordan man omsetter systemarkitektur-teori i praksis mangelvare.

”MODAF (Ministry of Defence Architecture Framework) executive summary” [3] gjør en tilnærmet kvantifisering av innsparingspotensialet gjennom å tallfeste et beløp for ”rework” for et gitt program. Gjennom å øke innsatsen innen integrasjon og interoperabilitet på et tidlig stadium, skal ”rework”-delen reduseres betydelig. Både USA (DoD) og UK (MoD) stiller nå krav til produksjon av strukturerte arkitekturbeskrivelser fra alle kontraktører. Begrunnelsen er at dette skal effektivisere anskaffelsesprosessene og føre til mer interoperable systemer. Et slikt krav ser ut til å ligge noen år frem i tid i Norge. I forbindelse med debatten omkring dette temaet er det et stort behov for å ta frem konkrete eksempler der systemarkitektur og UML har praktisk nytte, og der den nye metodikken representerer en forbedring over andre måter å gjøre ting på. Arbeidet som dokumenteres i denne rapporten er et tiltak som skal legge til rette for å gjøre dette.



Figur 4.1 En visualisering fra MODAF av innsparepotensialet ved å fokusere tidlig på systemarkitektur. "Rework"-omfanget reduseres betydelig.

4.2 Behovet for problemstrukturering

Anskaffelse av nye kampfly er et godt eksempel på den type problemstilling som preger anskaffelser generelt innenfor Forsvaret idag. Det er problemstillinger der det kan synes uklart både hva man skal frem til og hvordan. Hva vil det si å være interoperabel i 2020? Hvilke systemer vil Forsvaret da bestå av, og hvordan skal disse samvirke? Hva betyr dette for kampflyanskaffelsen? Hvilke konkrete krav må vi stille til nytt kampflysystem for at dette skal være interoperabelt? Typen problemstilling som kampflyanskaffelsen står overfor betegnes (kanskje noe spøkefullt) i litteraturen som "a mess" eller "a wicked problem" [4].

"Mess[and wicked problems; red anm.]: situation in which there is considerable disagreement about what needs to be done and why; therefore, it is impossible to say how it should be done. Thus, the mess is unstructured and must be structured and shaped before any solution, should such exist, can be found." ([4], side 7)

Denne typen problemstilling betegnes som en "type 4"-problemstilling i et kurs i systemteknikk som er holdt av Metier for personell i Forsvaret, deriblant en rekke sentrale aktører i kampflyprosjektet i FLO:

"Type 4-prosjektene er de som har minst sjanse til å lykkes. Her finner vi organisasjonsendringsprosesser, hvor både målene og hvordan vi skal nå dem er lite avklart." (M03 Kurs i Systemteknikk, Metier)

At de aktuelle kampflysystemene utvikles av andre nasjoner er i visse sammenhenger en forenkling og i andre sammenhenger en komplikasjon av problemstillingen. Det er en forenkling gjennom at mange løsninger bestemmes av andre. Det er en komplikasjon gjennom at informasjon om løsningene er vanskelig tilgjengelig, og at vi som kunde har begrenset påvirkning på løsningene.

Kampflyanskaffelsen lever i en brytningstid mellom en forholdsvis veldefinert strategisk situasjon med et Forsvar der systemer levde i relativ isolasjon, og en ”ny tid” der sikkerhetssituasjonen gir uklare krav til forsvarsutviklingen, og der systemer ønskes integrert. Samtidig gir ny teknologi betydelig større fleksibilitet i løsningsform. Der man tidligere byttet plattformer med nye, forbedrede versjoner av samme konseptet, ser vi nå en oppløsning av bindingene til løsningsform. Vi står overfor et utvidet løsningsrom, med store vanskeligheter med å definere krav.

Bidragstyttere til anskaffelser i denne nye situasjonen står overfor en meget høy grad av kompleksitet. Vi står overfor to kilder til usikkerhet; naturens og samfunnets stokastiske egenskaper, og mangel på informasjon. Noen hevder at det er den siste usikkerhetskilden som skaper størst problemer i prosjekter i dag [5]. Flere forhold tyder på at realisering av visjonen om NbF krever transformasjon også i måten prosjekter og fagmiljøer jobber og kommuniserer. Disse forholdene er:

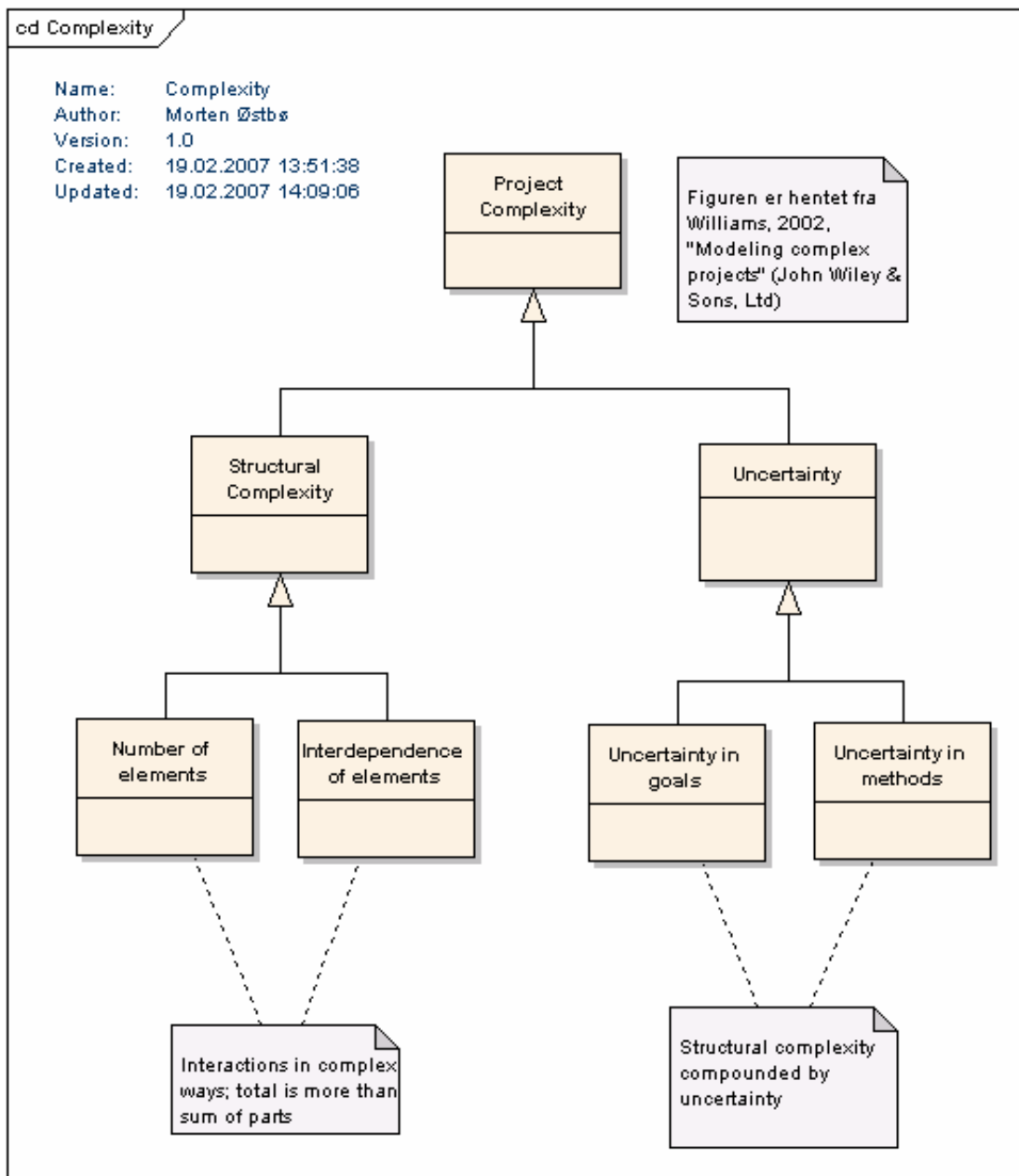
- ❑ Økt kompleksitet i tekniske systemer
- ❑ Økt tids- og kvalitetskritikalitet
- ❑ Flernasjonalitet
- ❑ Visjonen om økt samhandling på tvers av tradisjonsbundne grenser

Mange prosjekter står overfor krav om hurtige resultater innenfor arbeidsfelt som krever dybdekunnskap. Hastighet og dybde kan synes som en umulig kombinasjon i organisasjoner med begrensede ressurser. Nettopp denne situasjonen er en meget viktig motivasjon for økt samhandling ifm fremskaffelser av militære kapasiteter til Forsvaret.

4.3 Informasjonstilgjengelighet

Når de fleste prosjekter ikke lenger jobber med vel avgrensede problemstillinger, oppstår behovet for informasjon om andre prosjekter, og om en lang rekke systemer, teknologier og konsepter som prosjektet skal ta hensyn til. Å få oversikt over nåsituasjonen er den første oppgaven i de fleste tilfeller. Deretter skal man ha oversikt over gjeldende planer og forslag som er aktuelle både på kort og lang sikt. I dag innebærer dette tidkrevende søk etter og bestilling av dokumenter, tallrike samtaler med operativt personell, logistikkmiljøet, ”tenketanker”, industrien mfl. Informasjonsinnsamling er blitt en betydelig del av prosjekters hverdag. Det finnes ingen samlet kilde til oversikt over hvor informasjon finnes, og informasjonen er for det meste i form av lange tekstdokumenter eller ikke dokumentert i det hele tatt.

Det er uheldig om mange prosjekter bruker sine ressurser til å samle informasjon om samme tema parallelt. Det bør være et uttalt mål at prosjekter skal bruke mindre tid på orienteringsfasen, og anvende sine ressurser i større grad på fagspørsmål enn på informasjonssøk. For å nå dette målet, er det nødvendig med god tilgang til oversikt og innsikt som er opparbeidet av andre. UML-modelleringen som denne rapporten beskriver handler om å samle informasjon og gjøre denne tilgjengelig på en strukturert form.



Figur 4.2 "Kampfly og interoperabilitet" er en kompleks problemstilling. Aktørene som er involvert i anskaffelse av nye kampfly må forholde seg til et stort antall gjensidig avhengige aktiviteter, systemer og konsepter, og må takle usikkerhet rundt både teknologi og politiske vedtak. Dette gir opphav til metodiske utfordringer. (Fra [5])

4.4 Nye muligheter i samarbeidsform

Det er stor bevissthet rundt behovet for å koordinere prosjekter. Dette har resultert i at reise- og møtevirksomheten er meget stor. Den store møteaktiviteten synes å være uttrykk for at mennesker opplever at de kommuniserer best med andre mennesker gjennom fysiske møter. Mange har muligens nådd maksimum av det antall reiser som er mulig og hensiktsmessig. Økt samhandling og koordinering bør altså ikke resultere i flere reisedøgn, ideelt sett færre enn idag.

Møter, konferanser osv fungerer godt som kontaktskapere og motivatorer. Dette er en meget viktig funksjon i ”koalisjonsbygging”. Tid til forberedelse foran møter er imidlertid mangelfull med det store antallet møter og reiser. Det kan samtidig oppleves som om mye tid på møter brukes til å utveksle informasjon som lett kunne vært gjort tilgjengelig for deltakerne på annet vis. I tillegg etterspørres ofte informasjon i løpet av diskusjoner, og ofte noteres aksjonspunkter for oppklaring i etterkant. Manglende informasjon under møter fører altså ofte til redusert utbytte av møtet.

Det virker sannsynlig at møter kan gjøres betydelig mer effektive dersom de støttes av lettvinne informasjonstilgang, og dersom de suppleres av andre mer effektive måter å utveksle informasjon mellom møtene.

Hypotesen som fremsettes her er at aktiv bruk av strukturerte beskrivende modeller som dokumentasjons- og kommunikasjonsmedium vil effektivisere samvirke mellom mange av aktørene som bidrar til forsvarsutviklingen. Samvirket vil bli effektivisert delvis gjennom lettere tilgang til informasjon under og mellom møter, og delvis gjennom redusert behov for møter. Dette forutsetter følgende:

- 1) Modellene må være tilgjengelige over et nettverk eller de må være portable.
- 2) Modellene må finnes på en lett gjenkjennelig form, og ulike modeller må lett kunne sammenstilles.
- 3) Modellene må beskrive både systemer, operasjoner, konsepter, prosjekter og sammenhengen mellom disse.

5 Arkitekturmodeller

Med modell menes i dette arbeidet en grafisk ”beskrivelse av noe”. Det er viktig å skille mellom modeller som skal beskrive noe, og modeller som skal etterligne noe. Begrepet ”modell” oppfattes ofte i betydningen ”simuleringsmodell”. Simuleringsmodeller er dataprogrammer som skal etterligne oppførselen til et system, og som gjerne er koblet til programvare for visualisering av denne oppførselen.

En modell av en problemstilling eller et system ligger alltid til grunn når simuleringsmodeller skal utvikles. Imidlertid finnes sjelden disse modellene tilgjengelig på eksplisitt form, eller på noen standardisert form. Dvs de er vanligvis innebygget i selve programkoden og til en viss grad tilgjengelig gjennom tekstdokumentasjonen til programmene.

En modell kan gjerne beskrives som ”en felles kilde til sannheten” (begrepet ”one source of truth” brukes ifm arkitekturrammeverk) eller ”en felles hukommelse”. Felles for gode modeller er at de presenterer informasjon på en ryddig og strukturert måte.

”A model introduces a precise structure and terminology that serve primarily as an effective means of communication, enabling participants in the study to exercise their judgement and

intuition in a well defined context and in proper relation to the judgement and intuition of others... Through feedback from the model, the experts have a chance to revise early judgements and thus arrive at clearer understanding of the problem and its context.” (Sitat av Edward Quand fra RAND. Fra [4], side 180)

Arkitekturmodellene som omtales i denne rapporten er bredt omfavnende modeller som inneholder grafiske beskrivelser som er supplert med tabeller, tekst mm. Diagrammer, tekst og tabeller utgjør til sammen en helhetlig kilde til informasjon.

6 Arkitekturrammeverk

Et arkitekturrammeverk spesifiserer hva arkitekturbeskrivelser skal inneholde. Rammeverket spesifiserer hvilken type informasjon en (komplett) modell skal inneholde, og hvordan denne informasjonen skal organiseres. Arkitekturrammeverkene søker dermed å gjøre modeller kompatible. De sørger også for at en kunde får den type informasjon som kreves fra en tilbyder (og motsatt) gjennom en standardisert og helhetlig ”informasjonspakke”.

Noen arkitekturrammeverk legger ned regler for egenskapene til arkitekturbeskrivelser. De fokuserer på sluttproduktet, altså modellene, men nevner også prosessen med å utvikle dem. NATO C3 System Architecture Framework (NAF) [6] er et eksempel på denne typen rammeverk. Andre arkitekturrammeverk gir mer utførlig veiledning i hvordan man utvikler en arkitektur. Dvs de er veiledninger i systemutvikling. Et eksempel på dette er TOGAF – ”The Open Group Architecture Framework”. Dette rammeverket er sammen med ADM – ”Architecture Development Method” – en ”how-to” veiledning.

De tre mest dominerende og mest interessante rammeverkene for vårt bruk er NAF, DoDAF [7] og MODAF [3]. De to sistnevnte er ”Department of Defense Architecture Framework” fra USA og ”Ministry of Defence Architecture Framework” fra Storbritannia. MODAF og NAF har begge sitt utspring i DoDAF. Disse tre rammeverkene har derfor mye til felles.

Både rammeverkene og aktuelle grafiske modelleringsspråk som UML (Unified Modeling Language) [8] har sitt utspring i programvare- og informasjonssystem-verdenen. Derav ”C3” i NAF versjon 2. DoDAF er en videreutvikling av ”C4ISR Architecture Framework”. Imidlertid kan og bør rammeverkene anvendes til å veilede beskrivelser av også andre typer systemer. Informasjonssystemer, kommunikasjonssystemer og komplekse programvaresystemer kan ikke idag skilles fra plattformene i forsvarsstrukturen, da funksjonaliteten til disse systemene i stor grad er realisert av programvare og kommunikasjon. De fleste systemer er idag meget programvareintensive. Samtidig har beskrivelser av kommando-, kontroll- og informasjonssystemer liten mening uten en kobling til de prosesser og aktører som genererer og forbruker informasjon. Dette er anerkjent av gruppen som utviklet NAF v3. ”C3 System” er dermed fjernet fra tittelen i NAF v3.

NAF, DoDAF og MoDAF dikterer ikke bruk av UML eller noe annet spesifikt modellerings-

språk. NAF og DoDAF omtaler imidlertid UML, og gir eksempler på hvordan man kan bruke UML til støtte for representasjon av arkitekturprodukter. Begrepet ”arkitekturprodukter” refererer til eksempelvis diagrammer, tabeller og matriser som fremstiller en viss type informasjon. Egenskapene til disse produktene er spesifisert i rammeverkene.

Et arkitekturrammeverk kan fungere som et nyttig hjelpemiddel for informasjonsstrukturering, og kan dermed også være til stor hjelp når man forbereder informasjonsinnsamling. I de innledende fasene i et prosjekt kan det være vanskelig å ”vite hva man ikke vet”. Rammeverket kan fungere som en nyttig mal når man skal formulere klare spørsmål, hvilket er en av de viktigste oppgavene til en systemarkitekt.

Valg av ulike rammeverk skaper problemer dersom modeller skal integreres. USA bruker DoDAF, UK bruker MODAF og et fåtall andre land bruker NAF. Dette er en god illustrasjon på interoperabilitetsutfordringen. Likevel er det et stort skritt fremover om NATO-landene utarbeider og tilgjengeliggjør modeller iht ett av de tre rammeverkene NAF, MODAF eller DoDAF. I dag er det relativt få av våre samarbeidspartnere som forholder seg til rammeverkene i særlig grad, eller som modellerer etter en strukturert metode. De som gjør det, bruker modellering som et internt verktøy, og først og fremst på et teknisk nivå. Modellene er dermed ikke synlig på møter og konferanser, og de er ikke lett tilgjengelige for partnernasjoner.

6.1 NAF – NATO Architecture Framework

”NATO C3 System Architecture Framework” [6] er det mest aktuelle rammeverket for kampflyprosjektet. Versjon 3 skulle etter planen ha vært publisert vinteren 2007, men er blitt noe forsinket. NAF v3 er helt nylig kommet FFI i hende. NAF v3 er betydelig påvirket av MODAF, men beholder også noe fra NAF v2. NAF v3 er et mer omfattende og komplett produkt enn NAF v2, og inneholder blant annet en metamodel og en eksempelmodell i UML.

Kampflyprosjektet har hovedsaklig forholdt seg til NAF versjon 2. Arkitekturproduktene som er beskrevet i NAF v2 er delt i fire hovedgrupper. Disse kalles ”views”. I NAF v3 finnes syv ”views”. Disse gruppene består av en rekke enkeltprodukter som i NAF v2 kalles ”viewpoints”, og i NAF v3 ”subviews”. En sammenligning av produktene i NAF v2 og NAF v3 finnes i Tabell 6.1.

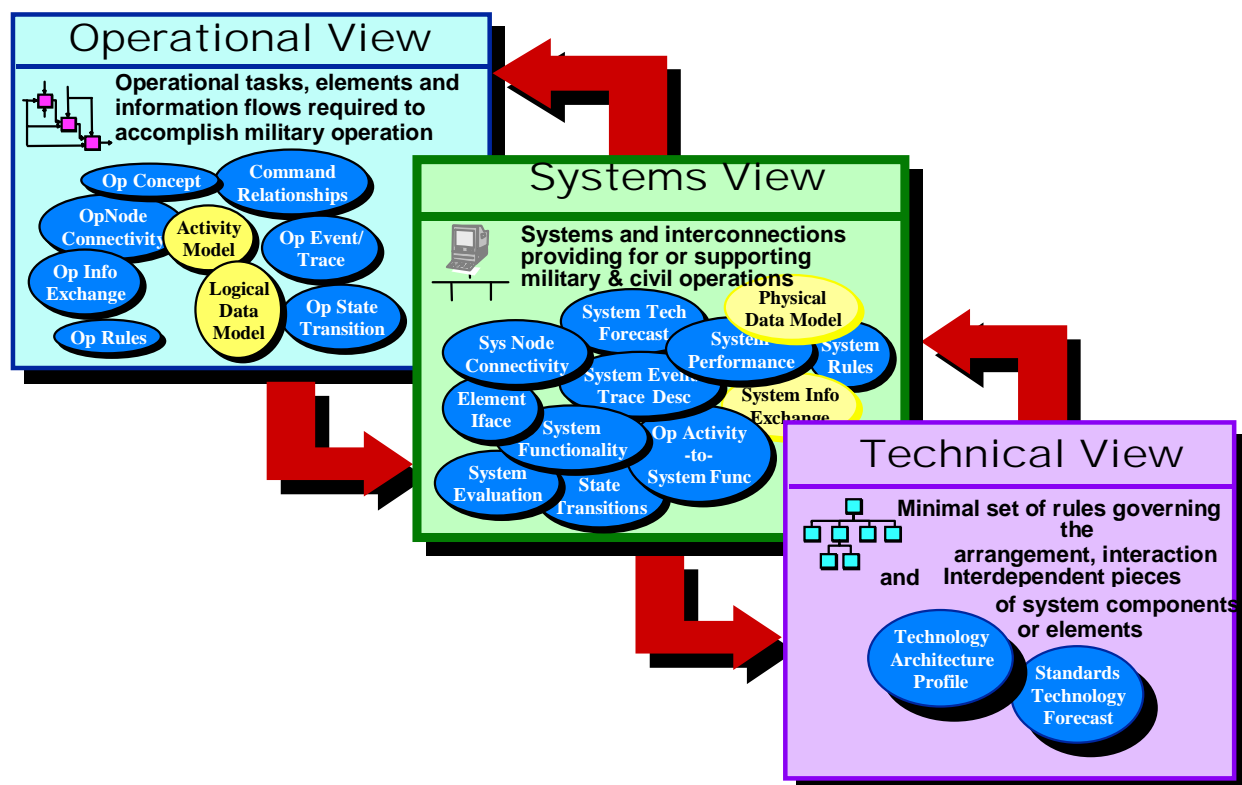
”All View”, ”Operational View”, ”Systems View” og ”Technical View” er gjensidig utfyllende og henger logisk sammen. Til sammen utgjør produktene et helhetlig og strukturert informasjonsprodukt.

”NATO All View” fungerer som en innledning til modellen. Her finner man informasjon av typen ”hvem, hva, hvordan, når og hvorfor” relatert til selve modellen. ”All View” inneholder også et felles bibliotek med begrepsdefinisjoner og gjerne en ”taksonomi” – en beskrivelse av hvordan begreper henger sammen (et strukturert begrepsapparat). ”NATO Operational View” beskriver operative konsepter, operative aktiviteter, organisasjonsforhold, operative elementer (”noder”) samt informasjonsflyt mellom disse. Dette operative perspektivet skal i utgangspunktet være

uavhengig av teknologiske løsninger. "System View" omhandler hvordan konkrete systemer løser oppgavene som er beskrevet i "Operational View". "Technical View" inneholder koblinger mellom systemer og gjeldende og kommende standarder.

De fleste prosjekter må prioritere hvilke NAF produkter de skal produsere og bruke. Hvilke diagrammer etc som skal taes frem avhenger av fokus for det aktuelle prosjektet. Et prosjekt som søker å få oversikt over helheten i en problemstilling bør sannsynligvis bygge opp en komplett modell på overordnet nivå før mer detaljerte modeller bygges på enkelte områder. Hvilke diagrammer en er i stand til å bygge (og til hvilket detaljeringsnivå) avhenger også av tilgang til informasjon – en betydelig utfordring for mange prosjekter.

NAF v3 bærer fremdeles preg av sitt utspring og hovedanvendelse innen IT-bransjen. Flere aspekter som er av interesse for kampflyprosjektet er ikke nevnt i NAF v3. En sterkere harmonisering med anskaffelsesprosessene er ønskelig. Blant annet mangler helt kravhåndtering i NAF, og beskrivelser av systemer er mangelfullt omtalt. Derfor har kampflyprosjektet definert en rekke egne typer diagrammer som skal supplere NAF v3. Det er sannsynlig at andre prosjekter opplever det samme. NAF vil derfor sannsynligvis fortsette å utvikle seg, og en NAF v4 er nok derfor ikke langt unna.



Figur 6.1 De tre gruppene arkitekturprodukter i NAF v2 ("All View" er ikke tatt med her).

NAF v2 Views	NAF v3 Views
NAV – NATO All View	NAV – NATO All View
<i>NAV-1 – Overview and Summary Information</i>	<i>NAV-1 – Overview and Summary Information</i>
<i>NAV-2 – Integrated Dictionary</i>	<i>NAV-2 – Integrated Dictionary</i>
	<i>NAV-3 - Metadata</i>
	NCV – NATO Capability View
	<i>NCV-1 – Capability Vision</i>
	<i>NCV-2 – Capability Taxonomy</i>
	<i>NCV-3 – Capability Phasing</i>
	<i>NCV-4 – Capability Dependencies</i>
	<i>NCV-5 – Capability to Organisational Deployment Mapping</i>
	<i>NCV-6 – Capability to Operational Activities Mapping</i>
	<i>NCV-7 – Capability to Services Mapping</i>
NOV – NATO Operational View	NOV – NATO Operational View
<i>NOV-1 – High Level Operational Concept Diagram</i>	<i>NOV-1 – High Level Operational Concept Description</i>
<i>NOV-2 – Operational Node Connectivity Diagram</i>	<i>NOV-2 – Operational Node Connectivity Description</i>
<i>NOV-3 – Operational Information Exchange Requirements (IER) Matrix</i>	<i>NOV-3 – Operational Information Requirements</i>
<i>NOV-4 – Organisation Relationships Chart</i>	<i>NOV-4 – Organisation Relationship Chart</i>
<i>NOV-5 – Operational Activity Models</i>	<i>NOV-5 – Operational Activity Model</i>
<i>NOV-6 – Concept Data Model</i>	<i>NOV-6 – Operational Activity Sequence and Timing Description</i>
	<i>NOV-7 – Information Model</i>
	NSOV – NATO Service Oriented View
	<i>NSOV-1 – Service Taxonomy</i>
	<i>NSOV-2 – Service Definition</i>
	<i>NSOV-3 – Services to Operational Activities Mapping</i>
	<i>NSOV-4 – Service Orchestration</i>
	<i>NSOV-5 – Service Behavior</i>
NSV – NATO System View	NSV – NATO System View
<i>NSV-1 – Systems Interface Description</i>	<i>NSV-1 – System Interface Description</i>
<i>NSV-2 – Systems Communications Description</i>	<i>NSV-2 – Systems Communications Description</i>
<i>NSV-3 – System-to System Matrix (S² Matrix)</i>	<i>NSV-3 – Systems to Systems Matrix</i>
<i>NSV-4 – Systems Functionality Description</i>	<i>NSV-4 – Systems Functionality Description</i>
<i>NSV-5 – Operational Activity to System Function Traceability Matrix</i>	<i>NSV-5 – System Function to Operational Activity Traceability Matrix</i>

<i>NSV-6 – Systems Information Exchange Matrix</i>	<i>NSV-6 – Systems Data Exchange Matrix</i>
<i>NSV-7 – System Performance Parameters Matrix</i>	<i>NSV -7 – System Quality Requirements Description</i>
<i>NSV-8 – System Evolution Description</i>	<i>NSV-8 – Systems Evolution Description</i>
<i>NSV-9 – System Technology Forecast</i>	<i>NSV-9 – Technology Forecast</i>
<i>NSV-10 – Physical Data Model</i>	<i>NSV-10 – SystemsRules, Sequence & Timing Description</i>
	<i>NSV-11 – System Data Model</i>
	<i>NSV-12 – System Service Provision</i>
NTV-NATO Technical View	NTV-NATO Technical View
<i>NTV-1 – System Standards Profile</i>	<i>NTV-1 – Technical Standards Profile</i>
<i>NTV-2 – Standards Technology Forecast</i>	<i>NTV-2 – Technical Standards Forecast</i>
<i>NTV-3 – Technical Configuration</i>	<i>NTV-3 – Standard Configurations</i>
<i>NTV-4 – Software Configuration</i>	
<i>NTV-5 – Product Selection Report</i>	
	NPV – NATO Program View
	<i>NPV-1 Program Portfolio Relationships</i>
	<i>NPV-2 Programme to Capability Mapping</i>

Tabell 6.1 En sammenligning av arkitekturproduktene i NATO Architecture Framework v2 og v3.

7 Modelleringspråk

Grafikk er som oftest en mer kompakt måte å formidle informasjon enn tekst. Bruk av definerte typer symboler og diagrammer gir presisjon i kommunikasjonen. Disse to faktorene kombinert med valg av et standardisert språk legger forholdene til rette for effektiv kommunikasjon, der modeller kan utveksles og umiddelbart leses og kobles med andre modeller.

Integrering av modellbidrag fra mange ulike prosjektmiljøer lettes betydelig av at det stilles visse krav til hvordan systemer beskrives. Eksempelvis bør en modell av et system som ACCS (Air Command and Control System) kunne gjenbrukes i andre modeller av systemer og operasjoner. En modell av ACCS er aktuell for kampflyprosjektet på FFI, såvel som for en rekke andre prosjekter som arbeider med luftfartøy og luftovervåkning. Det er ikke hensiktsmessig om disse prosjektene må utvikle egne beskrivelser av ACCS. På samme måte er det hensiktsmessig om en og samme kampflymodell kan brukes av alle prosjekter som arbeider med systemer og konsepter som involverer kampfly.

Standarder innenfor modelleringspråk er relativt ”ferske”. Det finnes flere ulike modelleringspråk som kan brukes av kampflyprosjektet. UML 2.0 (Unified Modeling Language version 2.0) er godt etablert i industrien. Noen andre derivater av UML – UML RT, xtUML og SysML – er også interessante å se nærmere på da de delvis forenkler bort (for oss) overflødige egenskaper, og legger til andre nyttige muligheter. Modelleringspråket UML RT er spesielt

interessant i forbindelse med en kobling mellom beskrivende modelleringsvirksomhet og simulering.

Arkitekturrammeverkene legger ikke føringer på hvilket modelleringsspråk som skal benyttes. Det er også flere arkitekturprodukter i NAF som ikke har en direkte ekvivalent i eksempelvis UML. I mange tilfeller er det mest hensiktsmessig å lage enkelte arkitekturprodukter med Excel, Word eller et tegneprogram som Visio. Et eksempel er NOV-3 (Operational Information Exchange Requirements Matrix), som egner seg godt som en tabell i Word eller Excel.

7.1 UML – Unified Modeling Language

UML 2.0 ble utgitt av OMG – Object Management Group – i 2005. Det er et grafisk modelleringsspråk som i utgangspunktet var laget for å støtte utvikling av programvaresystemer. UML-spesifikasjonen inneholder definerte elementer, relasjoner og diagrammer. Elementene og relasjonene representeres med unike symboler.

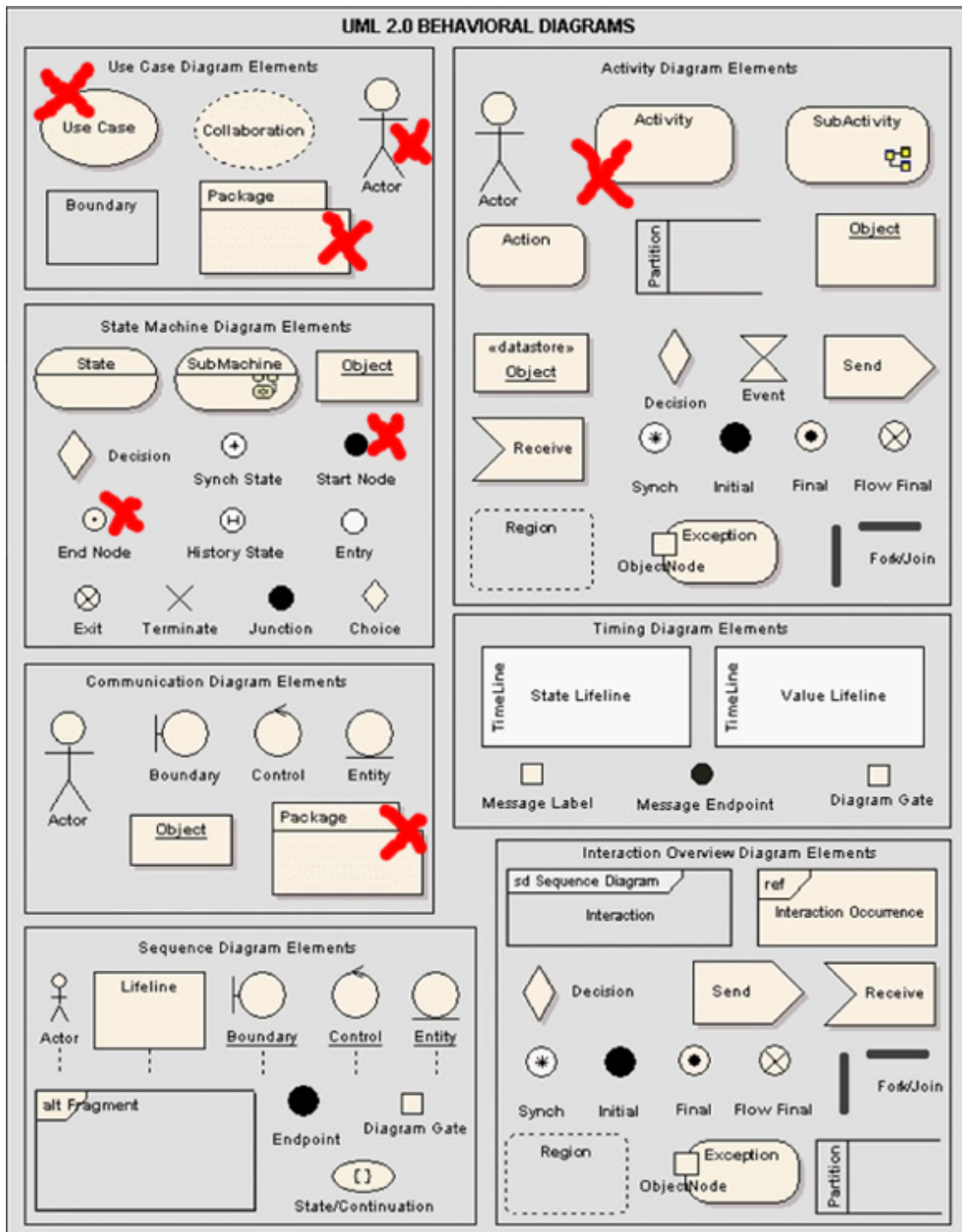
Man kan forstå mange UML-diagrammer uten å ha forkunnskaper om UML. Imidlertid er mange aspekter ved UML ikke direkte intuitive. En UML-modell bør derfor ikke alltid brukes som kommunikasjonsmedium når mottakere ikke har kunnskaper om dette språket.

Opplæringstid avhenger av brukerbehov og bakgrunn. De fleste lærer nok til å lese UML-modeller i løpet av noen timer eller dager. Det er heldigvis kun et fåtall av de mange mulighetene i UML som brukes ofte, og mange tilgjengelige verktøy har gode hjelp-funksjoner. Det er betydelig lettere å lære nok om UML til å forstå andres modeller enn å bygge egne gode modeller.

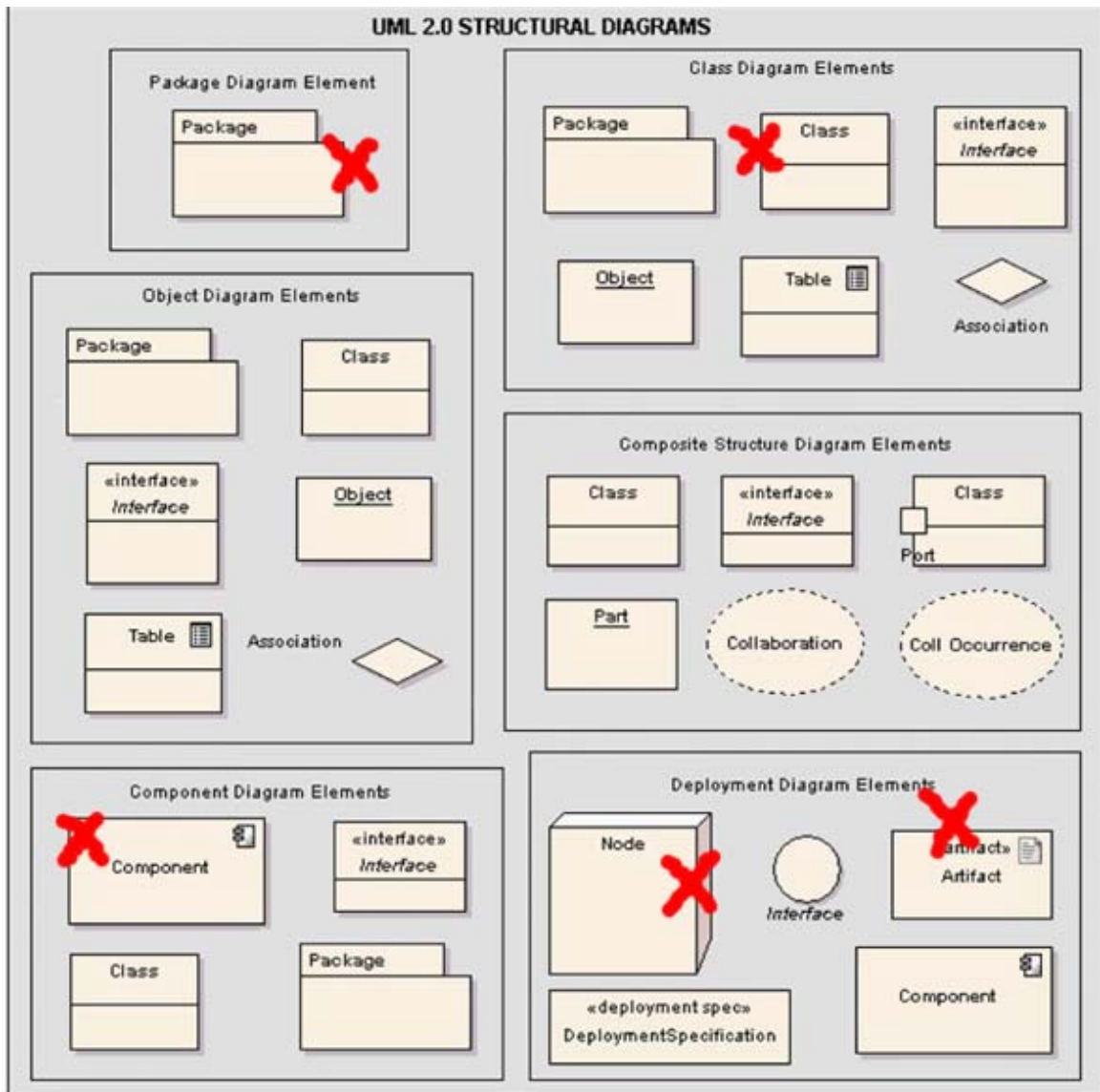
7.1.1 UML symboler og diagrammer

UML inneholder en lang rekke elementer som hver har en definert betydning. Ulike sammenhenger mellom disse elementene kan modelleres med definerte typer piler og streker. Elementene og relasjonene merkes med unike navn, og kan inneholde store mengder informasjon. Det er vanligvis ønskelig å skjule det meste av informasjonen som ligger i et element eller relasjon for å gjøre diagrammene lesbare. Egnede modelleringsverktøy og leseprogrammer gjør det mulig å vise eller skjule informasjon etter ønske.

UML definerer 13 ulike diagramtyper, fordelt på to grupper – diagrammer som viser oppførsel og diagrammer som viser struktur. Ved å bruke ”stereotyper” er det mulig å definere egne varianter av UML-elementene. Det er også fullt mulig å definere egne typer diagrammer etter ønske, selv om det finnes visse begrensninger. Et UML modelleringsverktøy vil stoppe enkelte forsøk på sammenstilling av elementer i et diagram.



Figur 7.1 UML-elementene som inngår i beskrivelser av systemoppførsel. Noen elementer går igjen flere steder. Elementer som er brukt til nå i kampflymodellen er merket med et rødt kryss. (Hentet fra den innebygde hjelp-funksjonen i Enterprise Architect)



Figur 7.2 UML-elementene som inngår i beskrivelser av systemstruktur. Elementer som er brukt til nå i kampflymodellen er merket med et rødt kryss. (Hentet fra den innebygde hjelp-funksjonen i Enterprise Architect)

Behavioral Diagram Connectors

Activity Diagrams	
↗ Control Flow	X
↗ Dependency	X
↗ Interrupt Flow	
↗ Object Flow	

Use Case Diagrams	
/ Associate	X
↗ Dependency	X
↗ Extend	X
↗ Generalize	X
↗ Include	X
↗ Realize	X
↗ Trace	X
↗ Use	X

State Machine Diagrams	
↗ Transition	
↗ Object Flow	

Timing Diagrams	
↑ Message	

Sequence Diagrams	
→ Message	
↻ Self-Message	
↻ Recursion	
↻ Call	

Communication Diagrams	
/ Associate	
↗ Dependency	
↑ Message	
⊗ Nesting	
↗ Realize	

Interaction Overview Diagrams	
↗ Control Flow	X
↗ Dependency	
↗ Interrupt Flow	
↗ Object Flow	

Structural Diagram Connectors

Composite Structure Diagrams	
↖ Assembly	
/ Connector	
↗ Dependency	
⊗ Expose Interface	
↗ Occurrence	
↗ Represents	
↗ Role Binding	
↗ Trace	

Package, Class and Object Diagrams	
⊗ Aggregate	
↖ Assembly	
/ Associate	
◇ Association	
⊗ Association Class	
↗ Compose	
↗ Dependency	
⊗ Expose Interface	
↗ Generalize	
⊗ Nesting	
↗ Pkg Import	
↗ Pkg Merge	
↗ Realize	
↗ Trace	
↗ Use	

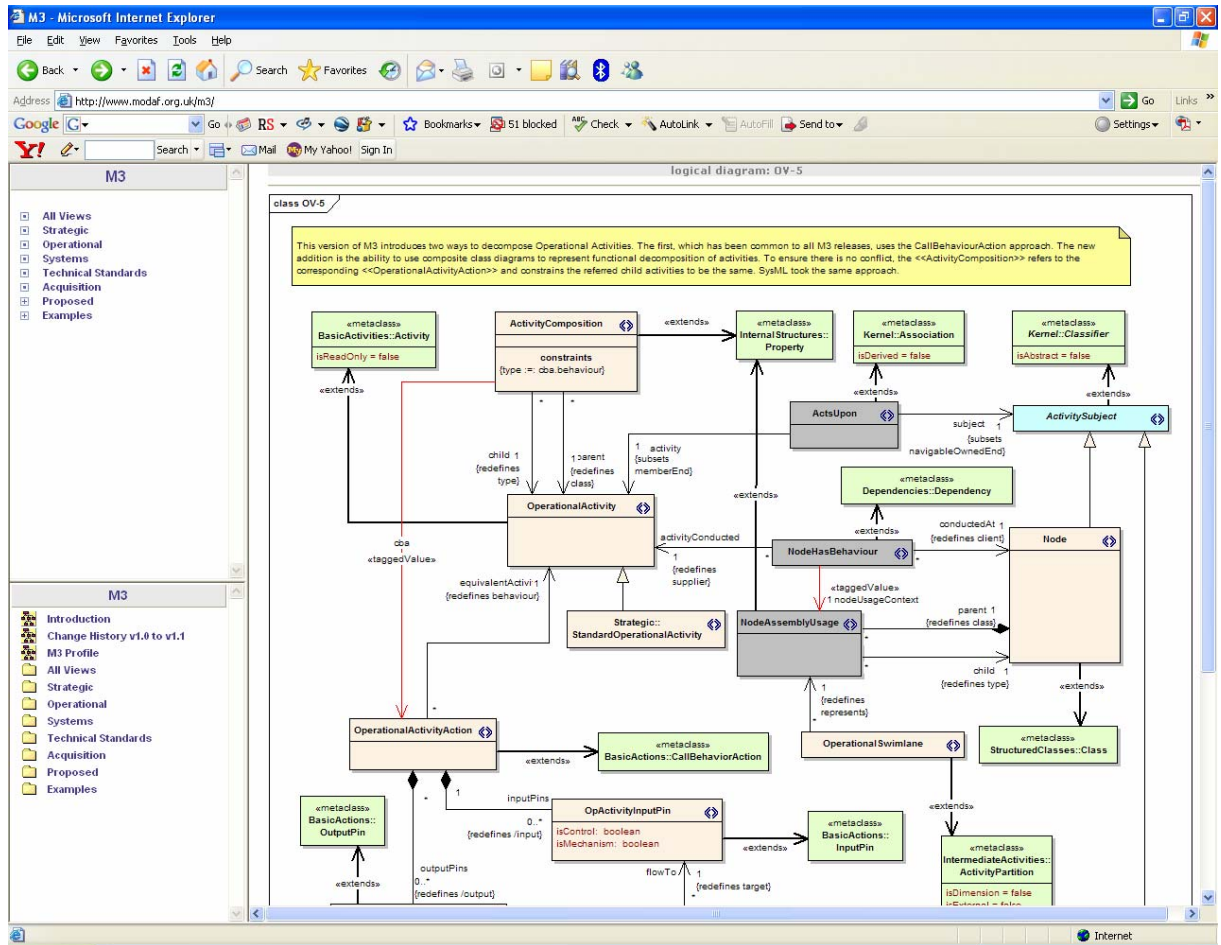
Component Diagrams	
↖ Assembly	
/ Associate	
↗ Delegate	
↗ Dependency	
⊗ Expose Interface	
↗ Generalize	
↗ Manifest	
↗ Realize	
↗ Trace	
↗ Use	

Deployment Diagrams	
/ Associate	
⊗ Association Class	
↗ Dependency	
↗ Deployment	
X ↗ Generalize	
↗ Manifest	
⊗ Nesting	
↗ Object Flow	
↗ Realize	
↗ Trace	

Figur 7.3 UML relasjonene ("connections") gruppert etter diagramtype. Flere av relasjonstypene vises her i flere diagramtyper. Relasjoner som er brukt til nå i kampflymodellen er merket med et rødt kryss. (Hentet fra den innebygde hjelp-funksjonen i Enterprise Architect)

8 Metamodeller og profiler

Hvordan informasjon skal organiseres og representeres er spesifisert i metamodeller og profiler. Det kan være vanskelig å få en klar forståelse av begrepene ”metamodell” og ”profil”. Det hjelper lite at de to begrepene brukes noe om hverandre og at metamodeller og profiler ofte slås sammen. Det vil si at både informasjonsstrukturen og eksempelvis hvilke UML-elementer som skal brukes er spesifisert i samme modell. ”MODAF Meta Model” (M3) er et eksempel på dette. Her er stereotypene i et MODAF-produkt (altså de ulike typene informasjon), og ”metaklassene” (symboltypene) de representeres med grafisk, fremstilt i samme diagram.



Model) gjør, men beskriver i rundere ordelag hvilken type informasjon de ulike produktene skal inneholde. NAF v3 inneholder en metamodell som har mye til felles med M3. NMM (NATO Meta Model) i NAF v3 fjerner en hel del av den usikkerheten (les: det frie spillerommet) som NAF v2 tillot.

Metamodeller kan også kalles ”referansemodeller” eller ”informasjonsmodeller”. Deres viktigste funksjon er å sikre kompatibilitet i modeller som utvikles av ulike aktører. De spesifiserer også hvordan modeller skal lagres og utveksles. Et vanlig lagringsformat er XML, og korresponderende utvekslingsmedium er XMI.

En type informasjon beskrives gjerne i metamodellene som ”stereotyper” med ulike egenskaper forbundet med disse. En metamodell bestemmer ikke nødvendigvis hvilke UML-symboler som skal brukes for å representere en stereotyp eller hvordan disse skal se ut.

8.2 Profiler

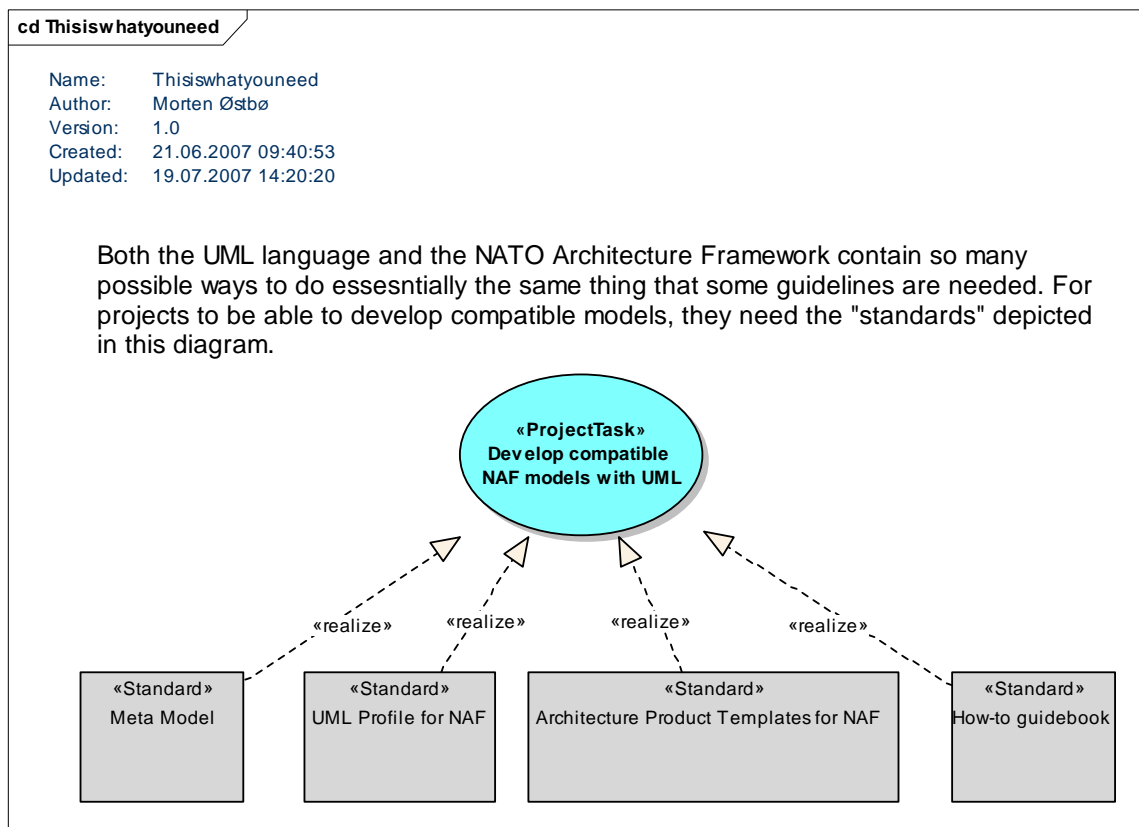
Arkitekturrammeverkene overlater mange valg til verktøyutviklere og modellerere (”arkitekter”) gjennom ikke å spesifisere notasjon. Utfordringen med bruk av UML er ikke å finne måter å representere det en ønsker. Utfordringen er å bli enig med andre om hvordan dette skal gjøres. Dersom man ikke klarer å bli enige, resulterer dette i redusert effektivitet i informasjonsutveksling. Uten maler og kortfattede veiledninger, vil sannsynligvis ikke modellering kunne vinne innpass i et stort antall prosjekter. Modeller blir vanskelige for andre enn opphavspersonen å forstå, og det blir ikke mulig å importere en delmodell inn i en annen modell uten mye tilpasning.

Ved hjelp av stereotyper, ”attributes” osv forteller en profil hvordan eksempelvis UML brukes i en modell. En profil kan fortelle at en operativ node skal representeres i modellen som en UML 2.0 ”Node” med stereotypen «OperationalNode». Profilen kan også spesifisere en rekke egenskaper for en slik node. Fargen på en type element kan velges når man oppretter profilen, slik at f eks en operativ node alltid blir grønn. Slik ivaretar man en bedre ”gjenkjennelsesfaktor” og bedre oversikt i modellene enn om alle elementene er UML default farge (beige).

En profil for NAFv2/v3 vil fjerne en hel del av den usikkerheten som nå eksisterer rundt bruken av UML. En profil (sammen med en metamodell) vil gjøre det mulig koble en kampflymodell til andre modeller på en lettvinns måte. En profil vil inneholde kun de UML-elementene som er aktuelle. Den begrenser hvilken funksjonalitet i språket som skal brukes til å realisere metamodellen. Profilen kan legges inn i modelleringsverktøyet. Man kan dermed plukke elementer og relasjoner fra denne når man skal bygge diagrammer. Slik sikres kompatibel bruk av symboler, farger osv, og man forenkler modelleringsjobben betraktelig.

En UML-profil er likevel ikke en ”modell-mal”. Den hjelper ikke brukeren til å lage NAF-produkter utover å spesifisere hvordan elementer i modellen representeres og hvilke relasjoner som kan brukes. En standard modell-mal er nødvendig i tillegg til profilen.

En profil er tatt frem for kampflymodellen ved hjelp av Enterprise Architect. Kampflyprofilen har hentet en del inspirasjon fra UPDM (UML Profile for DODAF and MODAF) [9] og M3 (MODAF Meta Model). UPDM skal utgis i 2008. Første versjon ble foreløpig lagt frem tidlig i 2007. Arbeidet på FFI har ikke kunnet nyte godt av en felles valgt standard i Norge, men i stedet måttet konstatere gjennom førstehånds erfaring at behovet er stort for en UML-profil for NAF.



Figur 8.2 Det ligger mange muligheter i UML og NAF. En gitt problemstilling kan modelleres på en lang rekke ulike, men ekvivalente, måter. Kompatibilitet i modeller krever derfor noen standarder.

9 Modelleringsverktøy

UML-diagrammer og andre arkitekturprodukter kan lages med vanlige programmer som PowerPoint, Excel, Visio og Word. Mange nye muligheter åpner seg imidlertid ved å benytte programverktøy som er utviklet spesielt for arbeid med systemarkitektur, og som er bygget opp rundt en databaseløsning. Særlig når antallet elementer og relasjoner blir stort, er det viktig å bruke et egnet databaseverktøy. Det blir svært vanskelig å holde rede på sammenhenger, og sikre at endringer blir gjort alle steder dette er nødvendig uten støtte av et dataverktøy som er utviklet for dette.

IFEAD [10] – Institute For Enterprise Architecture Developments – i Nederland har laget en veiledning i valg av verktøy, med en oversikt over eksisterende kommersielle produkter. FFI har

fokusert på "Enterprise Architect" fra Sparx Systems og "System Architect" fra TeleLogic. Både Sparx Systems og TeleLogic er medlem av "UPDM" team, sammen med en rekke store aktører i forsvarsindustrien, deriblant Lockheed Martin, Raytheon, BAE Systems, General Dynamics, Northrop Grumman og Thales. Valget til kampflyprosjektet har foreløpig falt på Enterprise Architect, da dette verktøyet brukes av FLO/IKT og FK KKIS. Valget var ikke et resultat av en systematisk vurdering av behov og funksjonalitet. Kampflyprosjektet har vurdert System Architect fra TeleLogic ved å installere en prøveversjon, samt ved å invitere TeleLogic til en demo her på FFI. System Architect virker godt egnet, og har et GUI som er tilpasset modellering i henhold til flere aktuelle arkitekturrammeverk. Støtte for DoDAF er veletablert, mens støtte for NAF og MODAF er under utvikling. System Architect er ikke i utgangspunktet UML-orientert, men støtter bruk av UML som ett av mange alternativer.

9.1 Enterprise Architect

"Enterprise Architect" (EA) er et databasebasert modelleringsverktøy for UML [11]. Det ikke et verktøy som er tilrettelagt spesielt for å modellere iht NAF, MODAF eller DoDAF. EA er heller ikke et prosjektstyringsverktøy, men man kan fint bruke det til å beskrive prosjekter, deres aktiviteter, aktører, fremdrift, status og sammenhenger til andre prosjekter.

EA definerer 4 diagramtyper i tillegg til de 13 som er standard for UML 2.0. Når man oppretter et nytt EA-prosjekt, etableres det automatisk en grunnstruktur for modellen, organisert etter "View". Disse tilsvarer "View" i NAF, altså en gruppering av diagrammer som skal belyse systemet fra en gitt synsvinkel. De fleste vil velge å legge opp en egen modellstruktur, noe kampflyprosjektet også har gjort.

Det er en viss brukerterskel forbundet med å bruke EA. Denne terskelens overkommelighet vurderes ulikt hos ulike brukere. Kampflyprosjektet har vurdert verktøyet til å være brukervennlig, og meget godt egnet til å modellere i henhold til NAF. Mangelen på et ferdig tilrettelagt grensesnitt med maler for NAF arkitekturprodukter oppleves som en begrensning i starten, men savnes ikke senere.

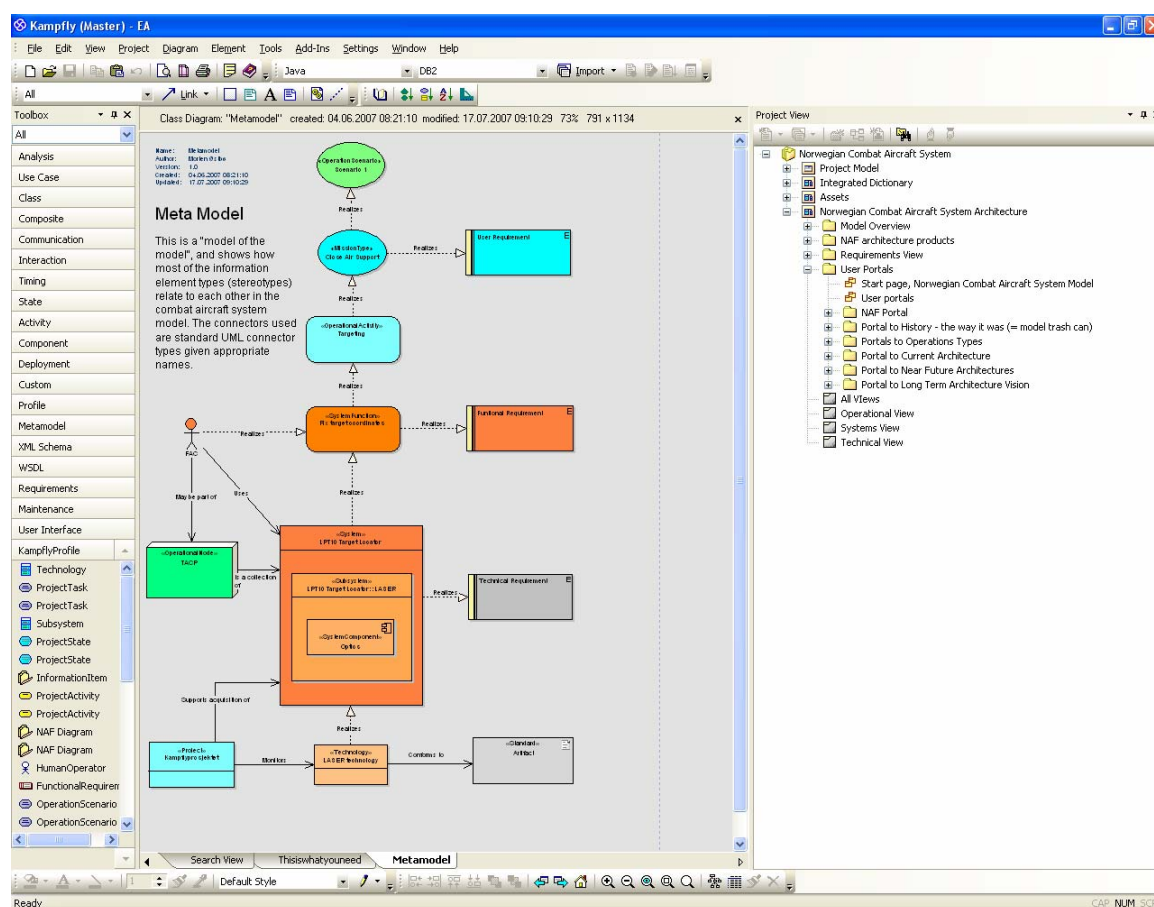
EA inneholder en lang rekke funksjoner som er meget nyttige når man skal utvikle egne modeller såvel som når man skal navigere i andres modeller. EA har en svært omfattende hjelp-funksjon innebygget. Det er også mulig å kjøpe tilleggsprogramvare for ytterligere hjelpfunksjonalitet.

Dersom det ikke er mulig eller hensiktsmessig å inkludere all informasjon man kunne ønske tilgang til i én modell, kan man med EA legge lenker til andre dokumenter eller modeller i den gjeldende modellen. Eksempelvis kan det være praktisk å kunne ta frem PowerPoint presentasjoner, Word-filer, pdf-filer mm fra modellen. Disse filene kan inneholde figurer, bilder og tekst som utfyller eller utdyper den info som finnes i UML-modellen. Man kan også linke til web-sider fra EA. Gjennom denne muligheten til å knytte sammen mange ulike kilder til informasjon, kan en EA-modell brukes som en komplett informasjonsportal.

EA kan automatisk generere en rekke ulike typer dokumentasjon fra modellen. Dersom det er

mest praktisk, kan hele eller deler av en modell dokumenteres i RTF tekstdokumenter. HTML kan også genereres. Dette gjør det mulig å tilgjengeliggjøre modellen på et internt nett eller Internett. Dette er eksempelvis gjort for INI-modellen som er utviklet av FLO/IKT med flere. Gjennom INI-portalene på FIS Basis kan brukere navigere i modellen med en vanlig web-browser.

Samarbeid mellom flere brukere av EA er mulig på flere måter. Kampflyprosjektet har benyttet seg av en "Master" og flere "Replica" som samkjøres ved hjelp av "Synchronize Replicas". Modeller og pakker kan importeres og eksporteres over XMI, og flere brukere kan arbeide mot samme database dersom denne legges på en felles katalog. "EA Light" er gratis tilgjengelig på Internett, og muliggjør lesing av EA-modeller.

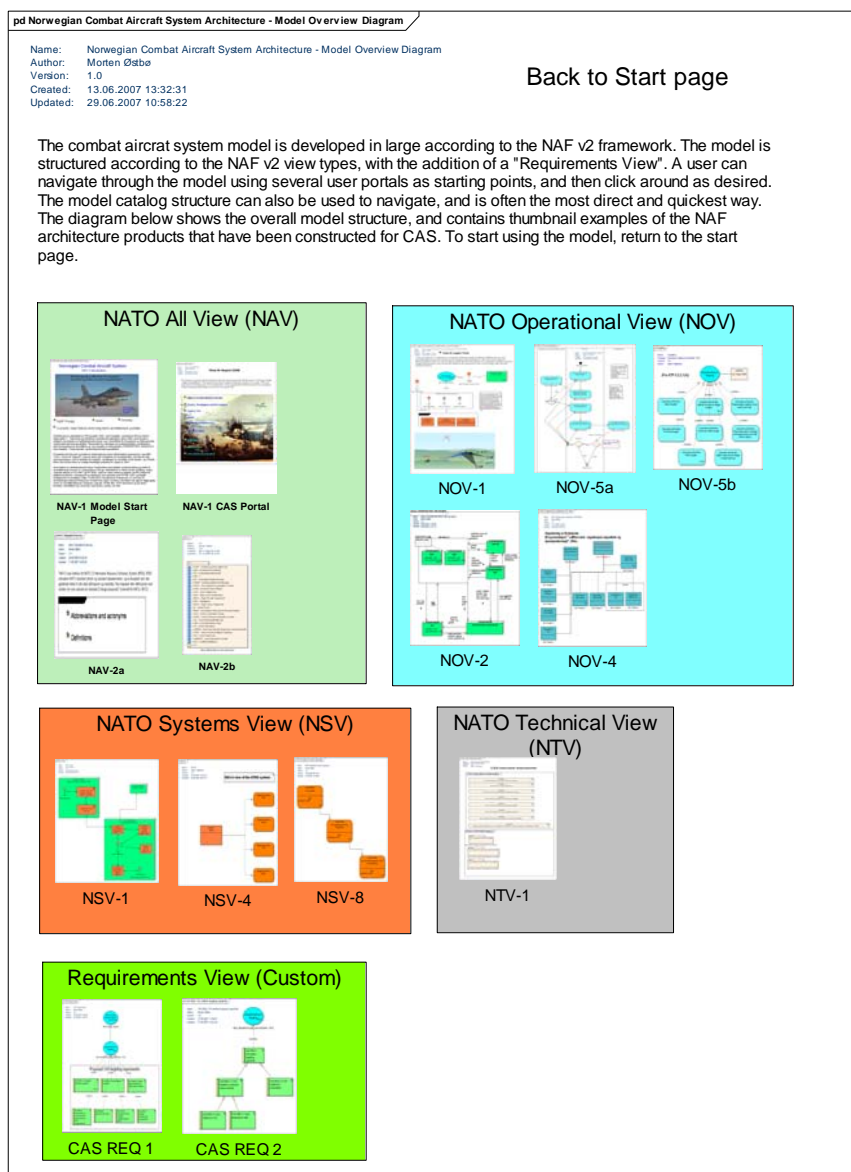


Figur 9.1 Et skjermbilde fra Enterprise Architect (EA). Dette bildet viser metamodellen for kampflymodellen. Ved bygging av diagrammer, plukkes elementer fra verktøykassen til venstre. Til høyre vises modellstrukturen. Navigering i modellen er mulig både herfra og direkte i diagrammene.

10 Kampflymodellen

Parallelt med kompetanseoppbygging på modellering, systemarkitektur og nærstøtteoperasjoner med kampfly, har prosjektet etablert en UML-modell. Modellen som nå foreligger danner et grunnlag for videre bruk av UML-modellering i støtte til anskaffelse av nye kampfly. Den legger

også grunnlaget for at kampflymiljøet kan bidra til den mer helhetlige prosessen rundt transformasjon og interoperabilitet i Forsvaret. I det videre omtales ambisjonsnivået for modellen, og resonnerer rundt struktur og innhold.



Figur 10.1 En oversikt over kampflymodellen. Den inneholder et "Requirements View" i tillegg til de fire "View" som er standard i NAF v2. Oversikten inneholder et lite utvalg av diagrammene i modellen.

10.1 Ambisjonsnivå

Modellering av kampflysystemet og kampflyoperasjoner kan bli en uforholdsmessig ressurskrevende aktivitet for et enkeltprosjekt dersom ikke det formuleres klare mål og prioriteringer. I utgangspunktet har det vært ønskelig for prosjektet å utforske alle mulighetene som modellering med UML gir til å støtte anskaffelsen av nye kampfly. Dette innebærer å ta frem alle de arkitekturproduktene som beskrives i NAF, samt en rekke andre produkter som ikke er

beskrevet der. Dersom dette ble gjennomført, ville kampflymodellen inneholde informasjon som går langt utover det kampflyprosjektet isolert sett har behov for på kort sikt. Slik informasjon ville for eksempel inkludere en forholdsvis detaljert beskrivelse av dagens kampflysystem, inkludert de deler av sentrale beslutningsprosesser som angår kampflyoperasjoner. Prosjektets ressurser har forhindret en slik ambisiøs satsing.

Behovene til kampflyprosjektet på kort sikt bestemmer hvilke arkitekturprodukter som er aktuelle nå i første omgang, samt hvilket domene som skal modelleres. Kampflyprosjektet har valgt å fokusere på operative oppgaver og systemfunksjoner forbundet med luft-til-bakke-operasjoner. Disse modelleres i en lang rekke diagrammer av typen NOV-5 og NSV-4.

Med utgangspunkt i overordnede modeller av dagens CAS-konsept, har prosjektet formulert forslag til brukerkrav til fremtidige evner til nærstøtteoperasjoner. "Brukerne" i denne sammenhengen er først og fremst Forsvaret, ikke eksempelvis flygere eller FAC. Dette for å sikre at Forsvarets behov står i sentrum, og for å åpne for at fremtidige konsepter ikke nødvendigvis er kampfly- eller FAC-sentriske. Det har vært ønskelig med en svak binding til løsningsform i dette innledende arbeidet med CAS, selv om det naturligvis har måttet bli mye fokus på FAC, som er sentral i dagens konsept.

Modellen vil videreutvikles iterativt gjennom brukererfaringer og debatt. Det vil være viktig å ta modellen i bruk som et arbeidsverktøy, slik at modellen blir godt tilpasset brukernes behov. Modelleringen er ikke et mål i seg selv, og bør ikke eksistere som en sideaktivitet, men snarere som en naturlig delaktivitet i det daglige. Modellen bør bli en levende ressurs, ikke "shelf ware". Dette innebærer eksempelvis at UML-diagrammer brukes til å støtte simuleringer i FFI Battle Lab. I denne sammenhengen kan modellen presentere informasjon på en ryddig måte, samt gi rask tilgang til beskrivelser av systemer og operasjoner til støtte for diskusjoner som oppstår under bruk av Battle Lab

10.2 Valg av rammeverk, modelleringsspråk og programverktøy

Andre tilsvarende aktiviteter på FLO/IKT og FK/KKIS samt systemarkitekturprosjektet på FFI har vært førende for kampflyprosjektets valg av verktøy og rammeverk. Valget har foreløpig falt på følgende:

- ❑ Rammeverk: NAF v2
- ❑ Modelleringsspråk: UML 2.0
- ❑ Modelleringverktøy: Enterprise Architect fra Sparx Systems
- ❑ Tekstspråk i modellen: Engelsk

Kampflyprosjektets arbeid har inntil nylig måttet forholde seg til NAF v2, da NAF v3 ikke kom til forespeilet tid. Noen tillegg og endringer er gjort basert på det nye rammeverket, men full overensstemmelse med NAF v3 har ikke vært mulig.

Beslutningen om å modellere på engelsk har sammenheng med ønsket om å kunne utveksle

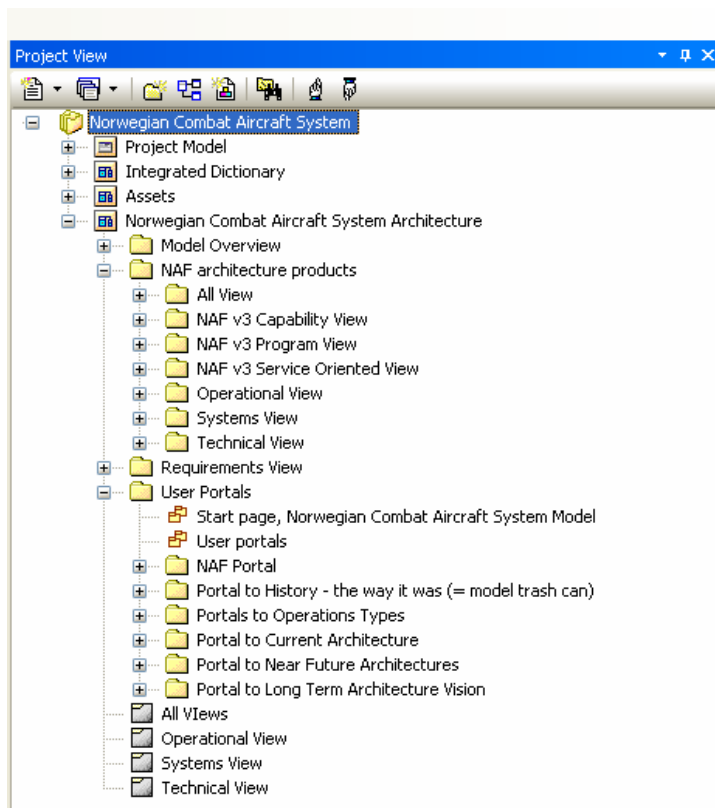
modeller med samarbeidspartnere i utlandet. Denne beslutningen kom imidlertid noe ut i prosjektløpet. Derfor finnes det fremdeles en hel del bruk av norsk språk i modellen.

10.3 Modellstrukturen

Begrepet ”Modellstruktur” refererer her til katalogstrukturen i Enterprise Architect der UML-elementene og diagrammene ligger.

Modeller kan organiseres på ulike måter med hensyn til synsvinkel, tema eller tidsperspektiv. Organisering påvirker brukervennligheten av modellene, og hvor lett det blir å sammenstille eller integrere ulike modeller. Modellstrukturen er synlig for brukere også når man genererer html-sider med Enterprise Architect. Man kan altså ikke skjule katalogstrukturen for en bruker dersom man benytter denne metoden for tilgjengeliggjøring (eksempelvis på FISBasis). En bruker kan altså velge å navigere via både modelltreet og diagrammene. Strukturen bør derfor være logisk og intuitiv både for modellerere og for andre brukere som er mindre kjent med UML eller det aktuelle modelleringsverktøyet.

Arkitekturrammeverkene spesifiserer ikke hvordan modeller skal organiseres. Samtidig vil de fleste modeller inneholde mer enn de produktene som eksempelvis NAF beskriver. Et eksempel er ”Assets” – en samlekatalog for modellens ”byggeklosser” og andre ressurser. Det finnes ennå noen etablert måte å gjøre dette på. Kampflyprosjektet har eksperimentert med flere ulike modellstrukturer underveis i arbeidet.



Figur 10.2 Strukturen til kampflymodellen.

Flere arkitekturmodeller i Norge har en inndeling etter tidsaspektet, med en "Nåsituasjonsarkitektur", en eller flere alternative "Målarkitekturer" og en eller flere alternative "Referansearkitekturer". Disse tre delmodellene inneholder gjerne stort sett de samme typene arkitekturprodukter og er inndelt på samme måte, men har ulike tidsperspektiv. Tidsperspektivet som man har forbundet med referansearkitekturen varierer fra 5-6 år frem i tid til omlag 20 år frem i tid. Eksempelvis snakker man i Norge (FLO/IKT og FFI) om "NbF grad 2" realisert i 2012 som en referansearkitektur. Målarkitekturerne kan befinne seg hvor som helst mellom nåsituasjonen og referansearkitekturen.

Kampflyprosjektet har foreløpig valgt å organisere modellen i stor grad etter NAF-strukturen, med ytterligere kataloger for innhold som ikke passer inn i NAF. Ettersom det ikke finnes en modellorganisering som er optimal for enhver brukers ønsker, er det åpnet for en mer fleksibel navigering gjennom å bruke "portaler" og lenker fra hvert diagram til de mest aktuelle relaterte diagrammene. Tidsaspektet er valgt nedtonet, med diagrammer som fremstiller arkitekturen til ulik tid osv lokalisert i de samme pakkene som de øvrige diagrammene for et gitt tema. Vi har altså valgt å ikke dele modellen i "nå"-, "snart"- og "fremtidsperspektiv".

10.4 To UML-tekniske problemstillinger

En gitt problemstilling kan modelleres på mange ulike, men likeverdige måter. Dette er tilfelle også der man har en metamodell og UML-profil å forholde seg til, selv om friheten da blir betydelig redusert. Problemstillingen med bruk av UML illustreres med et par konkrete eksempler i de følgende underkapitlene.

10.4.1 Modellering av mennesker i kampflysystemet

INCOSE "Systems Engineering Handbook" har følgende definisjon av et system:

"An integrated set of elements that accomplish a defined objective . These elements include products (hardware, software, firmware), processes, people, information, techniques, facilities, services, and other support elements"

Menneskelige operatører og organisasjoner kan representeres med UML på flere måter. I UML er vanligvis en "actor" en person som står utenfor og bruker systemet. I dette perspektivet er eksempelvis kampflygeren en bruker av systemet "kampflyet". Mennesker i Forsvaret er imidlertid både brukere av Forsvaret som system og komponenter i systemet. Flygeren er en komponent i systemet for en person eller organisasjon som står utenfor Forsvaret. Disse eksterne menneskene er da "brukere" av Forsvaret.

Mange UML-modeller inneholder diagrammer der menneskelig interaksjon med et system fremstilles i "Use Case Diagram". Menneskene er gjerne umiskjennelige "pinnemennesker", og "use cases" er ellipser med tekst. Denne metoden har blitt brukt mye i kampflymodellen inntil nylig. Enkelte andre UML-modeller kan fremstille den samme problemstillingen helt uten å bruke

hverken "use case" eller "actor". Eksempelvis er ikke disse to symbolene brukt i den foreløpige NAF v3, selv om de forekom hyppig i UML-eksempler i NAF v2. Dette illustrerer en dreining av NAF i retning større vektlegging av operative evner, tjenester og funksjoner. Dette har igjen sammenheng med at NAF er utviklet hovedsaklig av representanter for "INI-miljøet" gjennom NC3A.

Bruk av "actor"-symbolet gir et tydelig skille mellom mennesker og "maskiner" som kan være hensiktsmessig. Et menneske har såpass unike egenskaper at det er en fordel å bruke et symbol som er umiddelbart gjenkjennbart, og som gir en mer intuitiv kommunikasjon med brukere av modellen enn å bruke eksempelvis "component". Erkjennelsen av at menneskelige operatører ikke handler etter regler alene gjør at de må ha en spesiell behandling i beskrivende modeller og i simuleringmodeller. Det er likevel mulig å fremstille mennesker på samme nivå i modellen som en hvilken som helst annen systemkomponent som realiserer en operativ evne. Kampflyprosjektet begynte med en relativt "menneskesentrisk" modell. Mange diagrammer var bygget opp rundt en FAC. Modellen dreies nå i retning fokus på operative evner, oppgaver og tjenester. Dette vil gjøre det lettere å analysere alternative løsninger. Det meste av modellen vil beskrive løsnings-uavhengige aspekter ved luftmaktens virkeområde. Eksempelvis vil modellen fokusere på oppgaver som må løses ifm våpenlevering fra luften mot mål på bakken i nærheten av egne styrker. Dette vil legge grunnlaget for en analyse av ulike løsningsalternativer i neste omgang.

10.4.2 Representasjon av kampflyet i UML

Kampflyet er den luftbårne delen av et system som består av mange delsystemer.

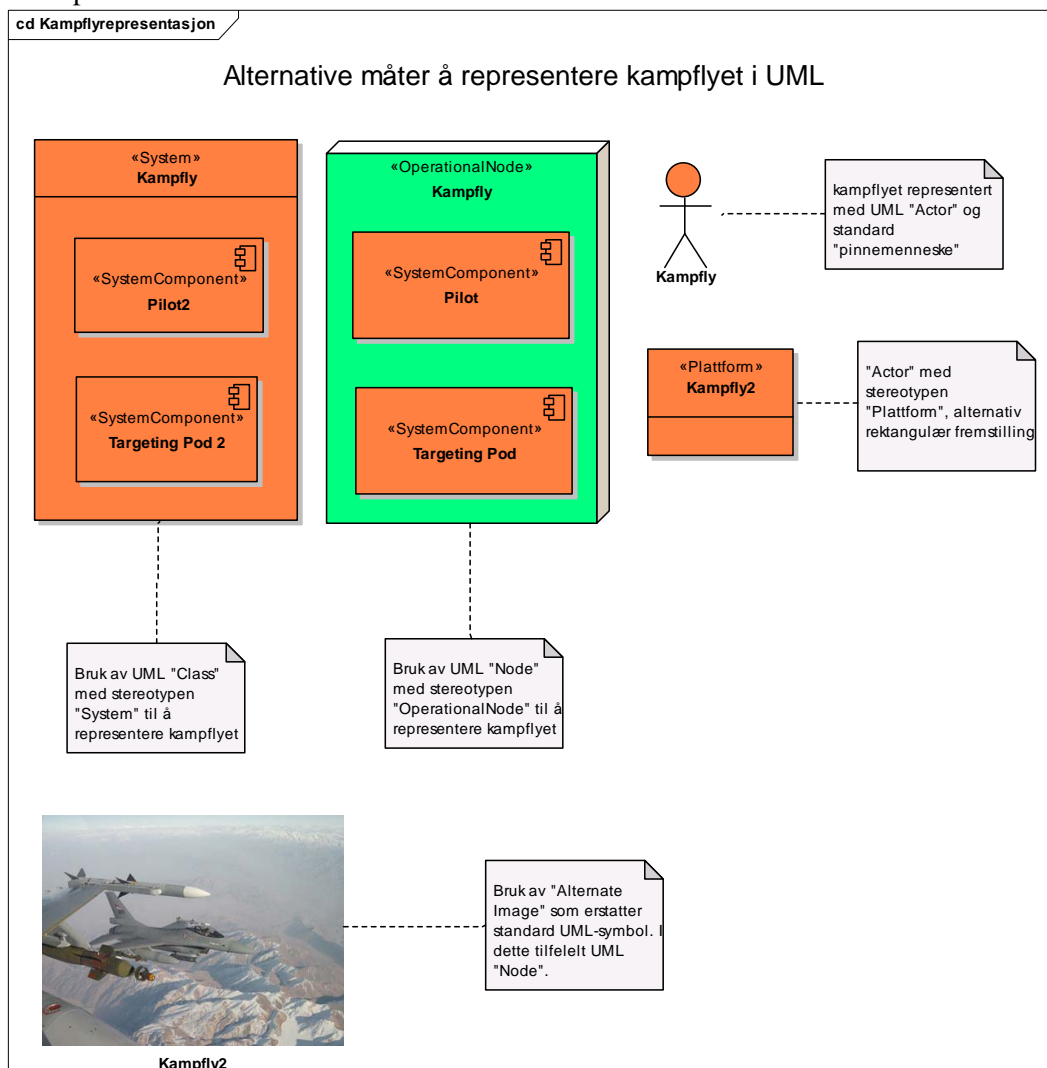
Kampflysystemet er igjen en integrert del av et større system – Forsvaret. Forsvaret er igjen del av et større system, osv. Er det mulig å representere kampflyet, eller kampflysystemet, på en enhetlig måte? Vil representasjonen av kampfly måtte avhenge av sammenhengen?

Et enkelt kampfly er intuitivt sett det UML-spesifikasjonen definerer som en "node". En node er i UML en fysisk lokalitet der en lang rekke systemer er implementert. Imidlertid er ikke lenger en enkelt plattform selvstendig i samme grad som tidligere, ettersom en hel del av dets funksjonalitet er realisert gjennom et nettverk. En kampflygruppe kan distribuere funksjonalitet seg imellom, eksempelvis ved kooperativ posisjonering av mål ved hjelp av ESM-sensorer. Taktiske datalinker, som Link 16, er distribuert over mange plattformer, og får input fra flere systemer utenfor det de fleste tolker å være kampflysystemet (eksempelvis LOS-systemet). I et slikt "system av systemer"-perspektiv er det i mange sammenhenger vanskelig å trekke opp klare grenser.

Et kampfly kan modelleres i UML med bruk av "Node", "Actor" eller "Class. En "node" er en fysisk lokalitet i UML (f eks en PC), mens NAF v3 ikke tolker en node som en fysisk samlokalisering. Her er en node en logisk eller organisatorisk "entitet" som gjerne kan være fysisk distribuert. Et "system" eksisterer ikke i UML eksplisitt, men fremstilles i NAF v3-eksempelet som en "class". Link 16 er et eksempel på system som er implementert på mange "fysiske noder". Det samme kan sies om kampflyet.

Kampflyprosjektet på FFI har foreløpig valgt å tolke det slik at en enkelt F-16 intuitivt sett best

representeres med en UML node og fremstilles i modellen med stereotypen «OperationalNode» brukt på UML-elementet "Node".



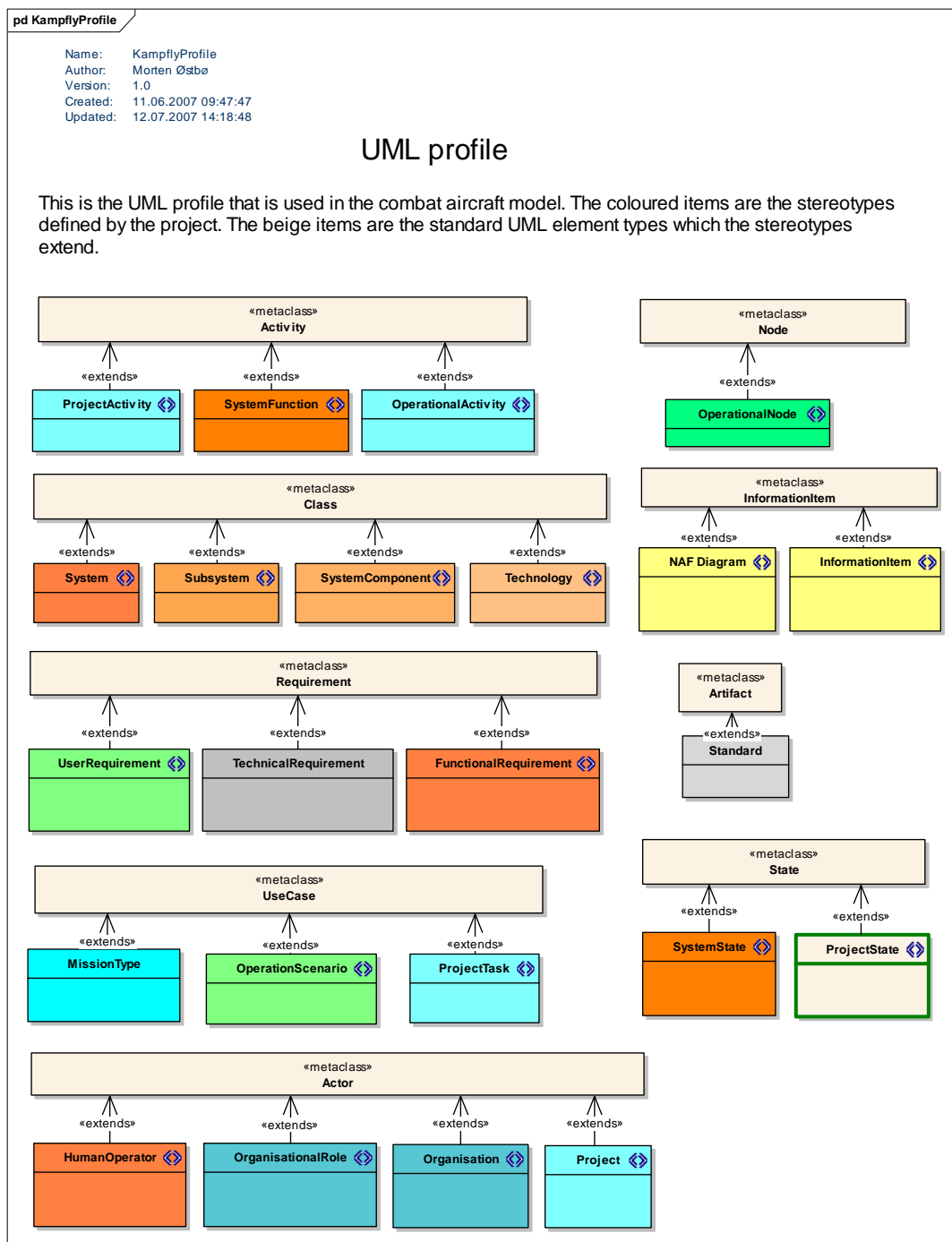
Figur 10.3 Fem alternative måter å representere kampfly i en UML-modell.

10.5 Metamodell og UML-profil for kampflymodellen

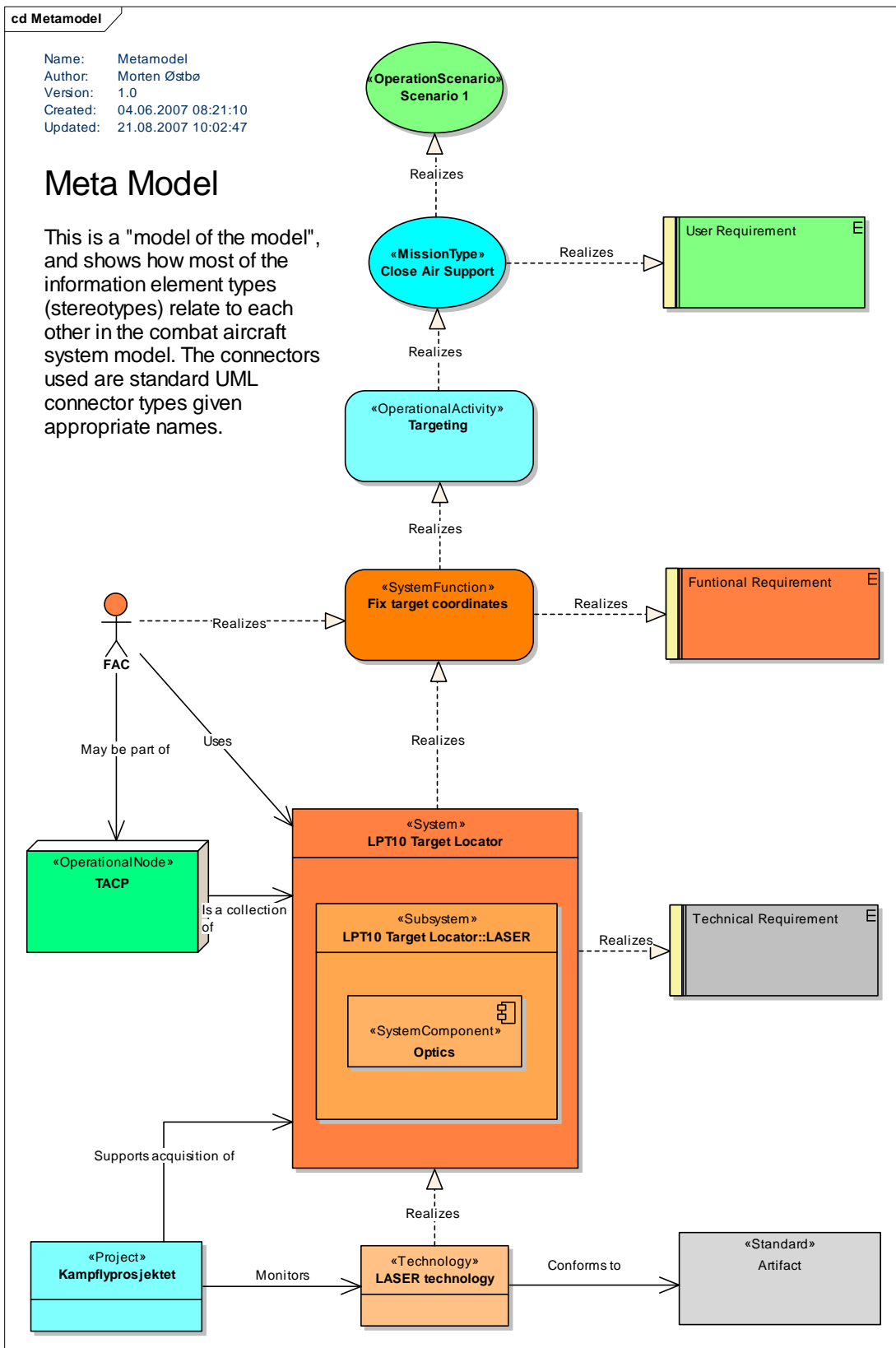
Da det ikke finnes nasjonale beslutninger, NATO-veiledninger eller defacto standarder blant kampflyaktørene, har kampflyprosjektet på FFI utviklet en egen UML-profil og en egen metamodell.

Selve symbolbruken eller metamodellen er ikke av stor betydning for kampflyprosjektet før det blir aktuelt å utveksle modeller og delmodeller med andre. Det betyr per i dag lite om man velger å bruke stereotypene «OperationalActivity», «SystemFunction», «OperationalFunction» eller «SystemActivity». Det viktige nå er at modellen er logisk og intuitivt bygget opp. Det vil uansett foregå en hurtig utvikling på dette feltet. Av større betydning på kort sikt er det imidlertid at kampflyprosjektet bygger opp kompetanse og utforsker mulighetene, slik at vi kan være med i debatten omkring felles løsninger. Når felles løsninger foreligger, vil kampflymodellen ikke være større enn at den relativt enkelt kan tilpasses den nye metodikken.

Kampflyprofilen og kampflymetamodellen er inspirert av UPDM og NAF v3.



Figur 10.4 UML-profil for kampflymodellen. De fargede symbolene er de egendefinerte typene elementer i modellen. Disse forlenger en rekke standard UML-elementer, her vist i lys brun farge.



Figur 10.5 Metamodellen for kampflymodellen. Denne viser hvordan de ulike elementene henger logisk sammen. Foreløpig er standard UML relasjonstyper benyttet. Man kan gi hver relasjon et eget navn hvis ønskelig, som det er gjort her.

10.6 Eksempeldiagrammer fra modellen

De følgende kapitlene inneholder noen eksempler på de aktuelle NAF arkitekturproduktene, samt noen egendefinerte diagramtyper, fra kampflymodellen. Diagramtypene fra NAF er ikke i direkte overensstemmelse med hverken NAF v2 eller NAF v3.


10.6.1 NAV-1

NAV-1 skal gi en oversikt over selve modellen – en slags ”hva, hvem, når, hvordan og hvorfor”. NAF foreslår å implementere dette i en form for tabell, eller et strukturert pakkediagram. NAV-1 kan også implementeres i en UML ”class” med stereotypen «ArchitecturalDescription», og en rekke ”tagged values”. Prosjektet har foreløpig valgt å kombinere denne typen informasjon med en startside for modellen. Informasjonen er foreløpig ikke strukturert. I stedet gis en tekstlig innledning til modellen. Denne siden kommer opp automatisk når man starter opp modellfilen i Enterprise Architect. Ved å klikke på lenkene på startsidene, kommer brukeren videre til nye portaler som følger samme oppskrift. De gir en tekstlig innledning til temaområdet, samt inneholder lenker videre til nye oversiktssider og diagrammer.

pd Start page, Norwegian Combat Aircraft System Model

Norwegian Combat Aircraft System

Introduction



[Klikk here to enter model](#) Assets Dictionary

Model Overview Diagram

This architecture model is under development by the FFI project "1026 - Nye kampfly - assistanse til Forsvaret II, delprosjekt 3 - Teknologi og arkitektur". The work on this model began in the spring of 2006. The goal is to explore the usefulness of architecture modeling as a tool to improve interoperability. The rationale for this endeavor is documented in the report "FFI/RAPPORT-2005/03321: Nye kampfly - Fellesoperativ og flernasjonal Interoperabilitet".

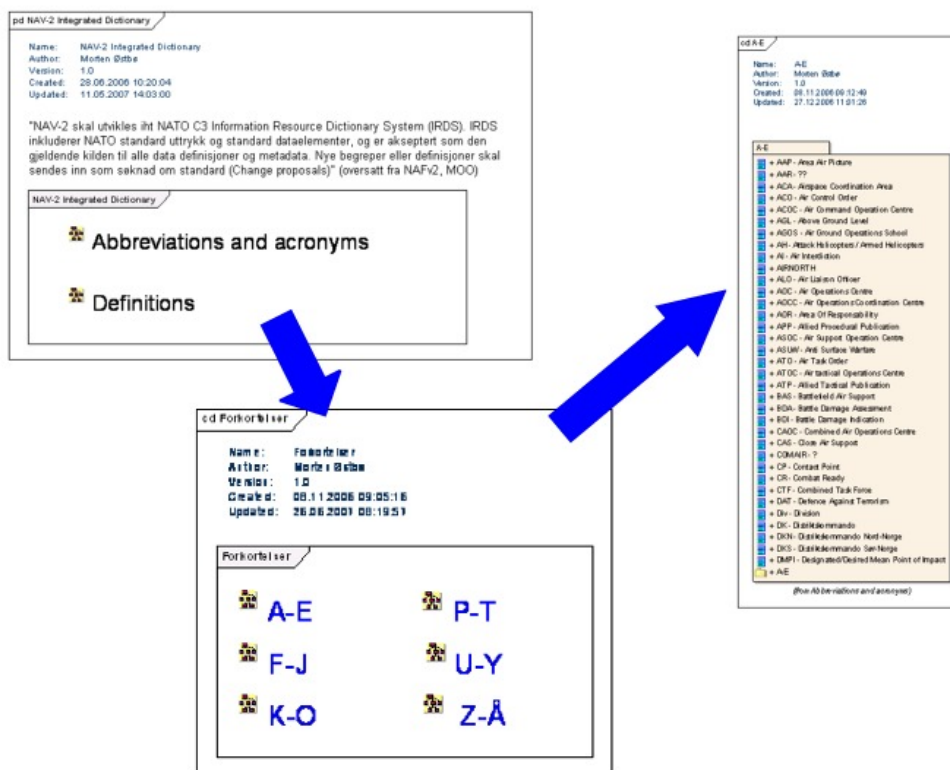
The model is currently focused on air-to-ground operations, specifically CAS - Close Air Support. This type of operation will be used as a test case for UML-modeling according to the NAF - NATO Architecture Framework. At the same time, the model is used to support the ongoing analysis of possibilities in future air to ground operations posed by the acquisition of a new combat aircraft system.

The tool used to develop the model is Enterprise Architect from Sparx Systems. The model may be made available in the proprietary file format .eap, in html, XML or as an RTF report. A free version of Enterprise Architect may be downloaded from the Internet. This allows you to view the model, but not to edit it. For more information about the model, contact either Morten Østbø or Jakob Gakkestad at FFI.

Figur 10.6 Startsidene i kampflymodellen. Denne fungerer som NAV-1.

10.6.2 NAV-2

NAV-2 skal sikre felles forståelse av begreper og en rask kilde til betydningen av forkortelser. NAF v2 krever at NAV-2 skal utvikles i et format som er kompatibelt med ”NATO C3 Information Resource Dictionary System (IRDS)”. IRDS inkluderer NATO standard uttrykk og standard dataelementer, og er akseptert som den gjeldende kilden til alle datadefinisjoner og metadata. Nye begreper eller definisjoner skal sendes inn som søknad om standard (Change proposals). Kravet om å følge IRDS er ikke tatt med videre til NAF v3. Kampflyprosjektet har implementert NAV-2 i form av pakker som inneholder en klasse for hvert begrep eller forkortelse.



Figur 10.7 NOV-2 i kampflymodellen består av en rekke lenkede pakke- og klassediagrammer som inneholder definisjoner og forkortelser.

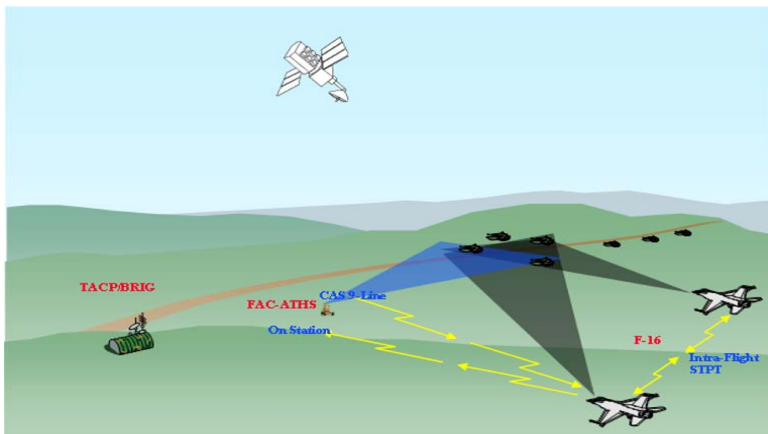
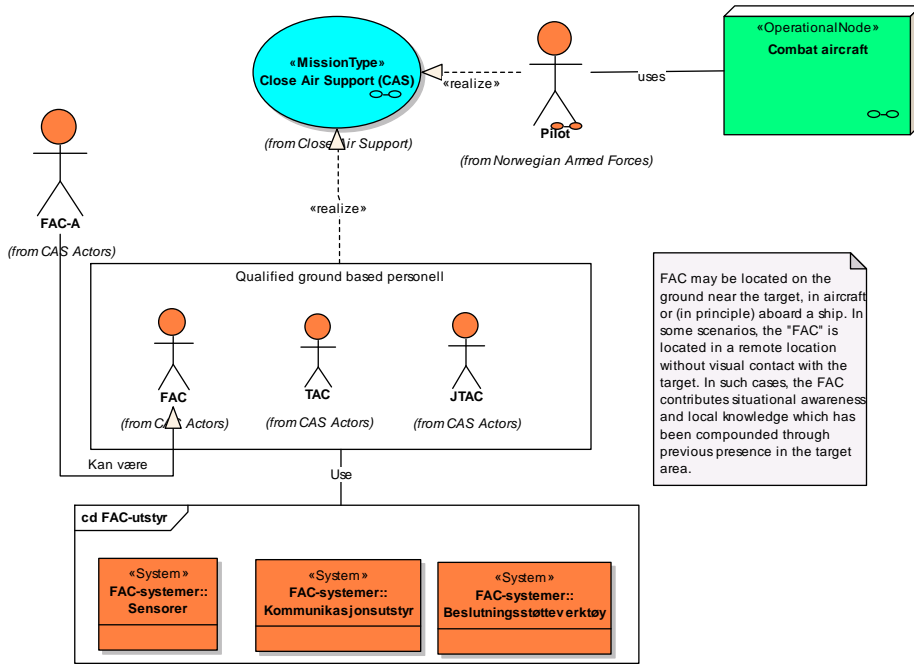
10.6.3 NOV-1

NOV-1 (High Level Operational Concept Diagram) skal gi brukeren av en modell ”the big picture” – et overordnet innblikk i konsepter. I metamodelen til kampflyprosjektet befinner NOV-1 seg på nivå med operative scenarier og oppdragstyper («OperationalScenario», «MissionType»). I henhold til NAF kan man bruke ulike typer grafikk, gjerne animasjoner, sammen med noe tekst til å lage et NOV-1. CAS-konseptet er fremstilt i ”NOV-1 High Level Operational Concept, Close Air Support” ved hjelp av et enkelt UML-diagram sammen med en mer tradisjonell konseptskisse og noe tekst. Operative scenarier (plansituasjoner) for kampfly bør beskrives i NOV-1 i videre arbeid.

Name: NOV-1 High Level Operational Concept Diagram, Close Air Support
 Author: Morten Østbø
 Version: 1.0
 Created: 06.12.2006 11:16:47
 Updated: 21.08.2007 09:03:12

Close Air Support Portal

Close air support is conducted in cooperation between weapons delivery platform aircrew and specially trained personell external to the weapons delivery platform. The crew of the weapons delivery platform works in concert with personell either on the ground or in aircraft to build the required situational awareness, and ensure the weapons are delivered on the correct target, with minimal undesired effects.



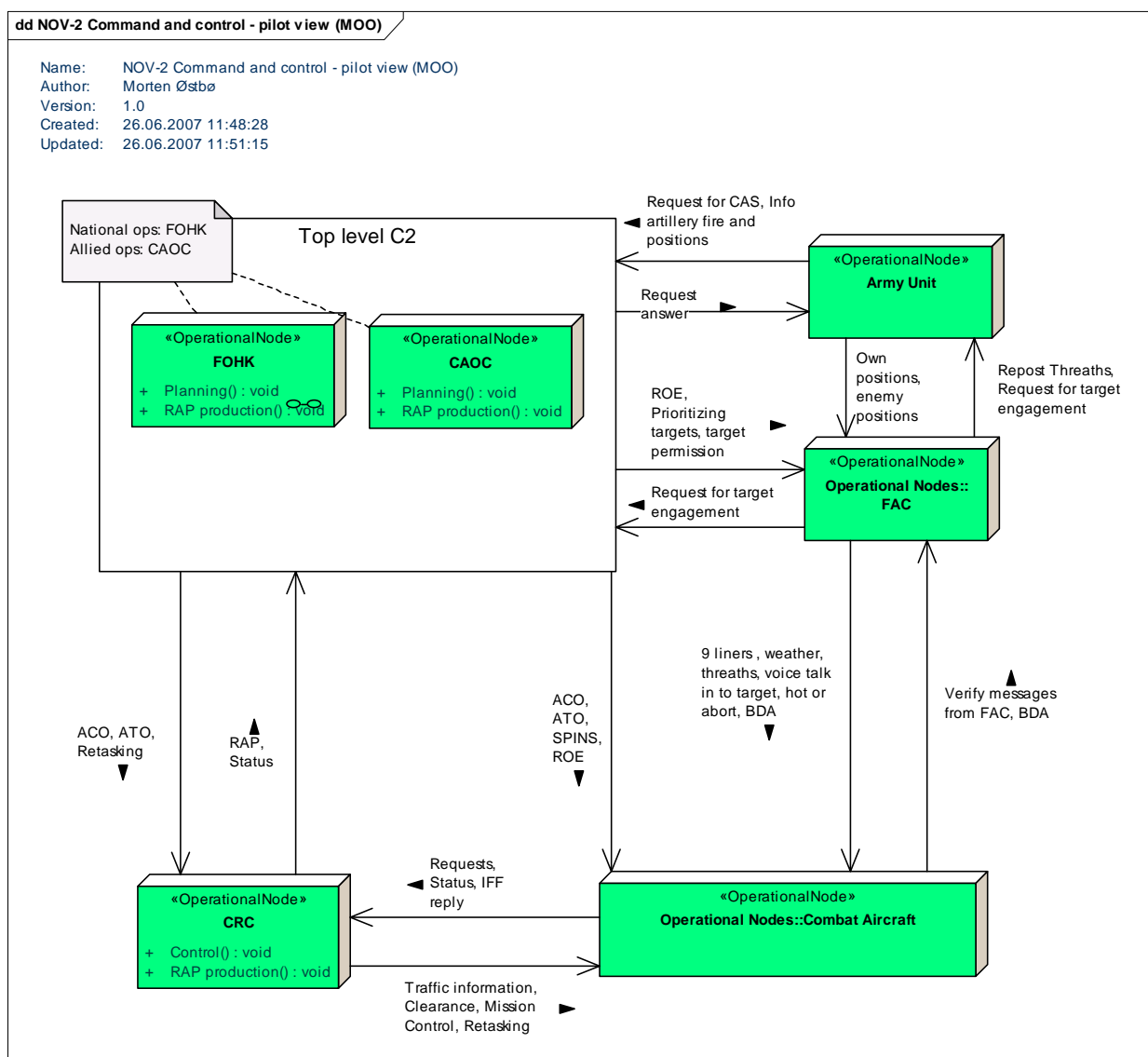
Figur 10.8 NOV-1 (High Level Operational Concept Diagram) for CAS (Close Air Support).

10.6.4 NOV-2

Utvexling av operativ informasjon mellom operative noder er fremstilt i ”NOV-2 Operational Node Connectivity Diagram, Close Air Support”. Informasjonsutvekslingen finnes igjen på tabellform i NOV-3. Det er viktig å merke seg at det kun er type informasjon som vises i et NOV-

2, ikke løsningen som brukes for å realisere utvekslingen. I NOV-2 inkluderes gjerne operative aktiviteter på hver av nodene. Disse aktivitetene finnes igjen i NOV-5.

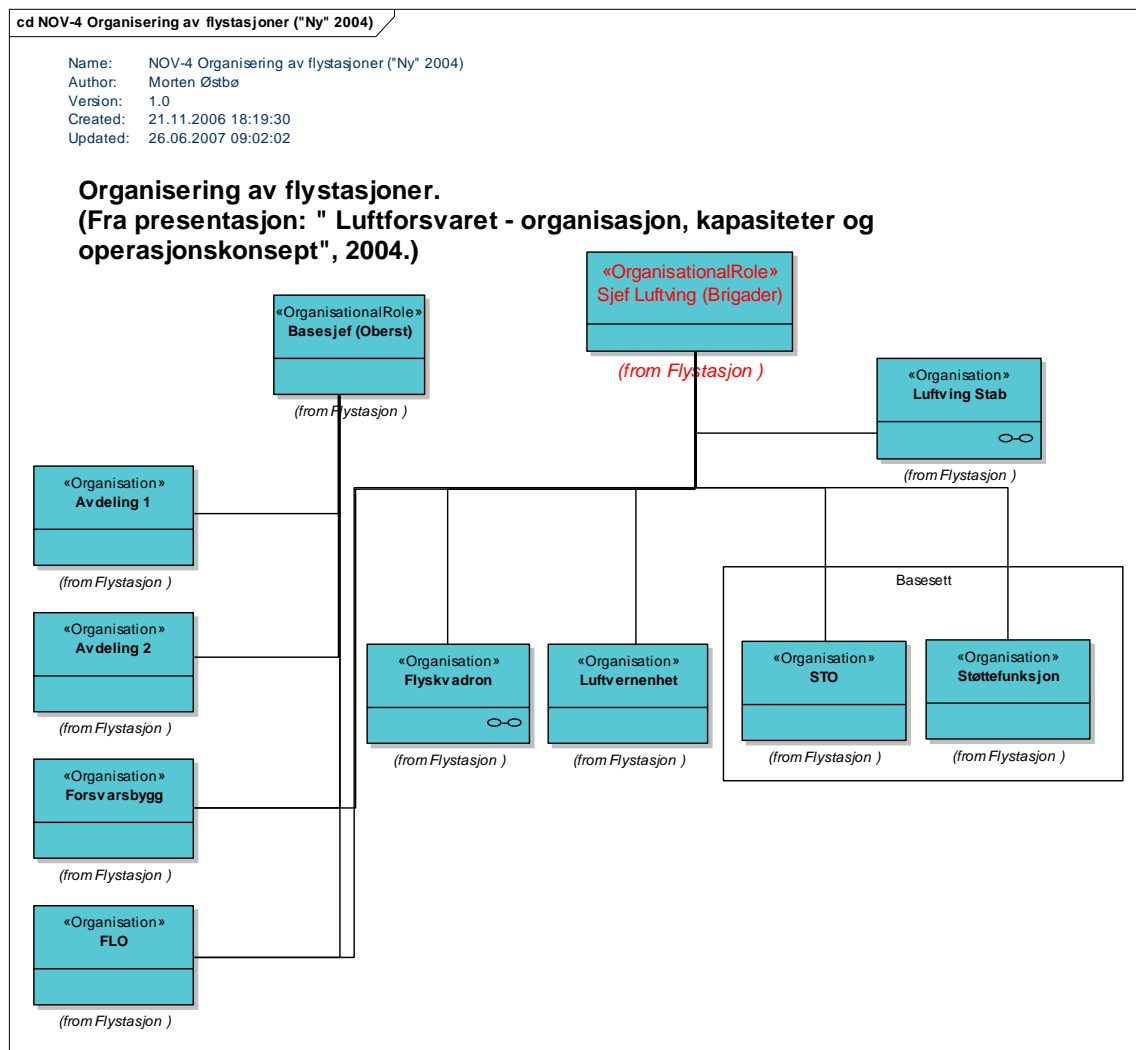
I metamodellen til kampflyprosjektet befinner et NOV-2 seg på nivå med «OperationalNode» og «OperationalActivity». Operative noder ble i utgangspunktet tolket som fysiske noder i kampflymodellen. Tolkningen av begrepet "node" i NAF v3 går imidlertid noe bort fra en fysisk samlokalisering, og mer i retning en logisk gruppering. Kampflyprosjektet har tatt dette inn i sin modell. Noder kan derfor være en hvilken som helst logisk gruppering av "noe", for eksempel en militær avdeling, et hovedkvarter eller en distribuert, men relatert, gruppe av systemfunksjoner. UML "deployment diagram" er brukt for å ta frem NOV-2 i kampflymodellen.



Figur 10.9 NOV-2 (Operational Node Connectivity Diagram) i form av et UML "deployment diagram"

10.6.5 NOV-4

NOV-4 "Organisational Relationships Chart" er et velkjent organisasjonsdiagram. I kampflymodellen er stereotypene «OrganisationalRole» og «Organisation» brukt til å modellere henholdsvis en konkret stilling i organisasjonen og organisasjonen. UML "actor" er brukt og representert grafisk med rektangler i stedet for "pinnemennesker". Elementene i dette diagrammet skal kobles til operative aktiviteter og noder i videre arbeid med modellen.



Figur 10.10 NOV-4 "Organisational Relationships Chart"

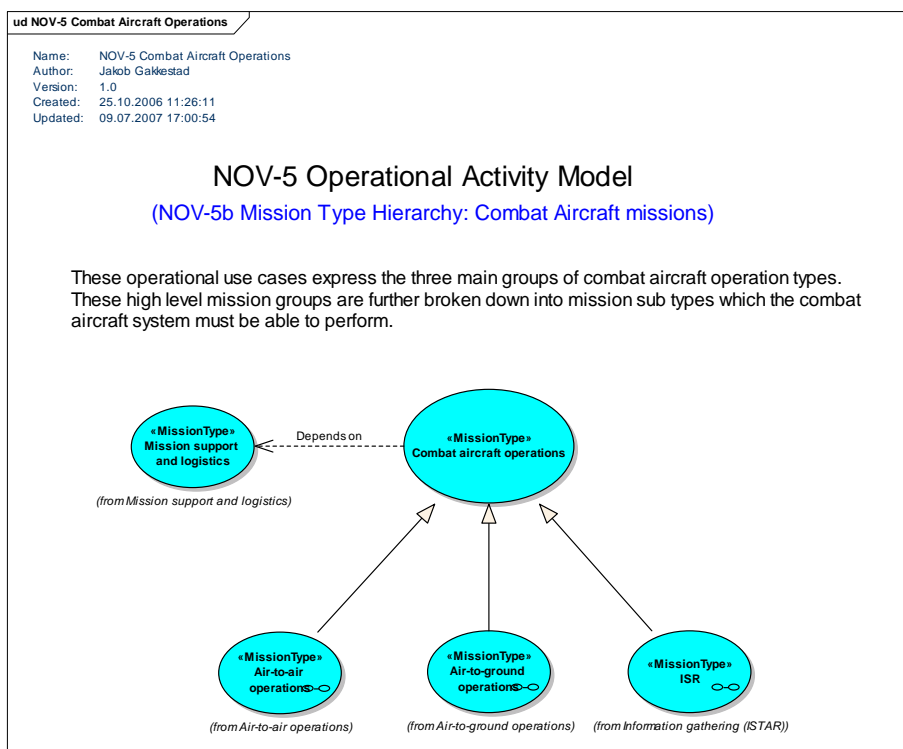
10.6.6 NOV-5

NOV-5 "Operational Activity Model" er den type arkitekturprodukt fra NAF som er mest brukt i kampflymodellen. Her har prosjektet funnet det nødvendig å utvide spekteret av diagramtyper i forhold til de som beskrives i NAF. Følgende typer NOV-5 er definert:

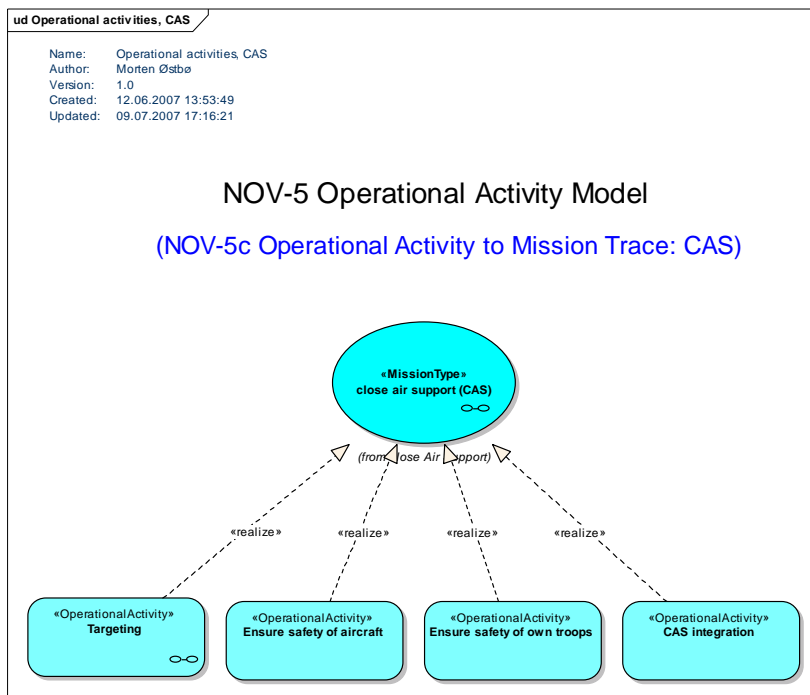
- ❑ NOV-5a "Mission Type to Operational Scenario Trace"
- ❑ NOV-5b "Mission Type Hierarchy"
- ❑ NOV-5c "Operational Activity to Mission Trace"

- NOV-5d "Operational Activity Hierarchy"
- NOV-5e "Operational Activity Description"

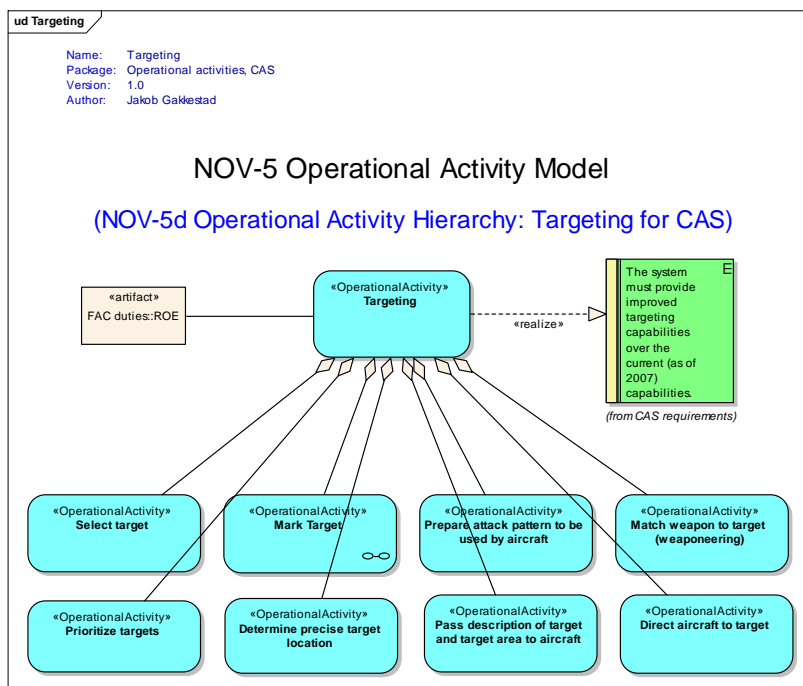
UML "use case diagram" ble brukt hyppig i begynnelsen av arbeidet, med UML symbolet for "use case" som representerte operative aktiviteter. Disse diagrammene var til dels "FAC-sentrisk" (eller operatørsentrisk), da de beskrev oppgavene som blir utført av FAC (Forward Air Controller) i dag. Dette var nyttig for å forstå hva en FAC gjør, som grunnlag for å forstå hvilke oppgaver som må løses i et fremtidig konsept. Modellen anvender nå i større grad stereotypen «OperationalActivity» representert med UML-elementet "activity". Disse operative aktivitetene er i størst mulig grad løsningsuavhengige. Det vil si at dette er oppgaver som systemet som helhet må løse for å gjennomføre oppdraget, hvilket i tilfellet CAS er å levere rett tilpasset effekt på rett sted til rett tid.



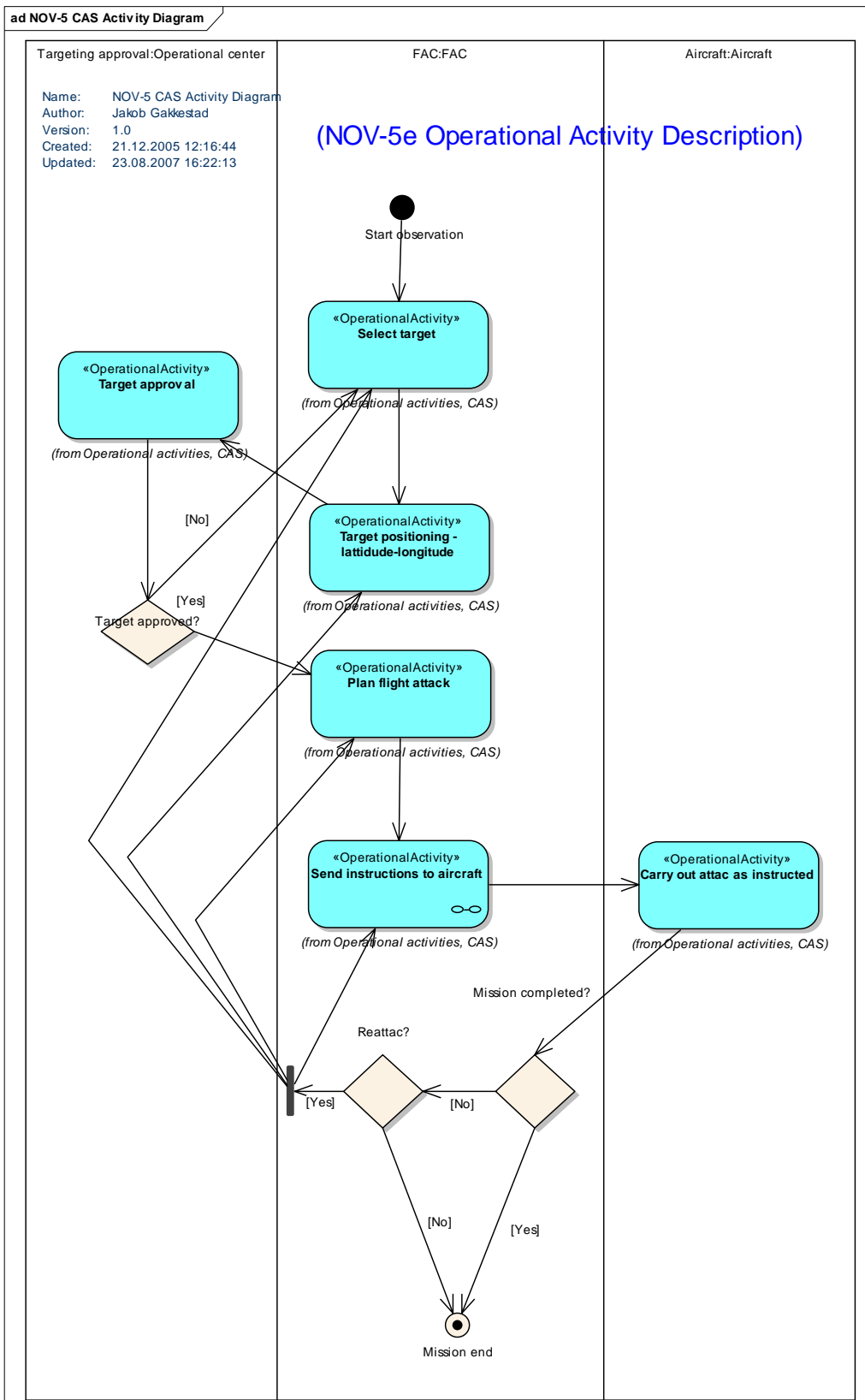
Figur 10.11 NOV-5b "Mission Type Hierarchy" er en egendefinert diagramtype som viser hvilke deloppdrag som inngår i en oppdragstype. Her vises hovedgruppene kampflyoppdrag.



Figur 10.12 NOV-5c "Operational Activity to Mission Trace" er en egendefinert diagramtype som viser hvilke operative aktiviteter som inngår i å løse en oppdragstype. Disse oppgavene er uavhengige av løsningsform.



Figur 10.13 NOV-5d "Operational Activity Hierarchy" er en egendefinert diagramtype som viser en nedbrytning av de operative aktivitetene fra NOV-5c i mer konkrete deloppgaver. Aktuelle krav kan gjerne inkluderes i slike diagrammer hvis plassen tillater. Ellers er krav modellert i egne diagrammer.



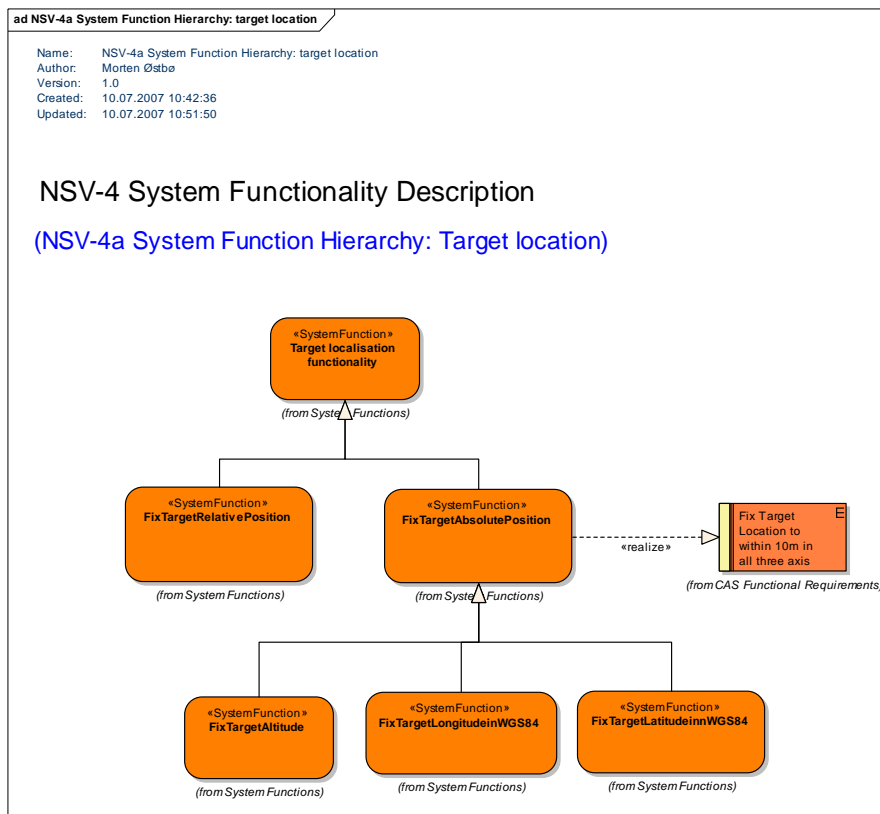
Figur 10.14 NOV-5e "Operational Activity Description" beskriver flyten i en aktivitet. Aktivitetene er gruppert i vertikale "swim lanes" etter hvilke aktører eller noder som utfører dem.

10.6.7 NSV-4

NSV-4 (Systems Functionality Description) fokuserer på systemfunksjoner, altså på hva systemet skal gjøre helt konkret for å realisere de operative aktivitetene som ble beskrevet i NOV-5. NSV-4 viser også hvilke systemer og delsystemer som realiserer funksjonene. NSV-4 gir en sporbarhet fra de (delvis) løsningsuavhengige aktivitetene som ble beskrevet i NOV-5 til det løsningsavhengige nivået med konkrete systemfunksjoner, systemer og teknologier.

Alle elementer i NSV er oransje i kampflymodellen. Dette harmonerer med fargebruken i NAF v3. Kampflyprosjektet har komponert en rekke egne avarter av NSV-4, ettersom NSV-4 i NAF mangler noen aspekter som prosjektet har behov for. Følgende ulike typer NSV-4 er definert for kampflymodellen:

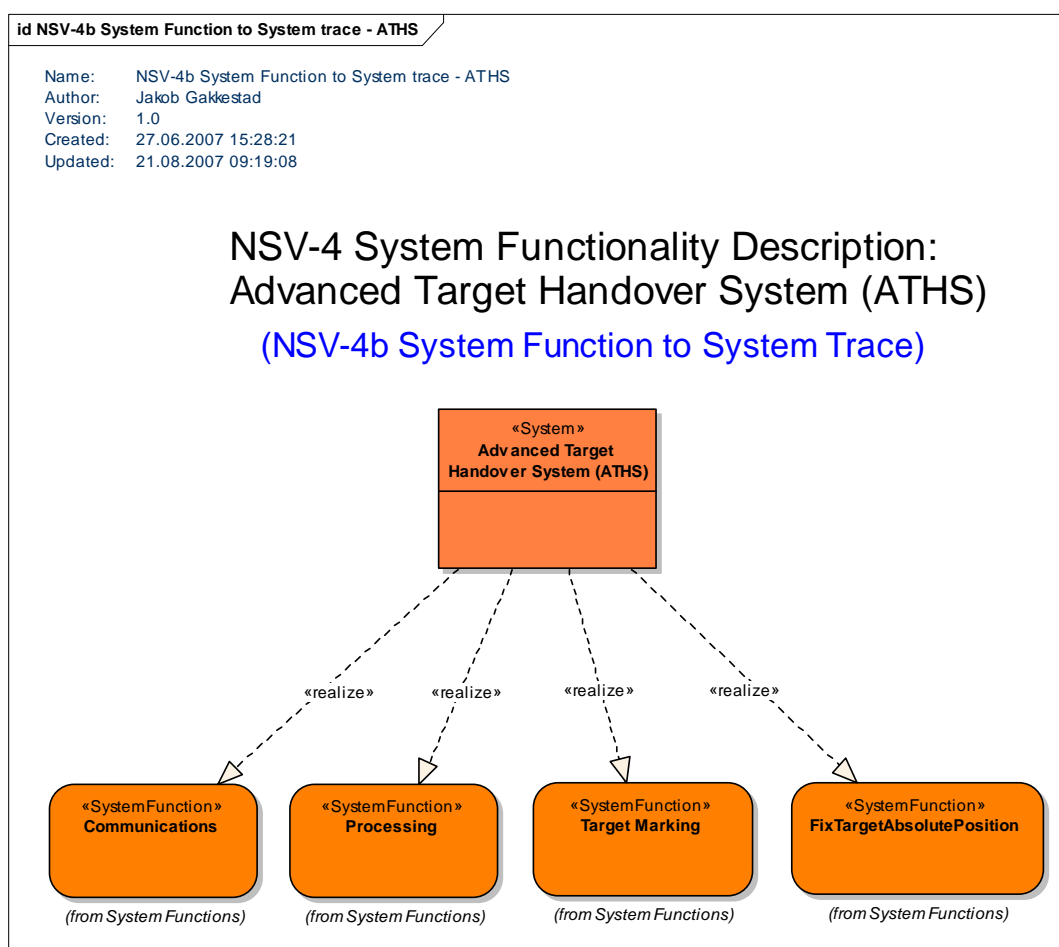
- ❑ NSV-4a "System Function Hierarchy"
- ❑ NSV-4b "System Function to System Trace"
- ❑ NSV-4c "System to Subsystem Decomposition"
- ❑ NSV-4d "Subsystem to Component Decomposition"
- ❑ NSV-4e "Subsystem to Technology Trace"



Figur 10.15 NSV-4a "System Function Hierarchy" er en egendefinert diagramtype som viser en nedbrytning av systemfunksjoner. De dypeste nivåene av en slik nedbrytning er mer løsningsavhengig enn det øverste nivået, som i stor grad tilsvarer de operative aktivitetene.

Kampflyprosjektet har definert stereotypen «SystemFunction» og valgt å representere denne informasjonstypen med UML "activity". Hvor grensen mellom operative aktiviteter og systemfunksjoner skal gå er foreløpig noe uklart. Det kan ofte virke like naturlig å bruke funksjonsbegrepet som aktivitetsbegrepet. Således kunne stereotypen «OperationalActivity» erstattes helt av «SystemFunction». Begrepet "aktivitet" gir gjerne inntrykk av noe mennesker gjør, mens kampflymodellen søker å beskrive det som "systemet" som helhet gjør. Det er da inneforstått at operatører som FAC og flygere er integrerte deler av systemet. Foreløpig brukes systemfunksjoner til å beskrive en dypere dekomponering av hva systemet må gjøre enn det aktivitetene gjør. Stereotypen «OperationalActivity» befinner seg altså i kampflymodellen på et noe høyere abstraksjonsnivå enn systemfunksjonene.

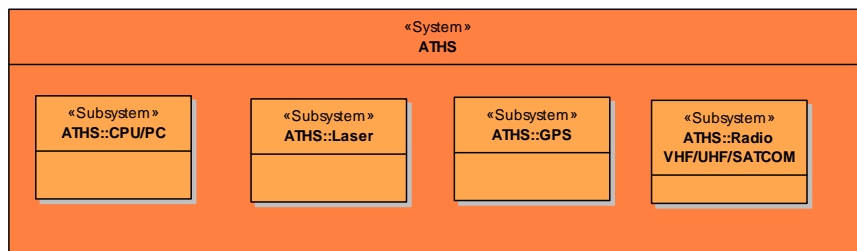
Systemfunksjoner, også kalt "systems use cases" i NAF v2, representerer indirekte de funksjonelle kravene til systemet. Disse kravene er eksplisitt uttrykt i kampflymodellen med bruk av UML kravsymbolet i det egendefinerte "Requirements View". Koblingen mellom systemfunksjoner og operative aktiviteter skal modelleres i NSV-5 i videre arbeid. Dette gjøres for å gi en tydelig og sporbar overgang mellom det operative domenet og systemdomenet i modellen.



Figur 10.16 NSV-4b "System Function to System Trace" er en egendefinert diagramtype som viser hvilke systemfunksjoner som realiseres av et gitt system.

Name: NSV-4c Advanced Target Handover System
 Author: Morten Østbø
 Version: 1.0
 Created: 10.07.2007 08:35:59
 Updated: 21.08.2007 09:16:21

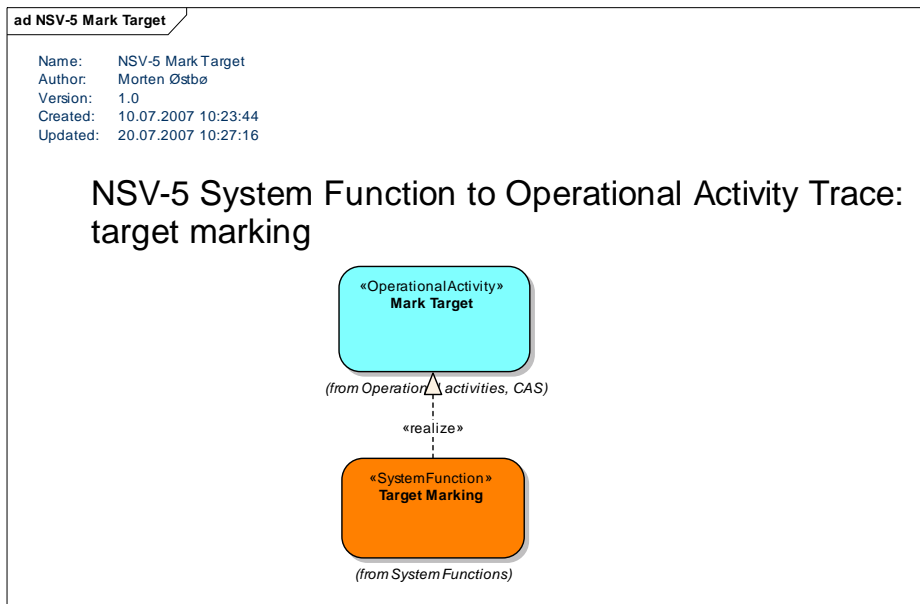
NSV-4 System Functionality Description (NSV-4c System to Subsystem Decomposition)



Figur 10.17 NSV-4c "System to Subsystem Decomposition" er en egendefinert diagramtype som viser hvilke delsystemer et system består av.

10.6.8 NSV-5

NSV-5 "System Function to Operational Activity Traceability Matrix" er i henhold til NAF, som navnet tilsier, egentlig en matrise. I NAF v3 vises den imidlertid som et diagram med sporbarheten fra operativ aktivitet via funksjoner til system. Det store antallet systemer, funksjoner og aktiviteter som er forbundet med hverandre taler for bruk av en matrise. For å visualisere sporbarheten i kampflymodellen brukes imidlertid diagrammer foreløpig. Enterprise Architect tillater også å generere matriser integrert med modellen. Dette bør bli implementert i videre arbeid.

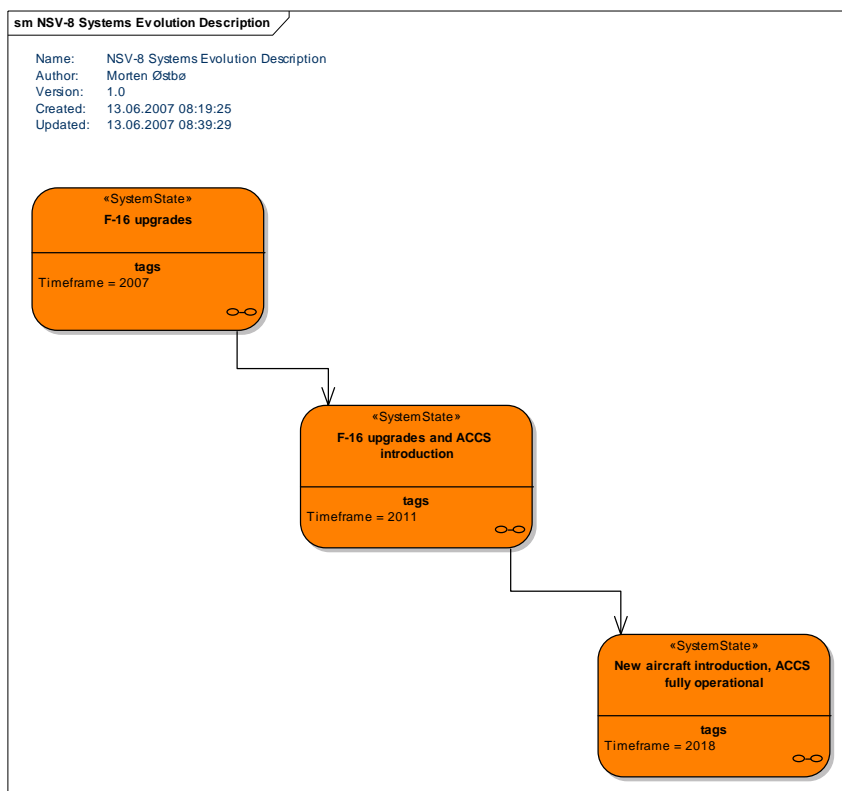


Figur 10.18 NSV-5 "System Function to Operational Activity Traceability Matrix" er et diagram som ivaretar sporbarheten mellom de operative aktivitetene og systemfunksjonene. Både aktivitetene og funksjonene er videre beskrevet i andre diagrammer. NSV-5 kan ofte være mest hensiktsmessig å implementere i en matrise, slik navnet uttrykker. Muligheten for å opprette modellintegreerte matriser ligger innebygget i Enterprise Architect, men er ennå ikke utforsket av prosjektet.

10.6.9 NSV-8

NSV-8 "Systems Evolution Description" gir en oversikt over planlagte og foreslåtte endringer i systemet. NSV-8 er ikke av stor betydning i det tidlige stadiet av arbeidet med nærstøtteoperasjoner. Fokus her har vært å beskrive nåsituasjonen. NSV-8 vil bli viktig når innføring av nye løsninger skal beskrives.

NSV-8 kan implementeres med PowerPoint e.l. som en tidslinje. NAF v2 har ikke forslag til implementasjon med UML. Det er likevel mulig å bruke et "State Chart Diagram" som viser overgang fra en tilstand til en annen. Tilstandene kan vise sentrale egenskaper og tidsangivelser. Hver "state" kan som vanlig i EA kobles videre til diagrammer og eksterne dokumenter som viser og omtaler de planlagte og foreslåtte systemendringene ytterligere. Sammen med portalene og koblinger fra de aktuelle diagrammene gjør NSV-8 det mulig for en bruker å finne frem til arkitekturprodukter for ønsket tidsrom.



Figur 10.19 “Systems Evolution Description” er et diagram som ikke er tatt med videre fra NAF v2 til NAF v3. Det viser den planlagte overordnede utviklingen i et system.

10.6.10 NTV-1

NTV-1 “Technical Standards Profile” skal i henhold til NAF gi koblingen mellom systemer og standarder. Kampflyprosjektet har foreløpig valgt å bruke NTV-1 til å gi en samlet fremstilling av CAS-relevante dokumenter. NTV kan også inneholde tekniske spesifikasjoner.

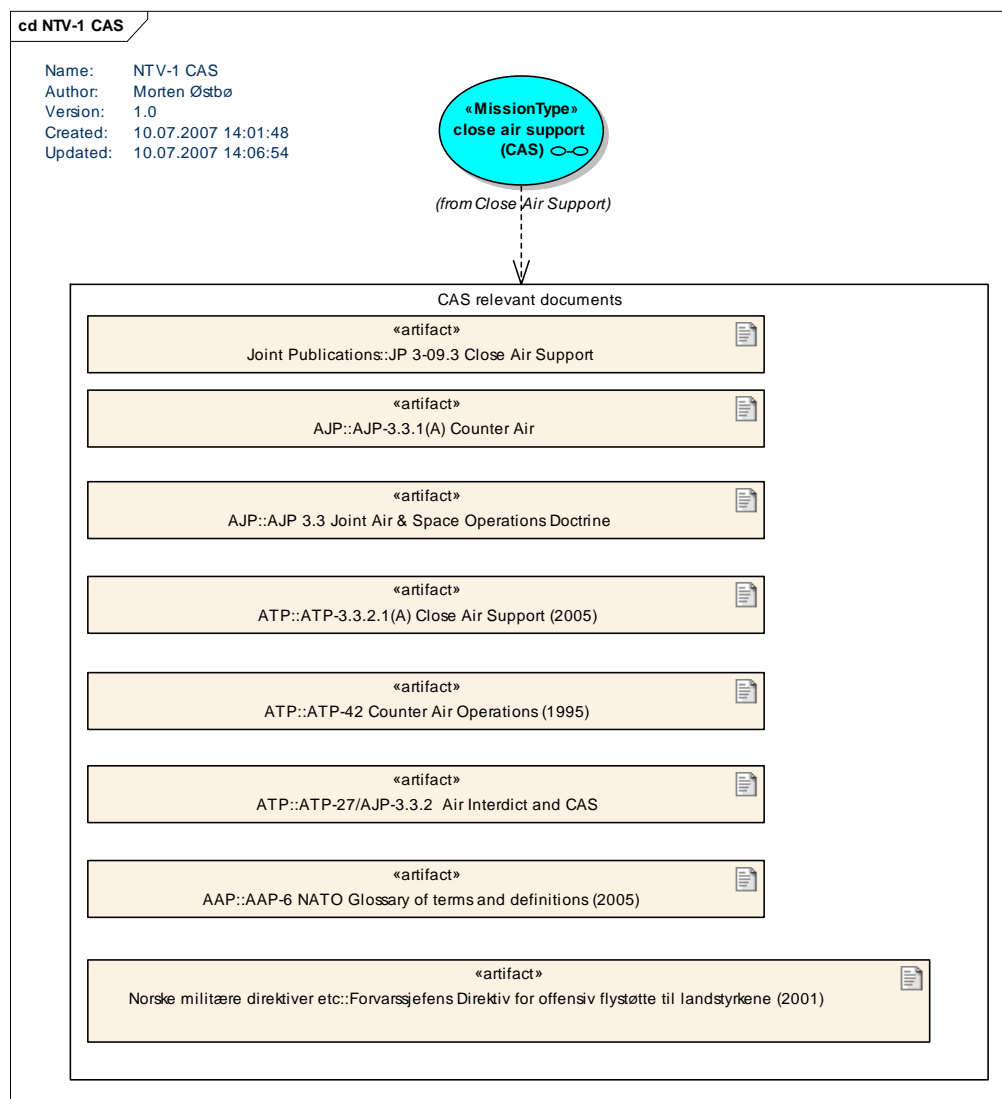
10.6.11 Kravdiagrammer

Kravstyring er helt sentralt i enhver anskaffelsesprosess, og i ”systems engineering” generelt. Å samle krav og uttrykke dem klart er det første og viktigste steget mot å finne gode løsninger. Krav til fremtidige evner til nærstøtte er et av målene med den luft-til-bakke-analysen som er utført i kampflyprosjektet.

Dersom modellering skal kunne integreres i anskaffelsesprosesser i Forsvaret, og i tilstøtende aktiviteter, må krav inkluderes i modellene. Disse kravene må kobles til de operative evnene (“capabilities”) som skal realiseres med et system, og til de systemelementene som skal muliggjøre oppfyllelse av kravene.

UML og Enterprise Architect inneholder flere muligheter til å representere krav i modeller. UML-elementet ”requirement” er en eksplisitt representasjon av eksterne krav til systemet, altså for eksempel brukerkrav. Det er mulig å opprette ulike typer krav etter eget ønske gjennom bruk av stereotyper. Eksempelvis kan man opprette ”funksjonelle krav” og ”tekniske krav” som kan

fremstilles noe ulikt grafisk gjennom bruk av en egendefinert profil. På denne måten kan kravterminologi fra ”Konseptuell løsning for prosjekt 7600 Fremtidig kampflykapasitet” brukes i modellen. Kravtyper som ”samfunns mål”, ”effekt mål” og ”objekt mål” som er kjent hos interessentene kan altså representeres i UML og dermed gi konsistent terminologi.



Figur 10.20 NTV-1 “Technical Standards Profile” er valgt å vise hvilke dokumenter som gjelder for utførelsen av oppdragstypen CAS.

Krav omtales kort i NAF v2, men ser foreløpig ikke ut til å være inkludert i NAF v3. Et nasjonalt tillegg for kravstyring er diskutert med FLO/IKT. Kravstyring er heller ikke omtalt direkte i MODAF eller UPDM. NAF v2 bruker ”systems use cases” i NSV-4 til å representere funksjonelle krav implisitt. Systemfunksjonene som realiserer kravene representeres med class og class operations, og disse er koblet til ”system use cases” med ”realize”-relasjonen. Ellers omtaler ikke NAF v2 krav eksplisitt. Brukerkrav uttrykkes implisitt gjennom ”use cases” i NOV-5, dvs modeller av hva brukeren skal kunne gjøre med systemet.

Enterprise Architect inneholder tilstrekkelig funksjonalitet for å håndtere krav. Alle former for krav kan representeres og integreres i modeller. Krav kan importeres til modellen fra regneark, og RTF rapporter som lister opp kravene kan genereres automatisk. Sammenhengen mellom kravene, og mellom krav og andre elementer i modellen kan synliggjøres i diagrammer, matriser og i rapportvinduer. Krav kan også legges inn i egenskapene til modellelementer. Kravene kan fargekodes etter eget ønske, eller etter et predefinert system i EA som viser status for et krav, eksempelvis ”godkjent” eller ”implementert”. Kampflyprosjektet har foreløpig definert de tre kravtypene ”User Requirement”, ”Functional Requirement” og ”Technical Requirement”.

Modellering av brukerkrav hører hjemme i ”requirements model” hvis en bruker den generiske modellstrukturen i EA. I kampflymodellen er krav lagt i et eget ”Requirements View” på lik linje med ”Operational View” osv.

Kun noen få av mulighetene i EA for kravhåndtering er implementert i kampflymodellen til nå. Dette bør bli et viktig satsningsområde for videre arbeid med modellen. I første omgang bør alle krav som kan identifiseres i gjeldende dokumenter inkluderes i modellen.

11 Modellering i systemteknikk (”Systems Engineering”)

Vitenskapen og beste praksis rundt utvikling av systemer betegnes i Norge som ”systemteknikk”. Dette norske begrepet tilsvarende det engelske ”systems engineering”. Systemarkitektur er en naturlig del av systemteknikk. Likevel fremkommer ikke dette tydelig i NAF eller i mye av litteraturen om systemarkitektur. Til tider kan det være vanskelig å se forskjellen mellom de to fagområdene, eller hvordan de forholder seg til hverandre.

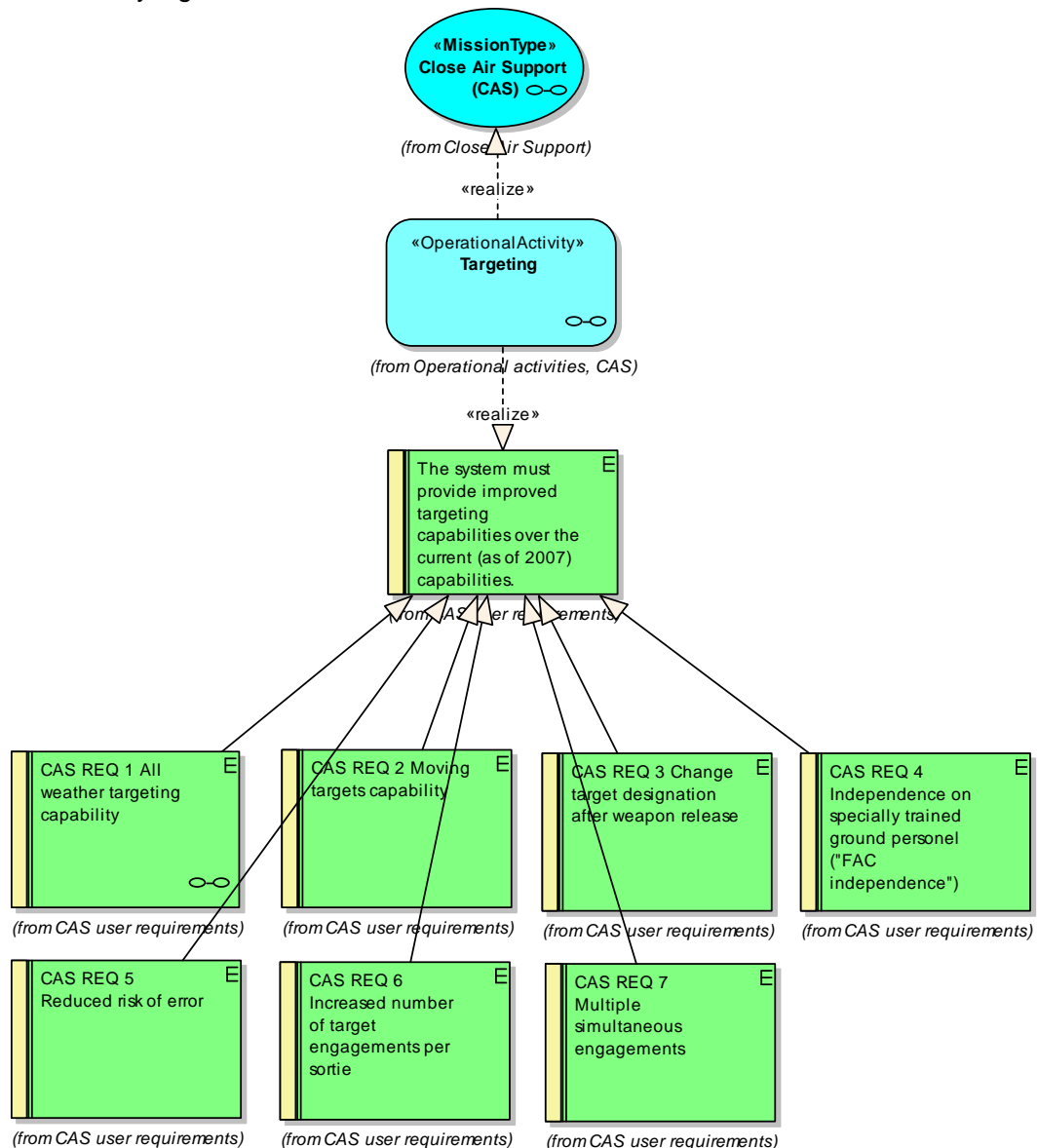
Systemteknikk er et veletablert fagområde. INCOSE – International Council on Systems Engineering – og ISO – International Standards Organisation – utgir sentrale dokumenter som beskriver prosesser og metoder for systemteknikk. Det finnes imidlertid ikke én enkelt akseptert standard prosess, eller ett sett standardverktøy som brukes innenfor industrien eller i andre miljøer.

Arkitekturrammeverkene (bl.a. NAF) er utviklet på siden av standardene for systemteknikk. Mange av konseptene og begrepene kommer imidlertid fra de sentrale ISO-standardene 1471 og 1220 for systemteknikk. Til tross for mange likheter mellom de to fagområdene, er fraværet av representasjon av krav i NAF en betydelig forskjell. Det kan også synes som en betydelig svakhet ved NAF at kravhåndtering ikke er inkludert eksplisitt, men snarere implisitt i produktene.

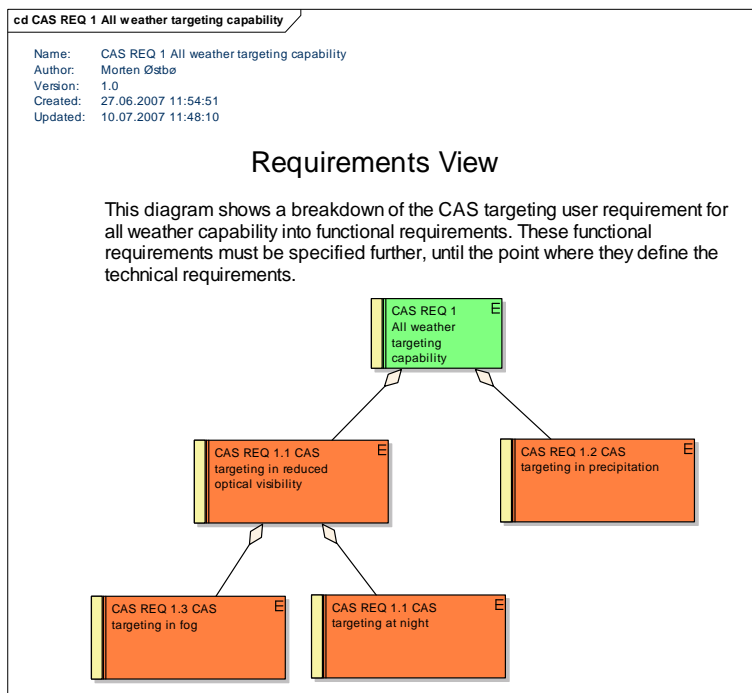
Name: CAS targeting user requirements
 Author: Morten Østbø
 Version: 1.0
 Created: 27.06.2007 11:36:35
 Updated: 21.08.2007 09:36:28

Requirements View: Proposed user requirements: CAS targeting capabilities

The user requirements express what the system is required by the user to be able to do. In this context, the "user" is the user of the Norwegian defence force as a whole, not necessarily e.g. the FAC.



Figur 11.1 Et kravdiagram som viser sporbarheten fra oppdrag gjennom operativ aktivitet til krav. Fargene i margene i kravsymbolene viser status for kravet. I dette tilfellet er de gule. Dette betyr at kravet er ”proposed”.



Figur 11.2 Kravdiagram som viser en konkretisering av brukerkrav i funksjonelle krav.

11.1 Fra dokumentsentriske til modellsentriske prosesser

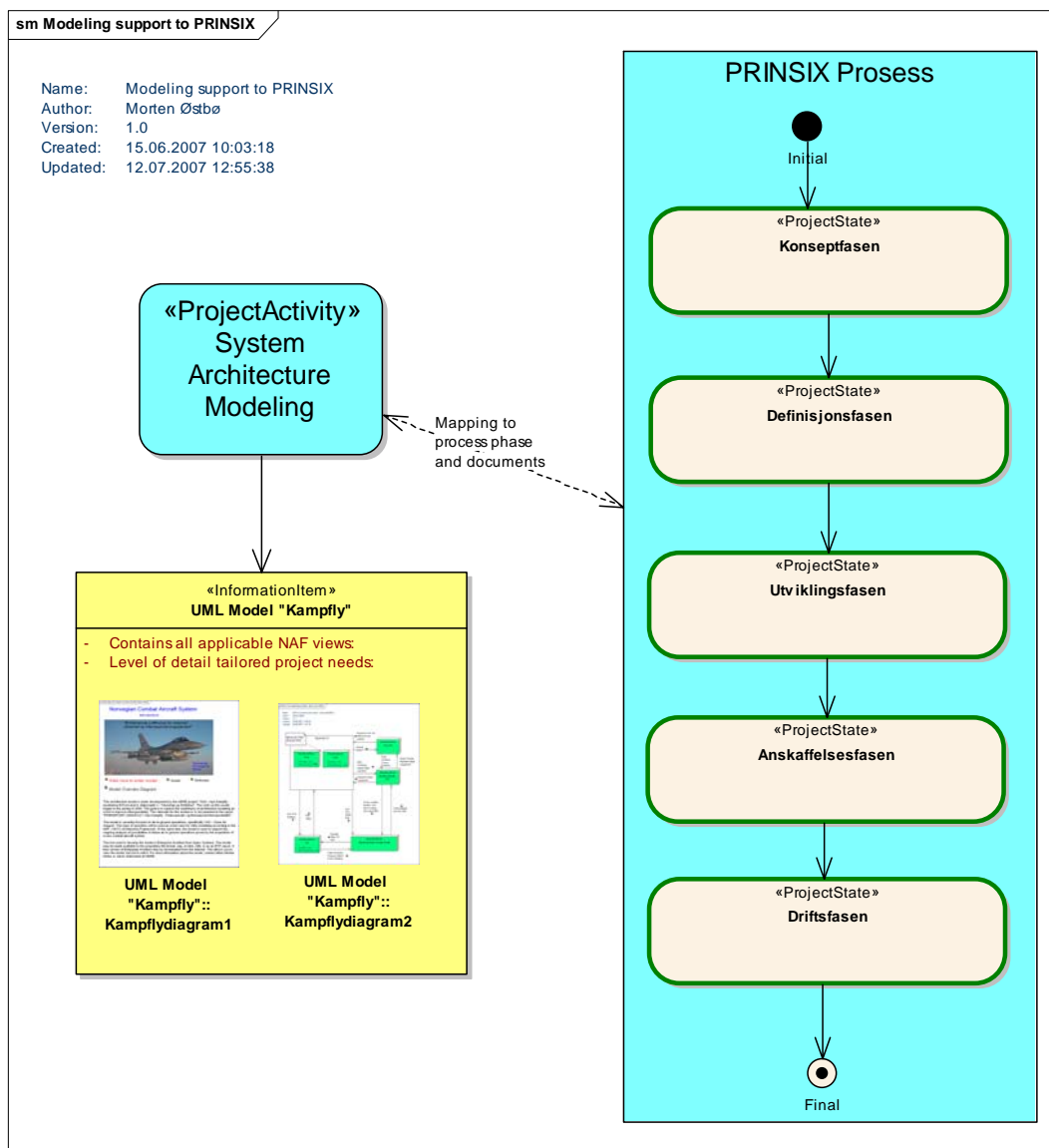
Modeller (les: tegninger og diagrammer) har fungert som støtte til systemutvikling, anskaffelser og forskning så lenge disse aktivitetene har pågått. Systematisk modellering etter standardisert formalisme og arkitekturrammeverk osv er imidlertid en nyere utvikling, og ikke særlig utbredt ennå. Det vi ser idag er begynnelsen på en utvikling i retning mer systematisk og gjennomtenkt bruk av modeller. Det er mulig at vi nå ser begynnelsen på en utvikling fra et dokumentbasert paradigme til et modellbasert paradigme.

”Model based development” og ”model based systems engineering” er sentrale begreper på vei inn i industrien. Systemteknikkmiljøet anerkjenner betydningen av modellering og rammeverk. Imidlertid er ikke modellering godt integrert i systemteknikk-prosessene. Hvordan dette skal gjøres er gjenstand for debatt. Temaet kompliseres av at det finnes en rekke modelleringspråk, rammeverk og systemteknikkprosesser.

Modellering med verktøy som UML bør ikke være en aktivitet på siden av, men integrert med, prosessene for både anskaffelser, utvikling, drift og avhending. Arkitekturmodellering bør altså være et integrert hjelpemiddel i hele livsløpet til systemer. I systemteknikk-kretser sies det at systemteknikk er ”95 % kommunikasjon”. Det er denne rapportens hypotese at systematisk modellering kan effektivisere denne kommunikasjonen betydelig.

11.2 Arkitekturmodellering som støtte til PRINSIX

PRINSIX er et eksempel på en nasjonal, dokumentsentrisk prosess for anskaffelser som har mange elementer fra "best practices" innen systemutvikling. I følge uformell kommunikasjon med FLO/IKT arbeides det der for å integrere modellering iht NAF med PRINSIX. PRINSIX har en sterk innvirkning på arbeidet i mange forskningsprosjekter på FFI. Særlig FFI-prosjektene som arbeider med støtte til systemanskaffelser har et aktivt forhold til dokumentstrukturen i PRINSIX. Modelleringsaktivitetene (med UML) på FFI er i dag initiativ av typen "bottom up". Det er sannsynligvis nødvendig med et visst "top down" engasjement for å klare å integrere metodisk modellering med anskaffelsesprosesser.



Figur 11.3 Modellering med UML er tenkt å støtte alle fasene i PRINSIX.

11.3 Ansvar og roller

Bredden i en modell som den av kampflysystemet, samt det betydelige antall felleselementer mellom mange prosjekter, gir et sterkt argument for at modellering må være et koordinert ”fellesprosjekt”. Kampflymodellen vil ikke bli ”komplett”, og den vil meget raskt bli utdatert, uten et klart plassert ansvar for å sørge for modellens kvalitet og uten en hensiktsmessig arbeidsdeling. Bred involvering er nøkkelen til god modellering. Områdeeksperter må involveres på alle plan. Dette bør skje uten krav om opplæring i UML for alle. Selve modelleringen i UML bør generelt sett ivaretas av et begrenset antall ”systemarkitekter”. Det bør finnes systemarkitekter primært på FFI, i FLO og på FK KKIS.

Det er ikke nødvendig at en person i hvert prosjekt (på FFI og ellers) eller hver organisatoriske enhet under FD lærer UML-modellering i dybden. Evnen til å lese modeller bør imidlertid tilegnes av de fleste som jobber på et teknisk nivå eller med operasjonsanalyse. Systemarkitektene bør opprettholde jevnlig kontakt med de operative miljøene og logistikkmiljøene. Samtidig må utvalgt personell i disse miljøene forpliktes til å bidra med informasjon om endringer og planer. Det blir så arkitektenes rolle å inkludere dette i modellene. Det er samtidig viktig at arkitektrollen defineres på en slik måte at arkitektene ikke blir ”skrivere”, men har en sentral rolle i å utforme et helhetlig forsvar.

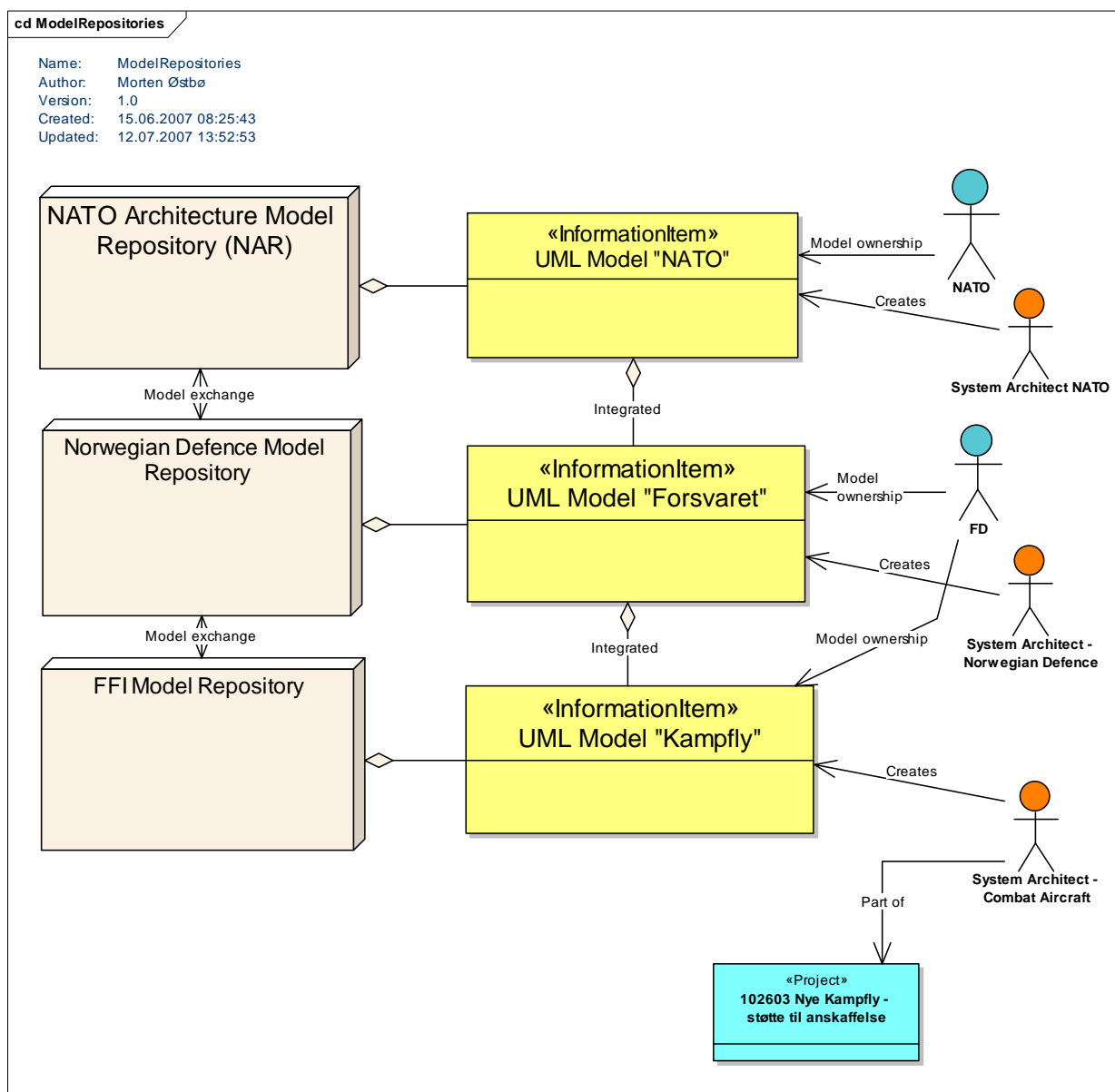
En systemarkitekt for kampflysystemet er avhengig av en lang rekke forhold som er utenfor hans/hennes kontroll. Slike felles problemstillinger som de fleste systemer er koblet til bør håndteres på høyt nivå, i samråd med systemarkitektene for de enkelte (del)systemene. Eksterne systemgrensesnitt må defineres i en helhetlig prosess. Deretter kan de konkrete løsningene innenfor hvert system utformes lokalt.

De aller fleste prosjekter på FFI og i FLO er relativt kortvarige, og de har en begrenset oppgave å løse. Således har prosjektene selv en kortsiktig og snever interesse av modellene de eventuelt bruker i sitt arbeid. De er derfor interessert i å bruke minst mulig tid på modellering, og de vil helst ikke bruke tid på å modellere ”ting de vet”. Modellering, slik det omtales her, har imidlertid relevans utover det enkelte prosjekt. Modeller bør derfor fortsette å eksistere, og i mange tilfeller vedlikeholdes, også etter at prosjektet er ferdig. For at resultatene av modelleringsarbeid ikke skal gå tapt, må modellene eies av en organisasjon, ikke av et prosjekt. Både eierskap av kampflymodellen (og andre modeller) og ansvaret for å vedlikeholde modellen må plasseres organisatorisk, og ikke være knyttet opp mot et prosjekt.

11.4 Felles modellbibliotek

Potensialet i å innføre den beskrevne metodikken er størst dersom modeller gjøres tilgjengelige for andre enn de som utviklet dem, og dersom modellene kan integreres i en felles modell av Forsvaret og NATO. NAF v3 understreker dette behovet, og omtaler et felles modellbibliotek for NATO – NATO Architecture Repository (NAR). Den nødvendige formaliseringen av vedlikehold og endringer i en slik modelldatabase fører gjerne med seg et betydelig element av byråkrati. De fleste som blir involvert i arkitekturmodeller vil nok føle behov for en mindre byråkratisk norsk

løsning i tillegg til fellesløsningen i NATO. Ulike aktører kan også ønske å etablere egne modellbibliotek. Eksempelvis kan det være hensiktsmessig å etablere en slik løsning på Forsvarets nettverk (FISBasis). Utveksling og integrering av modeller i en felles database bør følge et system. Man bør likevel åpne for bidrag som ikke er i henhold til noen standard der det ikke har vært ressurser til å modellere i henhold til NAF og med riktig formalisme.



Figur 11.4 Et forslag til struktur for modellbibliotek, modelleierskap og utførelse av modellering. Eierskapet til modellene legges hos FD og NATO i dette forslaget, mens modelleringsjobben gjøres av et begrenset antall systemarkitekter som er distribuert på ulike nivåer i organisasjonen under FD, dvs på FFI, FLO, FK KKIS, TRADOK osv.

11.5 Begynnerterskelen

Det vil oppleves av mange prosjekter som et betydelig løft å opprette modeller, og å finne en

koherent måte å jobbe på. Når metodikken først er etablert, vil dette imidlertid kunne bli "second nature" – en naturlig del av det daglige arbeidet. Man vil bli vant til å forholde seg til et sentralt modellbibliotek som den gjeldende informasjonskilden, og man vil bli vant til å bidra med delmodeller fra sitt arbeidsområde til dette sentrale biblioteket. Man vil kunne oppleve at man "ser" problemstillinger klarere når man bruker modellering som arbeidsverktøy sammenlignet med dokumenter. Når man må bestemme seg for den beste (eller "riktige") måten å beskrive en prosess eller et system med et grafisk beskrivesspråk, blir man mer opptatt av å kjenne prosessen eller systemet i dybden. Når man så har opparbeidet en erkjennelse av "tingenes" sammenheng, er denne erkjennelsen dokumentert på en konsis måte, og kunnskapen kan lett gjøres tilgjengelig for andre.

Mange spør: "Er ikke dette bare mer bryderi enn nyttig?", "Hvilken nytte har jeg av å bruke tid på å modellere osv?", "Er ikke dette å overkomplisere ting? Forsvaret er jo svært lite etter hvert!", "Trenger vi flere regler (standarder) å forholde oss til?". Som svar kan man spørre om ikke fraværet av regler skaper mer komplikasjoner enn noen nye regler?

Små prosjekter egner seg i utgangspunktet bedre for ustrukturerte arbeidsmetoder, eller mangel på metode, enn store. Alle prosjekter på FFI og i Forsvaret er per definisjon små i seg selv, men de inngår alle i en større sammenheng. Ethvert prosjekt må derfor ha et avklart forhold til denne større helheten. Slik kan de begrensede ressursene man har fokuseres på prosjektets kjerneproblestillinger, og minst mulig ressurser brukes på å områ seg og definere oppgaver.

11.6 Gradering

Dagens lave tilgjengelighet på informasjon er et sikkerhetstiltak i seg selv, om enn ikke et bevisst tiltak. "Security by obscurity" vanskeliggjør tilgang til informasjon på godt og vondt. Ved å samle og systematisere informasjon og visualisere sammenhenger på en lettfattelig måte kan det bli behov for å gradere modellene. Dermed kan de virke mot sin hensikt. En del av den informasjonen som er ønskelig å inkludere i modeller kan også fra før være gradert. Spørsmålet om gradering av modeller er uavklart, og kan vise seg å forsinke eller forhindre etablering av en integrert modell for Forsvaret som helhet.

11.7 Felles forum for systemarkitektur i Norge

Både FFI og Forsvaret har faglige forum som diskuterer hvordan man skal arbeide med systemarkitektur i Norge. Deltakere i disse to fora har en sporadisk kommunikasjon idag. Det er svært viktig å etablere god kommunikasjon mellom de som driver med modellering allerede nå, slik at flest mulig erfaringer blir representert i prosessen med å ta frem felles løsninger. Med et levende norsk fagmiljø rundt systemarkitektur vil vi få god oppslutning om de valgte løsningene som følge av involvering og god tid til å begynne tilpasning.

"Arkitekturforum" i Forsvaret har en formell rolle i forbindelse med godkjenning av frem-skaffelsesløsninger, foruten å være et forum for faglig utvikling. På FFI er det flere grupper under kompetanseforvaltningen som er aktuelle. Foreløpig er Kompetansegruppen for Modellering og

Simulering mest aktiv i forbindelse med systemarkitektur. Mulighetene for et felles arkitekturforum som diskuterer implementasjon av NAF v3 bør undersøkes videre.

11.8 Implementasjonsplan og opplæring

Det er i dag et fåtall personer i det norske forsvarsrelaterte miljøet som har kompetanse på UML-modellering iht NAF. Et beskjedent antall (ca 20-30?) personer bør få opplæring i NAF v3 og bruk av et aktuelt modelleringsverktøy, eksempelvis Enterprise Architect eller System Architect. Disse personene vil dermed kunne fungere som ”arkitekter” i Forsvaret.

Håndbøker, prosedyrer og maler må utarbeides og gjøres tilgjengelige så snart som mulig for å redusere tiden det tar å innføre den nye metodikken. En implementasjonsplan for NAF v3 i Norge bør utarbeides.

Forkortelser

CAS – Close Air Support

C4ISR – Command, Control, Communications, Computers, Intelligence, Surveillance, Reconnaissance

DoDAF – Department of Defense Architecture Framework

EA – Enterprise Architect

FAC – Forward Air Controller

FD – Forsvarsdepartementet

FK KKIS – Forsvarets kompetansesenter for kommando- og kontroll informasjonssystemer

FLO – Forsvarets logistikkorganisasjon

IFEAD – Institute for Enterprise Architecture Developments

INCOSE – International Council on Systems Engineering

INI – Informasjonsinfrastruktur

KKIS – Kommando- og kontroll informasjonssystemer

MODAF – Ministry of Defence Architecture Framework

M3 – MODAF Meta Model

NAF – NATO Architecture Framework

NAV – NATO All View

NCV – NATO Capability View

NOV – NATO Operational View

NPV – NATO Program View

NSV – NATO System View

NSOV – NATO Service Oriented View

NTV – NATO Technical View

SySML – Systems Modeling Language

TRADOK – Hærens transformasjons- og doktrinekommando

UML – Unified Modeling Language

UPDM – UML Profile for DoDAF and MODAF

Referanser

- [1] M. Østbø, "Nye Kampfly - fellesoperativ og flernasjonalt interoperabilitet (Unntatt Offentlighet)," FFI/RAPPORT-2005/03321, 2005.
- [2] INCOSE, *Systems Engineering Handbook*, 2a ed 2004.
- [3] "Ministry of Defence Architectural Framework (MODAF) www.modaf.org.uk," 2007.
- [4] M. e. Pidd, *Systems Modelling, Theory and Practice* John Wiley & Sons, Ltd., 2004.
- [5] T. Williams, *Modeling Complex Projects* 2002.
- [6] "NATO C3 Systems Architecture Framework (NAF) version 2, ANNEX 1(AC/322-D(2004)0041)," 2004.
- [7] "DoD Architecture Framework, version 1.5," 2007.
- [8] "Unified Modeling Language, version 2.0," 2007.
- [9] "<http://syseng.omg.org/UPDM.htm>," 2007.
- [10] "IFEAD www.enterprise-architecture.info," 2007.
- [11] "www.sparxsystems.com," 2007.