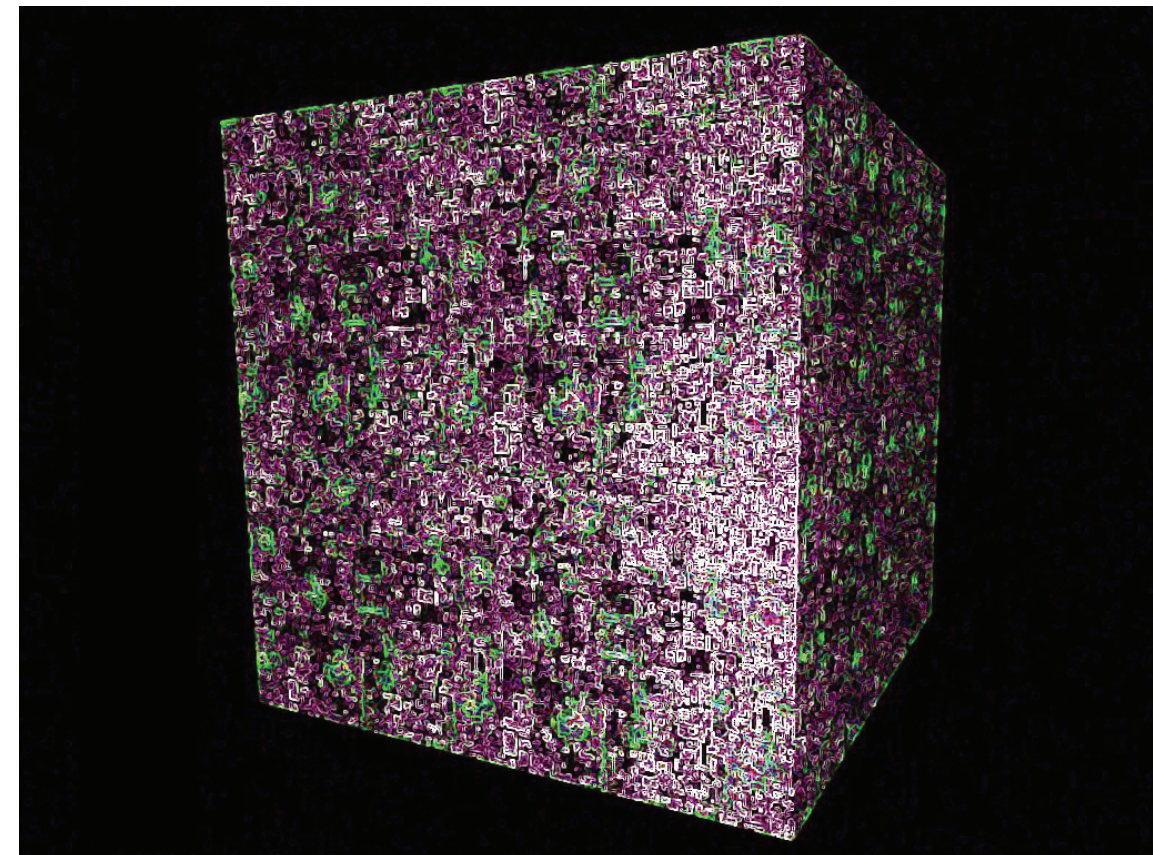




## Verktögsstöd för MBFU ur ett M&S-perspektiv

VAHID MOJTAHED, ULRİK FRANKE, MIKA COHEN,  
LINUS LUOTSINEN, HIRAD ASADI, MARTIN EKLÖF



FOI är en huvudsakligen uppdragsfinansierad myndighet under Försvarsdepartementet. Kärnverksamheten är forskning, metod- och teknikutveckling till nytta för försvar och säkerhet. Organisationen har cirka 1000 anställda varav ungefär 800 är forskare. Detta gör organisationen till Sveriges största forskningsinstitut. FOI ger kunderna tillgång till ledande expertis inom ett stort antal tillämpningsområden såsom säkerhetspolitiska studier och analyser inom försvar och säkerhet, bedömning av olika typer av hot, system för ledning och hantering av kriser, skydd mot och hantering av farliga ämnen, IT-säkerhet och nya sensorers möjligheter.



FOI  
Totalförsvarets forskningsinstitut  
Informationssystem  
Box 1165  
581 11 Linköping

Tel: 013-37 80 00  
Fax: 013-37 81 00

[www.foi.se](http://www.foi.se)

FOI-R--3306--SE  
ISSN 1650-1942

Underlagsrapport  
December 2011

**Informationssystem**

Verktögsstöd för MBFU  
ur ett M&S-perspektiv



Vahid Mojtahead, Ulrik Franke, Mika Cohen, Linus Luotsinen, Hiran Asadi,  
Martin Eklöf

## Verktögsstöd för MBFU ur ett M&S-perspektiv





<b>Titel</b>	Verktygsstöd för MBFU ur ett M&S-perspektiv
<b>Title</b>	Tool support for Model-based Capability Development
<b>Rapportnummer / Report no</b>	FOI-R--3306--SE
<b>Rapporttyp / Report type</b>	Underlagsrapport / Base data report
<b>Utgivningsår / Year</b>	2011
<b>Antal sidor / Pages</b>	102
<b>Kund / Customer</b>	FOI
<b>Forskningsområde</b>	2. Operationsanalys, modellering och simulering
<b>Research area</b>	2. Operational Research, Modelling and Simulation
<b>Delområde</b>	21. Modellering och simulering
<b>Sub area code</b>	21. Modelling and Simulation
<b>Projektnummer / Project no</b>	E53330
<b>Godkänd av / Approved by</b>	Magnus Jändel
<b>ISSN</b>	ISSN-1650-1942

FOI Totalförsvarets forskningsinstitut  
Informationssystem  
Box 1165  
581 11 LINKÖPING



## Sammanfattning

Verksamheter kan ses som komplexa socio-tekniska system där ett antal sammankopplade, och förhoppningsvis välfungerande, komponenter interagerar med varandra för att uppnå verksamhetens strategiska målsättning. Kombinationen av personal (med olika bakgrund, kultur och erfarenhet) med diverse tekniska system och processer gör det svårt för verksamhetens ledning att identifiera vilka kritiska faktorer som påverkar verksamhetens strategiska målsättning. För att kunna beskriva, se samband och även analysera en komplex verksamhet från ett holistiskt perspektiv använder man idag verksamhetsarkitekturer (Enterprise Architecture – EA). En EA fångar ”vad som händer” i en organisation, det vill säga hur organisationens aktiviteter, processer, funktioner, system, informationsresurser och andra komponenter hänger ihop och kan relateras till organisationens uppdrag och mål.

Även Försvarsmakten har fattat ett principbeslut avseende användande av *verksamhetsarkitektur* (EA), där man har bestämt sig för att MODAF (Ministry of Defence Architecture Framework), ska användas för all utvecklingsverksamhet. Oaktat MODAF, kommer ett *modellbaserat angreppssätt* för Försvarsmaktens förmågeutvecklingsarbete att ligga i fokus under de kommande åren. Införandet av ett modellbaserat angreppssätt sker emellertid inte momentant. Även om ramverk som MODAF potentiellt kan bidra till att lösa ut ett flertal av de behov som finns kopplat till förmågeutveckling, krävs i sammanhanget även utbildning, metoder, processer, verktyg, etc. Det finns exempelvis behov av stöd vad gäller framtagning av modeller, värdering av olika lösningar, identifiering av konsekvenser av handlingsalternativ, synliggörande av beroenden, med mera. Denna rapport visar på ett antal bidrag, primärt från området modellering och simulering (M&S), som kan tillfredsställa vissa av dessa behov.

*M&S-stöd för MBFU* är ett FoT-projekt som har det övergripande målet att inom ramen för *modellbaserad förmågeutveckling* skapa modellerings- och simuleringsbaserade metoder och verktyg som kan stödja Försvarsmaktens förmågeutvecklingsarbete. För att kartlägga existerande behov startade projektet med att genomföra en behovsanalys som gav en bra inblick och förståelse kring behov kopplat till både arkitekturutveckling och arkitekturanvändning inom Försvarsmakten. Därefter initierades en omvärldsskanning för att belysa ett antal område som kunde bidra till analys- och beslutsstöd inom MBFU, vilket har resulterat i föreliggande rapport. Rapporten ämnar därmed beskriva state-of-the-art inom ett antal för MBFU relevanta områden såsom semantiska tekniker, agent-teknologier, analysmetoder, simuleringsverktyg, planering med mera. En utgångspunkt för rapporten är att EA har en stor potential att användas, inte bara för beskrivning och kommunikation, utan även för analys. För att hjälpa den icke EA-kunnige läsaren innehåller rapporten även en övergripande beskrivning av EA, dess användning och nytta, och beskriver även översiktligt ett antal relevanta verktyg som är tillgängliga på marknaden.

## Nyckelord

Domänspecifik modellering, Modellering och simulering, Simuleringsramverk, Enterprisesimulering, Analysmetod, Beslutsstöd, Arkitekturanalys, Planeringsmetod, Konsistenskontroll, Modellbaserad förmågeutveckling, Exekverbar MODAF, MBFU, MODAF, EA, EA-verktyg.



## Abstract

An enterprise can be viewed as a complex, sociotechnical system in which a number of interconnected components interact in order to realise the strategic goal of the enterprise. The combination of personnel (of various background, culture and experience) with technical systems and processes makes it difficult for the enterprise management to identify what critical factors affect the strategic goal of the enterprise. Today Enterprise Architecture (EA) is used to describe and analyse a complex enterprise from a holistic perspective. An enterprise architecture captures “what happens” in an organisation, i.e., how activities, processes, functions, systems, information resources and other components combine and relate to the goal and purpose of the enterprise.

The Swedish Armed Forces have decided to use EA, and, more specifically, MODAF (Ministry of Defence Architecture Framework), throughout its capability development. Irrespective of MODAF, a model based approach will be central to the capability development during the coming years. The introduction of a model based approach, however, does not occur over night. Even if EA frameworks such as MODAF has the potential to considerably improve capability development as such, training, methods, processes, tools, etc. are also needed. This report identifies a number of technologies, primarily taken from the Modelling and Simulation (M&S) field, which support an enterprise architect in constructing, evaluating and visualising models.

M&S support for model based capability development (“M&S-stöd för MBFU”) is a research and technology (“FoT”) project with the purpose of developing M&S-based methods and tools that can assist the capability development within the MBFU. In order to identify existing needs, the project started with a requirement analysis that gave insight into the needs associated with both the use and development of architecture within the Swedish Armed Forces. Subsequently, the project performed a state-of-the-art survey in a number of selected subject areas that could contribute to analysis and decision support within MBFU; this report is the result of the survey. The report describes state-of-the-art in a number of areas with strong relevance to MBFU, including semantic technologies, agent technologies, EA-analysis, simulation tools, and planning. The starting point for the report is the belief that EA has the potential to be used not only for description and communication but also for analysis. To help the reader not familiar with EA, the report includes an introduction to EA with a summary of commercially available EA tools.

## Keywords

Domain specific modeling, Modeling and simulation, Simulation frameworks, Enterprise simulation, EA-analysis, Decision support, Planning, Model checking, Model based capability development, Executable MODAF, MBFU, MODAF, EA, EA-tools.

# Innehåll

<b>Förkortningslista</b>	<b>13</b>
<b>1 Inledning</b>	<b>15</b>
1.1 Bakgrund . . . . .	15
1.2 Projektet M&S-MBFU . . . . .	15
1.3 Frågeställningar . . . . .	16
1.4 Genomförandeplan . . . . .	17
1.5 Rapportens syfte . . . . .	18
1.6 Läsanvisning . . . . .	19
<b>2 Enterprise Architecture – EA</b>	<b>21</b>
2.1 Introduktion . . . . .	21
2.2 Vad är EA? . . . . .	21
2.3 Vad är nyttan med EA? . . . . .	23
2.4 Arkitekturramverk . . . . .	24
2.4.1 MODAF (DoDAF & NAF) . . . . .	25
2.4.2 Zachman . . . . .	28
2.4.3 TOGAF . . . . .	30
2.4.4 Användning av TOGAF och NAF tillsammans . . . . .	31
2.5 Sammanfattning . . . . .	32
<b>3 EA-verktyg – en översikt</b>	<b>35</b>
3.1 Alfabet . . . . .	35
3.2 Mega . . . . .	37
3.3 Troux Technologies . . . . .	38
3.4 IBM . . . . .	38
3.5 Metastorm . . . . .	39
3.6 Software AG . . . . .	40
3.7 Avolution . . . . .	41
3.8 QualiWare . . . . .	41
3.9 Sammanfattning . . . . .	42

<b>4</b>	<b>Krafftfulla simuleringsverktyg för MBFU</b>	<b>43</b>
4.1	Introduktion . . . . .	43
4.2	Vad är ett simuleringsramverk? . . . . .	44
4.3	Nyttan med simuleringsramverk . . . . .	45
4.4	Några exempel på simuleringsramverk . . . . .	46
4.4.1	Flames från Ternion . . . . .	47
4.4.2	Strive, Aeria, Stage och AI-implant från CAE . . . . .	48
4.4.3	MASA Sword och MASA Life från MASA Group . . . . .	51
4.4.4	VR-Forces från MÄK . . . . .	53
4.5	Sammanfattning . . . . .	53
<b>5</b>	<b>Modelleringsmetod för exekverbar MODAF</b>	<b>57</b>
5.1	Introduktion . . . . .	57
5.2	Domänspecifik modellering (DSM) . . . . .	57
5.3	Automatiska analysmetoder . . . . .	60
5.3.1	Ontologi och automatisk inferens . . . . .	61
5.3.2	Model checking: automatisk tillståndsrymdssökning . . . . .	62
5.3.3	Automatisk planering . . . . .	62
5.4	Domänspecifik MODAF . . . . .	63
5.4.1	MODAF-profiler . . . . .	64
5.4.2	Arbetsdelning mellan MODAF och MODAF-profiler . . . . .	65
5.4.3	MODAF-profiler för verksamhetsmodellering . . . . .	66
5.5	Sammanfattning . . . . .	67
<b>6</b>	<b>Analysmetoder för MBFU</b>	<b>69</b>
6.1	Introduktion . . . . .	69
6.2	Bakgrund . . . . .	69
6.3	Arkitektur vs. arkitektur . . . . .	69
6.3.1	Arkitekturanalys . . . . .	71
6.4	Vikten av mål . . . . .	71
6.5	Behovet av att hantera osäkerhet . . . . .	72
6.6	Formalismer för analys . . . . .	73
6.6.1	Probabilistiska relationsmodeller . . . . .	73
6.7	Tillämpningsområden . . . . .	73

6.7.1	Analys av systemegenskaper . . . . .	74
6.7.2	Analys av militära förmågor . . . . .	74
6.8	Verktögsstöd . . . . .	74
6.9	Sammanfattning . . . . .	75
<b>7</b>	<b>Verksamhetssimuleringsmetoder för MBFU</b>	<b>77</b>
7.1	Introduktion . . . . .	77
7.2	Simuleringsmetoder . . . . .	78
7.2.1	Diskret händelsestyrd simulering . . . . .	79
7.2.2	Systemdynamik . . . . .	82
7.2.3	Agent-baserad modellering och simulering . . . . .	84
7.3	Sammanfattning . . . . .	85
<b>8</b>	<b>Effektiva planeringsmetoder för MBFU</b>	<b>87</b>
8.1	Introduktion . . . . .	87
8.2	Förmågor, operationsplanering och arkitekturbeskrivningar . . . . .	87
8.3	Algoritmer . . . . .	89
8.4	Sammanfattning . . . . .	90
<b>9</b>	<b>Summering</b>	<b>91</b>
9.1	Modellbaserad förmågeutveckling . . . . .	91
9.2	Metoder för kvalificerat beslutsstöd . . . . .	91
	<b>Litteraturförteckning</b>	<b>93</b>



## Förkortningslista

AB	Agent-baserad modellering
ADM	Architecture Development Method
BPM	Business Process Modelling
C2	Command & Control
CGF	Computer Generated Forces
CLD	Causal Loop Diagram
COTS	Commercial-off-the-shelf
CPN	Colored Petri Net
DHS	Diskret händelsestyrd simulering
DoDAF	Department of Defense Architecture Framework
DSM	Domain Specific Modeling
EA	Enterprise Architecture
EA <sup>2</sup> T	Enterprise Architecture Analysis Tool
EBAO	Effects Based Approach to Operations
EC	Enterprise Continuum
ES	Enterprise Simulation
FEAF	Federal Enterprise Architecture Framework
FMEA	Failure Modes and Effects Analysis
M3	MODAF Meta Model
M&S	Modellering och simulering
MBFU	Modellbaserad förmågeutveckling
MODAF	Ministry of Defence Architecture Framework
NAF	NATO Architecture Framework
OCL	Object Constraint Language
OMG	Object Management Group

P<sup>2</sup>AMF Predictive, Probabilistic Architecture Modeling Framework

PRM Probabilistic Relational Model

ROI Return On Investment

SD Systemdynamik

SFD Stocks and Flows Diagram

SRD System Requirements Document

TOGAF The Open Group Architecture Framework

UML Unified Modeling Language

URD User Requirements Document

# 1 Inledning

## 1.1 Bakgrund

Försvarsmakten strävar efter att införa ett modellbaserat angreppssätt vad gäller utveckling av förmågor – Modellbaserad förmågeutveckling (MBFU). Detta förväntas bidra till ökad spårbarhet i förmågeutvecklingens samtliga faser, samt i det långa loppet ge bättre förutsättningar för välgrundade beslut genom en tydligare kravställning med högre kvalitet, jämfört med det idag rådande dokumentbaserade angreppssättet.

FM har tagit ett principbeslut avseende användande av verksamhetsarkitektur (Enterprise Architecture - EA), närmare bestämt MODAF (Ministry of Defence Architecture Framework), för all utvecklingsverksamhet. Oaktat MODAF, kommer med stor sannolikhet EA och utveckling utifrån ett modellbaserat angreppssätt att vara tongivande de kommande åren. För att dra full nytta av den potential som EA och modellbaserad utveckling har, måste befintliga ramverk kompletteras med metod, teknik och verktyg som gör dem användbara i en FM-kontext.

Mot denna bakgrund behöver Försvarsmakten stöd i sitt arbete att smidigt övergå från ett dokumentbaserat till ett modellbaserat angreppssätt för förmågeutveckling. MODAF är "endast" ett ramverk och som sådant ger det enbart regler och riktlinjer för hur information ska struktureras/ordnas. MODAF är just inriktat på att beskriva strukturen för artefakter (modeller), snarare än att som exempelvis TOGAF även fokusera på den process/metod som leder fram till en arkitektur. Inom förmågeutvecklingsprocessen, exempelvis i samband med målsättningsarbete, finns behov av stöd vad gäller framtagning av modeller, värdering av olika lösningar, identifiering av konsekvenser av handlingsalternativ, synliggörande av beroenden, med mera. Genom att använda MODAF och tillämpa ett modellbaserat angreppssätt tillgodoses dessa behov till en viss nivå, men ytterligare verktyg och metoder behövs för att utveckla förmågor som uppfyller de krav som ställs på Försvarsmakten.

## 1.2 Projektet M&S-MBFU

M&S-MBFU är ett nystartat FoT-projekt inom området Modellering och Simulering (M&S), som har för avsikt att utnyttja erfarenhet och kompetens, främst från två tidigare FoT-projekt, nämligen DCMF (Defence Conceptual Model Framework), samt InfoM&S (Effektiv informationshantering av M&S-resurser i Ledningssystem), i ett MBFU-sammanhang. Projektet skall vidare undersöka hur metoder och tekniker från M&S-området kan användas i kontexten modellbaserad utveckling. Projektets övergripande mål är att inom



ramen för modellbaserad förmågeutveckling skapa M&S-baserade metoder och verktyg som kan stödja Förvarsmaktens (förmåge-)utvecklingsarbete. Syftet är att stödja utvecklare och användare i förmågeutvecklingsprocessen genom att underlätta framtagning, utvärdering och användning av modeller.

Målet på sikt är, genom att utveckla och kombinera modellerings- och simulerings-baserade metoder, att bidra till att Förvarsmakten enklare och snabbare kan producera och utnyttja sina arkitekturmodeller. De modeller som tas fram ska vara av hög kvalitet och på ett effektivt sätt bidra till styrning och inriktning av FMs verksamhet.

Nyttan som projektet ämnar tillföra FM är därmed effektivitets- och kvalitetshöjning i utvecklingsverksamheten genom M&S-stöd. Mottagargruppen för projektets resultat är dock bredare än ett enskilt projekt eller en individ inom FM. Projektet ska snarare bistå med M&S-stöd dels till de projekt och individer som jobbar med arkitekturutveckling genom att bidra till modeller av hög kvalitet och dels med M&S-stöd till de projekt och individer som jobbar med arkitektur användning genom att bidra till ökad nytta av en arkitektur. Därmed avser projektet stödja både utvecklare och användare i förmågeutvecklingsprocessen genom att underlätta framtagning, utvärdering och användning av modeller.

### 1.3 Frågeställningar

De för projektet mest relevanta frågeställningarna kan ses från två olika perspektiv: dels M&S-stöd till arkitekturutveckling som behandlar aspekter som förväntas bidra till modeller av högre kvalitet och dels M&S-stöd till arkitektur användning som förväntas bidra till ökad nytta av en arkitektur.

När det gäller frågeställningar kopplat till arkitekturutveckling har projektet identifierat ett antal intressanta områden som kan komma att ligga i fokus för metod- och verktygsutveckling. En fundamental egenskap som ofta omnämns i MBFU-sammanhanget är en modells återanvändbarhet. Modeller som utvecklas i skilda sammanhang ska kunna återanvändas för att bygga modeller som möter andra och/eller nya behov. Givet detta är det av intresse att se hur en modell kan verifieras (automatiskt), dels "internt", det vill säga att den följer uppsatta (design)"regler", dels "externt", det vill säga att den bidrar till att skapa en sammanhängande och giltig modell i en konfiguration av flera modeller (skapad för ett nytt syfte). Vi kallar detta för verifiering och konsistenskontroll.

Ett annat område är simulering. Under ett arkitekturarbete kan simulering vara ett viktigt hjälpmedel för att närmare studera komplexa skeenden och frågeställningar. Resultatet från en simulering kan ge värdefull feedback till utvecklingen av en arkitektur. Potentiellt kan simulering även användas som ett medel för att validera de krav som uttrycks i en modell. Här är det av intresse att främst studera hur simuleringsmiljöer för "Rapid Prototyping" kan användas i anslutning till utveckling av en arkitektur, det vill säga att under

förhållandevis korta ledtider, utifrån den kunskapsmängd som en arkitektur uttrycker, sätta samman en meningsfull simuleringsmodell som kan ge värdefull förståelse av det "system" som är i fokus.

När det gäller frågeställningar kopplat till arkitekturanvändning har bland annat beslutsstöd identifierats som ett centralt område. Syftet med uppbyggnaden av en verksamhetsarkitektur är att den ska bidra till dokumentation, kommunikation, analys och design men även styrning och inriktning av en verksamhet. De artefakter som tas fram inom ramen för ett arkitekturarbete är dock inte alltid lätta att ta till sig och de verktyg inom vilka dessa modelleras erbjuder eventuellt inte de former av analys/presentation av data som är intressant givet en specifik roll. Här är det främst två aspekter som är intressanta att belysa ur projektets perspektiv. Det ena är: givet en formell representation av modeller, vilka nya former av analys medger detta? Om förutsättningarna ändras i en del av modellen, vilka följd effekter får det för andra delar av modellen (exempelvis inverkan på en förmåga givet förändringar i en resurs som bidrar till förmågan). Det andra är: givet koppling av arkitektur och simuleringsramverk (såsom nämns i föregående punkt), vilka möjligheter till analys kan detta bidra till? Kan det användas för att på ett "enkelt" sätt utföra "what-if"-analys givet den verksamhet som en arkitektur representerar?

## 1.4 Genomförandeplan

För att kartlägga existerande behov av M&S-stöd inom MBFU-verksamhet identifierade projektet ett antal tillämpningsområde och inom dessa områden identifierades och kontaktades olika typer av användare. Utöver Forsvarsmakten så identifierades specifika individer och verksamheter även inom FMV, FOI och industrin som utför konkreta modelleringsarbeten på uppdrag av FM och FMV. Med dessa individer och verksamheter som bas genomfördes under första halvåret 2011 en krav- och behovsanalys som gav en bra inblick och förståelse kring behov kopplat till både arkitekturutveckling och arkitekturanvändning inom FM. Behovsanalysen genomfördes förhållandevis brett och täckte in intresser från ovannämnda organisationer. Resultatet har dokumenterats i [1].

En omvärldsskanning av för MBFU relevanta områden initierades under första halvåret vilket har legat i fokus för arbetet under hösten 2011. Omvärldsanalysen, föreliggande rapport, förväntas ge input kring state-of-the-art inom forskning relaterat till skärningen mellan MBFU och ett antal relevanta områden såsom simulering, semantiska tekniker, formella metoder, agent-teknologier, etc. Baserad på dessa behovs- och "state of the art"-analyser skall mot slutet av 2011 ett vägval göras, en fokusering föreslås och en utvecklingsplan för fortsatt arbete framläggas.

Under 2012 förväntas detta vägval och fokusering ligga till grund för metod- och verktygsutveckling. Användartester ska genomföras och efterföljande utvärdering ska ligga till grund för den fortsatta projektinriktningen.

Genomförande av enklare prototyp-demonstrationer för både användare och uppdragsgivare planeras också äga rum mot slutet av 2012. Därefter ska ytterligare användartester genomföras och utvärderas så att vidareutveckling och förfining av metoder och verktyg kan utföras. Slutligen ska (första versioner av) framtagna metoder och verktyg testas in i en utvald verksamhet innan projektet avslutas i slutet av 2013.

## 1.5 Rapportens syfte

En verksamhetsarkitektur (Enterprise Architecture – EA) tillåter beskrivning och analys av en verksamhet från ett holistiskt (helhets-) perspektiv. Det finns idag en rad olika EA-ramverk tillgängliga, både inom den civila och militära världen. Försvarsmakten har dock beslutat att vid förbands- och materielutveckling ska MODAF användas. Införandet av ett modellbaserat angreppssätt är emellertid inte något som sker momentant. Även om ramverk som MODAF potentiellt kan bidra till att lösa ut ett flertal av de behov som finns kopplat till förmågeutveckling, krävs i sammanhanget även utbildning, metoder, processer, verktyg, etc. Rapporten kommer att visa på ett antal bidrag detta arbete kan få primärt från området modellering och simulering (M&S).

För läsare med liten eller ingen erfarenhet av modellering med EA-ramverk kan det vara av intresse att förstå vilka komponenter som vanligtvis används vid framtagning av verksamhetsmodeller samt hur dessa är sammankopplade.

En verksamhetsmodell beskrivs med ett EA-ramverk som i sin tur består av ett antal delkomponenter. Notera att EA-ramverk vanligtvis inte tillhandahåller alla dessa komponenter till fullo. Vissa EA ramverk fokuserar på att beskriva verksamheter med s.k. taxonomier (t.ex. Zachman-ramverket) medan andra EA-ramverk endast tillhandahåller processer för hur en verksamhet bör gå till väga för att utveckla en verksamhetsmodell (t.ex. TOGAF). Utöver taxonomier och processer så finns ramverk som tillhandahåller meta-modeller (t.ex. MODAF) vilka representerar abstrakta beskrivningar av de modeller som kan skapas i ramverket (dessa tre ramverk beskrivs utförligare i kapitel 2). Modeller visualiseras med vyer och perspektiv, där perspektiv representerar ett antal vyer. Slutligen, och kanske viktigast av allt, är verksamhetsmodellens användningsområden där den vanligaste är att kommunicera verksamhetens struktur och beteende till beslutsfattare och andra intressenter för analys och beslutsstödsändamål [2], [3].

Rapportens syfte är att göra en omvärldsanalys med fokus på området modellering och simulering för att belysa ett antal område som just kan bidra till analys- och beslutsstöd inom MBFU ur ett M&S-perspektiv. Rapporten har därmed för avsikt att beskriva state-of-the-art inom ett antal relevanta områden såsom semantiska tekniker, agent-teknologier, analysmetoder, simuleringsverktyg, planering med mera och deras skärningar med MBFU. Observera att rapporten inte försöker utvärdera olika metoder och verktyg, utan syftet snarare är att ge översikt av vad som erbjuds på

forskningsfronten, samt hur det relateras till MBFU/MODAF.

## 1.6 Läsanvisning

Till och börja med vill vi utfärda en rekommendation till läsare som känner sig väl förtrogna med EA-världen eller har erfarenhet av modellering med EA-ramverk att vederbörande kan hoppa över kapitel 2 och 3 och börja direkt med kapitel 4, då dessa två kapitel är avsedda som introduktion för läsare med liten eller ingen erfarenhet av EA-området. Därmed projektets egentliga bidrag till MBFU-arbetet börjar först från kapitel 4.

Kapitel 2 börjar med att ge en övergripande beskrivning av EA, dess användning och nytta. Här finns också en övergripande beskrivning av MODAF (inklusive DoDAF och NAF). Slutligen beskrivs även två ramverk från den civila världen översiktligt: Zachman och TOGAF, två av de mest använda och spridda ramverken på marknaden idag.

Kapitel 3 introducerar och beskriver kort ett antal relevanta verktyg som är tillgängliga på marknaden inom området arkitekturramverk.

Kapitel 4 introducerar begreppet simuleringsramverk och diskuterar simulering som ett viktigt komplement till traditionellt arkitekturarbete för att närmare studera komplexa skeenden och frågeställningar. Resultatet från en simulering kan bland annat ge värdefull feedback till utvecklingen av själva arkitekturen.

Kapitel 5 introducerar domänspecifik modellering som ett sätt att hantera komplexitet genom att bygga små, avgränsade och specifika modeller. Sådana modeller har rönt en hel del uppmärksamhet på senare år, eftersom de tenderar att bli mer begripliga och användbara än modeller som görs med mer generella beskrivningspråk.

Kapitel 6 förklarar hur traditionellt arkitekturarbete sällan utnyttjar den fulla potentialen i organisationsövergripande arkitekturbeskrivningar. Med lämpliga metoder kan man dock gå från att mest ta fram *as-is*-beskrivningar till att fullt ut använda sina arkitekturbeskrivningar som indata till analysmodeller som kan stödja kvalificerat beslutsfattande.

Kapitel 7 introducerar verksamhetssimulering (*Enterprise Simulation* – ES) som ett komplement till dagens EA-ramverk. Medan EA-ramverk ofta stannar vid verksameters statiska egenskaper och uppbyggnad kan man med ES modellera även emergenta egenskaper som dyker upp först i ”run time”.

Kapitel 8 förklarar hur arkitekturmodeller kan användas som stöd för planering av militära operationer ur en effektbaserad synvinkel, *Effects Based Approach to Operations* (EBAO). Genom att använda sig av delkomponenterna i en MODAF-modell som beskriver en förmåga så kan man utnyttja detta för att sätta samman mer effektiva förmågor. Tanken är att sätta samman delkomponenter på ett optimalt sätt utifrån arkitekturbeskrivningarna och en effektiv algoritm.

Slutligen sammanfattar kapitel 9 rapporten genom att ge en summering av

FOI-R--3306--SE

resultaten.

## 2 Enterprise Architecture – EA

### 2.1 Introduktion

Under 2008 beslutade Försvarsmakten att modellbaserad utveckling skulle tillämpas för såväl förbands- som materielutveckling. Ansatsen går ofta under benämningen Modellbaserad Förmågeutveckling – MBFU. Den komplexitet som Försvarsmakten som organisation uppvisar, tillsammans med den samverkan som sker med olika partners i internationella sammanhang, kräver en övergång från det rådande dokumentbaserade arbetssättet vid förbands- och materielutveckling till ett, genom modeller, mer strukturerat angreppssätt. I modellbaserad utveckling ges modeller en central och styrande roll. En av fördelarna med detta är ökad spårbarhet, vilket är ett eftersatt behov inom Försvarsmakten idag. Vad som eftersöks är möjlighet till spårbarhet (att kunna visualisera, analysera och förstå beroenden), från krav på Försvarsmakten från regering och riksdag, ned till krav på förmågor, förband och materiel. Exempelvis är det önskvärt att kunna härleda vilka system, i en specifik konfiguration, som bidrar till en viss förmåga. Vidare förväntas det modellbaserade angreppssättet ge förutsättningar för en helhetssyn vad gäller all den utveckling som pågår inom Försvarsmakten, i syfte att undvika utveckling av solitärer, utan hänsyn tagen till relaterade system och ytterst de förmågor som Försvarsmakten förväntas upprätthålla. Detta gäller såväl befintliga som framtida förmågor. Sammantaget förväntas modellbaserad förmågeutveckling bidra till mer välgrundade beslut i utvecklingssammanhang, med hänsyn tagen till helheten, i linje med organisationens mål och vision, utifrån sunda ekonomiska principer.

Vid realisering av modellbaserad förmågeutveckling anses Enterprise Architecture (EA) vara en av grundpelarna. Avsnitt 2.2 och 2.3 ger en övergripande beskrivning av EA, dess användning och nytta. För att nyttja EA på ett systematiskt sätt krävs ett ramverk. Försvarsmakten har beslutat att MODAF (*Ministry of Defence Architecture Framework*) ska användas för detta syfte. Avsnitt 2.4 ger en övergripande beskrivning av MODAF (inklusive amerikanska *Department of Defense Architecture Framework* – DoDAF och NATO:s *NATO Architecture Framework* – NAF), samt två ramverk från den civila världen: Zachman och *The Open Group Architecture Framework* – TOGAF. Zachman och TOGAF är förmodligen de mest använda och spridda ramverken på marknaden idag.

### 2.2 Vad är EA?

EA är en rigorös beskrivning av strukturen hos en verksamhet (*enterprise*). Enterprise kan ha olika innebörd beroende på sammanhang och syfte med

arkitekturen i fråga, exempelvis kan *enterprise* syfta till en hel koncern eller företag, såväl som en del av en koncern eller företag, myndighet, eller en outsourceed verksamhetsprocess. Strukturen hos en verksamhet beskrivs i form av *entiteter*, *relationer* mellan dessa, samt eventuella *attribut* (egenskaper) som entiteter innehar. Beskrivningen av en verksamhet inkluderar hela det socioekonomiska systemet, det vill säga inkluderar beskrivningar av människor, verksamhetsprocesser, teknologi, organisation och information [4]. Det finns självklart flera olika definitioner av EA. Nedan följer några olika definitioner, varav den första är den syn på EA som MODAF representerar.

”The formal description of the structure and function of the components of an enterprise, their inter-relationships, and the principles and guidelines governing their design and evolution over time. (Note: ‘Components of the enterprise’ can be any elements that go to make up the enterprise and can include people, processes and physical structures as well as engineering and information systems.)” (MODAF, [5])

”A strategic information asset base, which defines the business, the information necessary to operate the business, the technologies necessary to support the business operations, and the transitional processes necessary for implementing new technologies in response to the changing business needs. It is a representation or blueprint” (FEAF, [6])

”A rigorous description of the structure of an enterprise, its decomposition into subsystems, the relationships between the subsystems, the relationships with the external environment, the terminology to use, and the guiding principles for the design and evolution of an enterprise” (Giachetti, [7])

”fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution” (IEEE, [8])

TOGAF ställer sig bakom denna formulering, men preciserar att arkitektur kan betyda två olika saker:

1. ”A formal description of a system, or a detailed plan of the system at component level to guide its implementation”
2. ”The structure of components, their inter-relationships, and the principles and guidelines governing their design and evolution over time” (TOGAF, [9])

## 2.3 Vad är nyttan med EA?

För att beskriva vad EA används till och vilken nytta det tillför kan en analogi till stadsplanering göras. En stad utgörs, i det enklaste fallet, av en mängd byggnader, samtliga konstruerade för att uppfylla ett specifikt behov. För varje hus finns konstruktionsritningar av skilda slag och på olika nivåer, från övergripande byggnadsarkitektur ned till beskrivningar av hur husets elektricitet är dragen. Ett hus i en stad kan dock sällan fungera utan relationer till den infrastruktur som finns där. Huset måste försörjas med vatten, el, avlopp etc. för att vara funktionsdugligt. När ett hus byggs måste således hänsyn tas till en lång rad olika faktorer, infrastruktur, men även andra perspektiv som husets användningsområde relativt dess placering. Det är exempelvis sällan lämpligt att placera en kemisk fabrik i anslutning till ett tätbefolkat område, med rekreativmiljöer, dagis etc. Med hjälp av övergripande planering — en stadsplan som anger hur en stad kan/får utvecklas — försöker man både underlätta och styra stadens funktion och utveckling. För utveckling av staden används en lång rad olika planer, ritningar, kartor, modeller etc., som är anpassade till de olika typer av användare/roller som bidrar till detta (i syfte att respektive yrkesman/kvinna ska kunna utföra sitt arbete på bästa möjliga sätt). Samtidigt pekar analogin till stadsplanering också på riskerna med felaktig planering som misslyckas med att ta hänsyn till alla behov och intressenter — ibland till och med på grund av att man försöker *för mycket* (intressant läsning om hur stadsplanering kan misslyckas finns i [10]).

På samma sätt är det inom en verksamhet/organisation inte att rekommendera att enskilda verksamhetsprocesser tillåtas fungera/operera isolerat (som en solitär). Det är ofta mer kostnadseffektivt och rationellt att dela på resurser, utveckla gemensam programvara för funktioner som är likartade inom flera processer, etc. Även om detta är eftersträvsamt kan det i praktiken vara svårt att realisera. Tanken med EA är att det ska kunna bidra till utveckling av en verksamhet med hänsyn tagen till helheten.

En organisation/verksamhet beskrivs genom EA ur ett holistiskt perspektiv, vilket ger förutsättningar för en planering som på lång sikt kan bidra till effektiv styrning och beslutsfattande (*governance*). Exempelvis kan man genom detta tillse att inga utvecklingsbeslut tas utan hänsyn tagen till den kontext som ett system, process etc. (som är i fokus för utvecklingen) är relaterad till. Vidare även att den aktuella utvecklingen stödjer och är i samklang med verksamhetens/organisationens övergripande målsättningar och ytterst dess vision.

Sammantaget anses EA kunna bidra till att bland annat följande uppnås (åtminstone i teorin) [11]:

- Lägre utvecklings-, support-, drift- och förvaltningskostnader
- Förbättrad interoperabilitet



- Ökad förmåga till hantering av kritiska, organisationsövergripande aspekter som exempelvis säkerhet
- Enklare uppgradering och byte av systemkomponenter
- Förbättrad ROI (Return On Investment)
- Flexibilitet som möjliggör outsourcing
- Reducerad risk avseende nya investeringar
- Enklare, snabbare och ytterst, mindre kostsam anskaffning

Det har inte gjorts särskilt många vetenskapliga genomgångar av nyttan med EA. En artikel från 2010 försöker råda bot på detta genom en metastudie av 14 tidigare artiklar [12]. Baserat på denna litteratursammanställning listar man bland annat följande kontexter där EA har visat sig medföra nytta:

- Organisationsdesign inte minst i samband med företagsförvärv och sammanslagningar
- Projektportföljhantering avseende IT och investeringsbeslut
- Beslutsfattande i allmänhet, inklusive *sourcing* och anpassning av COTS-produkter (*Commercial-off-the-shelf*)
- Efterlevnad av lagar och regler inklusive efterlevnad av kvalitetsstandarder
- Systemutveckling med tonvikt på de tidiga faserna
- Riskhantering nästan uteslutande avseende IT-risker
- Minskning av IT-kostnader antingen genom konsolidering eller genom bättre hantering av driftskostnader

## 2.4 Arkitekturramverk

En förutsättning för att kunna arbeta på ett systematiskt sätt med EA och uppnå den nytta som redovisas ovan är ett arkitekturramverk. Ett arkitekturramverk samlar verktyg, tekniker, artefaktbeskrivningar, processmodeller, referensmodeller och instruktioner och används av arkitekter vid utveckling av specifika arkitekturer. Det finns en lång rad arkitekturramverk tillgängliga på marknaden idag. I den civila världen finns ramverk som *The Open Group Architecture Framework – TOGAF* [11] och *Zachman* [13]. Inom den militära domänen finns ramverk som *NATO Architecture Framework – NAF* [14], *Department of Defence Architecture Framework – DoDAF* [15], och *Ministry of Defence Architecture Framework – MODAF* [16].

Ett arkitekturramverk ger den grundläggande struktur som är nödvändig för att utveckla olika typer av arkitekturer med behovsanpassade vyer mot olika intressenter till det ”system” (eller verksamhet) som är i fokus för en analys. Vanligtvis utgörs ett arkitekturramverk av tre huvudsakliga delar [17]:

- Vyer – ger förmågan att kommunicera ”innehållet” av en arkitektur till olika intressenter.
- Metoder – ger förutsättningar för att samla in och organisera data och konstruera vyer på ett sätt som bidrar till korrekt och komplett information, som är sammanhängande ur ett logiskt perspektiv.
- Utbildning/Erfarenhet – stödjer användning av metod och verktyg.

En annan observation som kan göras angående EA-ramverk (och som vi kommer att se i kommande avsnitt) är att ett EA-ramverk ofta delas in i fyra olika typer av arkitekturer, nämligen [17]:

- Verksamhet – beskriver den verksamhet (*business*) som är i fokus för en analys
- Data/information – beskriver den data och information som hanteras och krävs givet den aktuella verksamheten
- Applikation – beskriver de applikationer, som hanterar specificerad information, som krävs för att den aktuella verksamheten ska fungera
- Infrastruktur – beskriver den grundläggande infrastruktur som applikationer, data- och informationshantering och verksamhet kräver

Slutligen förhåller sig de flesta EA-ramverk på ett eller annat sätt till metamodeler, även om långt ifrån alla föreskriver någon specifik sådan. De militära ramverken MODAF, DoDAF och NAF är alla påtagligt metamodeltunga.

#### 2.4.1 MODAF (DoDAF & NAF)

Nedan följer en översiktlig beskrivning av MODAF, vilket är det arkitekturramverk som Försvarmakten har beslutat skall användas vid förbands- och materielutveckling. MODAF utvecklades med DoDAF som grund. Därav finns stora likheter mellan MODAF och DoDAF och den senaste versionen av MODAF, version 1.2, är även kompatibel med DoDAF (MODAF tillhandahåller dock ett antal utökningar, se nedan). NAF är också kompatibelt med MODAF, då dessa två ramverk i princip har identiska meta-modeller. Det pågår också arbete med att skapa en gemensam modelleringsstandard för MODAF och DoDAF i form av *Unified Profile for DoDAF/MODAF* – UPDM [18]. UPDM förväntas även kunna stödja NAF. Givet de likheter som dessa tre ramverk uppvisar och givet att Försvarmakten har beslutat sig för att använda

MODAF, beskrivs primärt MODAF i nedanstående text, men beskrivningen är i stort även giltig för DoDAF och NAF.

Det grundläggande syftet med MODAF är att det skall tillhandahålla ett sätt att beskriva, analysera och styra (*manage*) försvarsinriktade verksamheter. Detta möjliggörs genom ett rigoröst beskrivningsätt för förmågor, system, system av system, organisationsstrukturer och verksamhetsprocesser.

En central del av MODAF är vyer (*views*), vilka delas in i kategorier enligt att antal betraktelseperspektiv (*viewpoints*). Genom dessa betraktelseperspektiv kan MODAF presentera en verksamhet för olika typer av intressenter utifrån en och samma grundmodell.

MODAF tillhandahåller totalt 7 betraktelseperspektiv (se figur 2.1), enligt följande [19]:

**ALL Views** En övergripande beskrivning av en arkitekturmodell med aspekter såsom arkitekturs syfte, avgränsningar och ägandeskap. Vidare beskrivs här även de resultat och slutsatser som framkommit under utveckling av en arkitektur.

**TECHNICAL STANDARDS Views** Referenser till standarder, regler, riktlinjer med mera som är relevanta för en arkitektur. Här handlar det inte enbart om teknisk information, utan även andra informationskällor såsom doktrin och taktik är av intresse.

**STRATEGIC Views** Beskrivning av verksamhetens vision, förmågor, relationer mellan förmågor, tidsperspektiv för förmågor, samt koppling av förmågor till aktivitetsmodeller (modeller som beskriver vad en verksamhet utför). Vyerna ger förutsättningar för analys och optimering av organisationens leverans av förmågor över tiden.

**OPERATIONAL Views** Beskrivning av arkitekturs logiska perspektiv, det vill säga vyerna beskriver hur verksamheten bedrivs i form av uppgifter, aktiviteter, informationsflöde och mycket mer. Förmågor som specificerats i det strategiska perspektivet ges, genom det logiska perspektivet, ett sammanhang. Vyerna på denna nivå beskriver *vad* organisationen vill utföra, men inte i detalj *hur*.

**SYSTEM Views** Beskrivning av resurser som tillsammans bidrar till eftersökta förmågor. System-vyer kan representera en "lösning" till de krav som specificerats i de operationella vyerna, och/eller kan ses som ett sätt att tillföra mer detaljer till de operationella vyerna. Med andra ord används system-vyer till att specificera hur önskvärda förmågor realiserar.

**SERVICE Views** Ger möjlighet till att ta fram en arkitektur utifrån tjänstebaserade principer (SOA). Tjänstevyerna används för att specificera tjänster, hur tjänster tillsammans kan konfigureras för att

möta efterfrågade förmågor, samt på en övergripande nivå hur enskilda tjänster är implementerade.

**ACQUISITION Views** Ger möjlighet till beskrivning av program och projekt som bidrar till realisering av organisationens förmågor.



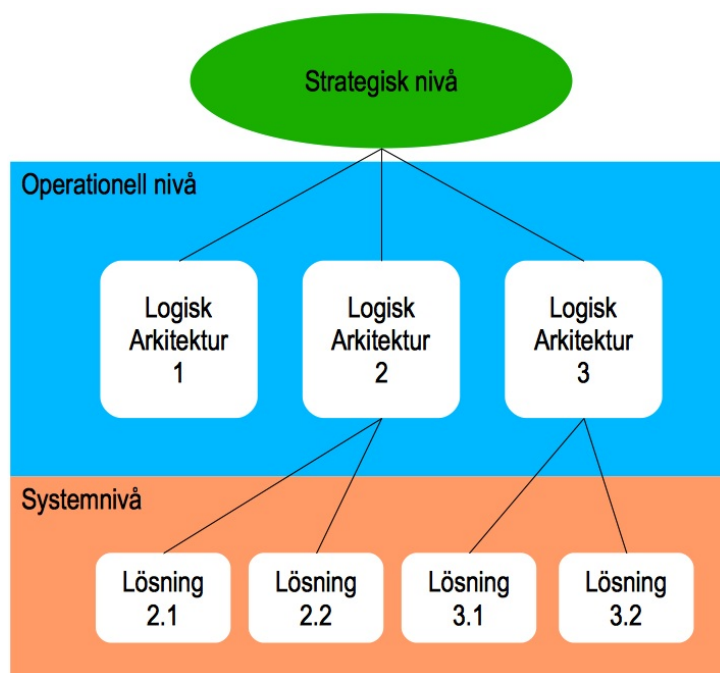
Figur 2.1: MODAFs sju betraktelseperspektiv [19].

I MODAF görs en viktig distinktion mellan de operationella vyerna och systemnivån. Som nämnts tidigare representerar operationsnivån *vad* organisationen förväntas göra, medan systemnivån beskriver *hur* något görs. Genom denna uppdelning är det möjligt att ha flera alternativ på operationsnivån (logiska arkitekturer) som var och en har en eller flera relaterade lösningar (i form av systemvyer). På det sättet kan olika lösningsförslag jämföras i termer av kostnad, kvalitet, interoperabilitet och så vidare, se figur 2.2. Det är därför viktigt att påpeka att specifikation av lösningsförslag ska undvikas på operationsnivån under arkitekturarbetet.

Den strategiska nivån användas normalt för att beskriva behov på hög nivå (förmågor som organisationen vill/bör ha). Operationsnivån beskriver då den funktionalitet som krävs kopplat till verksamhetens processer, medan systemnivån används för att beskriva lösningar. Något förenklat kan operationsnivån liknas vid en URD (*User Requirements Document*) och systemnivån med en SRD (*System Requirements Document*). Det finns dock inga strikta gränsdragningar här och överlapp mellan dessa nivåer är vanligt.

Många av de fördelar som en arkitekturbaserad ansats förväntas ge, bygger på möjligheten att kunna dela, sätta samman, söka och återanvända arkitekturmodeller som beskriver olika delar av en organisation. För att kunna realisera detta behöver den information som bygger upp en modell struktureras på ett standardiserat sätt och modellerna måste kunna utbytas mellan olika verktyg och lagras i ett gemensamt bibliotek. Med ovanstående i åtanke har MODAF Meta Model (M3) tagits fram.

M3 är en referensmodell som definierar den information, eller informationselement, som presenteras i MODAFs vyer. Tanken här att vyer



Figur 2.2: Arkitekturer på logisk- och systemnivå.

representerar en ögonblicksbild av den data som representeras enligt M3 (och som potentiellt lagras i någon form av databas/repository). Genom att samma informationselement potentiellt kan användas i flera olika vyer (där det representerar samma företeelse), och genom att referensmodellen definierar alla koncept och relationer mellan koncept som en vy kan inkludera, kan en arkitektur hållas konsistent över alla vyer som tas fram [20].

*Object Management Group* (OMG) XMI-specifikation används som filformat för MODAF M3. I syfte att kunna återanvända befintliga XMI-gränssnitt som EA-verktyg redan har, är M3 en utökning av UML 2.0. Detta innebär att varje element som M3 definierar utgör en specialisering av en stereotyp i UML [20].

### 2.4.2 Zachman

Zachman är ett arkitekturramverk som utvecklades av John Zachman (på den tiden vid IBM) under 80-talet. Sedan dess har ramverket uppdaterats vid ett flertal tillfällen till den form det har idag [21]. Den senaste versionen, v3.0, publicerades i augusti 2011.

Zachman kan liknas vid en taxonomi snarare än ett "traditionellt" arkitekturramverk. Med det menas att Zachman kan sägas ge ett

klassificeringssystem med vars hjälp arkitekturrelaterade artefakter kan organiseras. Artefakter kan i detta fall utgöras av designdokument, modeller, specifikationer etc. Klassificeringen tar både hänsyn till vilken mottagare som artefakten är utvecklad för (systemägare, utvecklare etc.) och vad den är tänkt att hantera [22].

I sin nuvarande form består ramverket av 36 celler som representerar skärningspunkter mellan sex klassificeringsperspektiv; *What, How, Where, Who, When, Why* och sex intressent-perspektiv; *Executive, Business Management, Architect, Engineer, Technician* och *Enterprise*. En rad i den matris som dessa två perspektiv tillsammans bygger upp beskriver således en och samma verksamhet utifrån samtliga klassificeringsperspektiv, men med samma intressent i åtanke, exempelvis logiska modeller för samtliga klassificeringsperspektiv avsedda för en arkitekt. En kolumn motsvarar således ett klassificeringsperspektiv, exempelvis *How* (som primärt inriktas mot processflöden), utifrån samtliga intressenters perspektiv. Figur 2.3 ger en förenklad bild av Zachman. Cellerna i mitten är de som fylls med innehåll under utvecklingen av en arkitektur.

<i>Classification Names</i>	What	How	Where	Who	When	Why	<i>Model Names</i>
Executive							Scope Contexts
Business Management							Business Concepts
Architect							System Logic
Engineer							Technology Physics
Technician							Tool Components
Enterprise							Operation Instances
<i>Audience Perspective</i>	Inventory Sets	Process Flows	Distribution Networks	Responsibility Assignments	Timing Cycles	Motivation Intentions	<i>Enterprise Names</i>

Figur 2.3: Zachmans ramverk.

Då varje cell har ”populerats” med lämpliga artefakter så finns tillräcklig information för att beskriva en verksamhet eller ett system utifrån samtliga aspekter och utifrån alla intressenters behov av information. Det är dock en subjektiv bedömning att besluta att arkitekturen i fråga är ”färdig”.

En tanke med Zachman är även att det ska finnas relationer mellan individuella celler i matrisen. Exempelvis ska det vara möjligt att ha spårbarhet, från verksamhetskrav ned till databasdesign, för att på så sätt kunna påvisa att designen drivs av verksamhetens behov.

### 2.4.3 TOGAF

The Open Group Architecture Framework – TOGAF är ett ramverk och en uppsättning verktyg för att utveckla EA-modeller. TOGAF ägs, som namnet anger, av *The Open Group*.<sup>1</sup> Utvecklingen av TOGAF påbörjades 1995 baserat på *Technical Architecture Framework for Information Management* (TAFIM), som hade utvecklats under många år av *US Department of Defense* (DoD). TOGAF delar in en arkitektur i fyra kategorier, nämligen [11]:

**Verksamhetsarkitektur** definierar verksamhetens strategi, styrning, organisering, och primära verksamhetsprocesser.

**Applikationsarkitektur** ger en karta över de individuella applikationer/system som används inom en verksamhet, dess interaktioner och dess relation till de primära verksamhetsprocesserna.

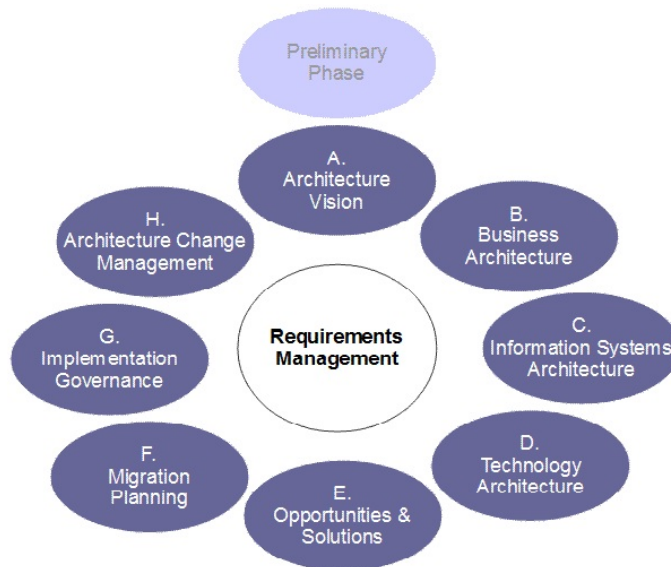
**Dataarkitektur** beskriver strukturen hos verksamhetens logiska och fysiska data- och informationsresurser, såväl som resurser för dess hantering (management).

**Teknologisk arkitektur** beskriver de förmågor/funktioner, på ett logiskt plan, som krävs för att stödja verksamheten, applikationer och data. Detta inkluderar IT-infrastruktur, nätverk, kommunikation, standarder etc.

Även om TOGAF anses vara ett ramverk för EA är kanske dess viktigaste del *Architecture Development Method* (ADM). ADM är en process/metod för att ta fram arkitekturer. Således är TOGAF inte enbart fokuserat på strukturen för artefakter, eller hur dessa ska kategoriseras, utan beskriver främst hur artefakter kan tas fram. ADM består av 8 huvudsakliga faser enligt figur 2.4. Dessa faser är i sin tur indelade i ett flertal delaktiviteter. En detaljerad genomgång av faser och delaktiviteter inom ADM ligger dock bortom syftet med denna rapport (för en detaljerad genomgång hänvisas till [11] och [9]). Observera att detta inte är en strikt sekventiell process, utan ADM är iterativ; över hela processen, mellan enskilda faser och inom specifika faser [22]. ADM är en förhållandevis generell metod och som sådan kan den behöva anpassas utifrån det sammanhang som är aktuellt för en given arkitekturutveckling. ADM kan exempelvis användas för att ta fram artefakter som följer ett annat arkitekturramverks format, såsom MODAF M3.

En annan grundläggande del av TOGAF är det som benämns *Enterprise Continuum* – EC. EC utgörs av en mängd arkitekturer som sträcker sig från det generella till det specifika och som samtliga tas fram med stöd av TOGAF ADM. Inom EC, vars artefakter (bör) hanteras i ett gemensamt bibliotek/repository, finns generella arkitekturer vars principer, i teorin, kan tillämpas inom godtycklig organisation, till arkitekturer som är

<sup>1</sup><http://www3.opengroup.org/>



Figur 2.4: TOGAF:s metod för arkitekturutveckling (ADM).

relevanta inom specifika domäner, ned till organisationsspecifika arkitekturer. ADM tillämpas för att driva fram utveckling från generell abstraktion mot organisationsspecifika arkitekturer [22].

#### 2.4.4 Användning av TOGAF och NAF tillsammans

Som framgått ovan har de olika arkitekturramverken olika fokus: medan TOGAF fokuserar på utvecklingsprocessen för arkitekturbeskrivningar har MODAF (och dess släktingar) en detaljerad metamodel. Det faller sig alltså naturligt att kombinera ramverken. Det norska försvaret har valt TOGAF ADM som sin arkitekturutvecklingsmetod men NAF som sin metamodel. Några erfarenheter från detta arbete är [23]:

- Det finns en spänning mellan NAF:s fokus på fysisk *systems engineering* och TOGAF:s fokus på informationssystem.
- Det finns en spänning mellan TOGAF:s fokus på övergripande portföljhantering i en organisation och NAF:s snävare fokus på upphandling.

[23] konstaterar att ramverken är komplementära: TOGAF:s komplexa metod kan förenklas med inspiration från NAF, och NAF:s komplexa metamodel kan förenklas med hjälp av TOGAF. Det norska arbetet har också resulterat i ett flertal förslag till förenklingar av NAF, som nu behandlas av NATO.



## 2.5 Sammanfattning

Försvarsmakten siktar på att realisera modellbaserad förmågeutveckling, MBFU. En hörnsten i detta sammanhang är Enterprise Architecture, EA, med Ministry of Defence Architecture Framework, MODAF, som grund. Fördelarna med EA utifrån en modellbaserad grund är flera, relativt det dokumentbaserade angreppssätt för förbands- och materielutveckling som oftast används idag:

- Förbättrad spårbarhet, samt förmåga att synliggöra och analysera relationer/beroenden
- Styrning och inriktning av utvecklingsverksamhet utifrån ett helhetsperspektiv
- Förbättrad interoperabilitet
- Förbättrad ROI (*Return On Investment*)
- Reducerad risk avseende nya investeringar
- Enklare, snabbare och ytterst, mindre kostsam anskaffning

För att bedriva utveckling med EA som grund krävs ett ramverk. Ovanstående genomgång är på intet sätt uttömmande, men ger en övergripande bild av de vanligast förekommande EA-ramverken inom den militära, såväl som den civila, världen. Många av de ramverk som idag tillämpas av företag och myndigheter har sin grund i IT-relaterad utveckling (och är därmed generellt mer fokuserade på just den biten). Detta är i sig inte konstigt med tanke på hur centralt IT är för en organisations förmåga att implementera visioner och mål. I den militära världen är dock inte enbart IT centralt i utvecklingssammanhang, en utgångspunkt som var central då MODAF och andra militärt relaterade ramverk, skapades. Den norska erfarenheten visar också tydligt på just detta [23]. Inom den militära världen kan man se en konvergens mellan de tre stora ramverk som ofta används av olika försvarsmakter i västvärlden. De tre ”stora” ramverken, MODAF, DoDAF och NAF, är i princip kompatibla/interoperabla, bortsett vissa mindre skillnader. I den civila världen (privat och offentlig verksamhet) är bilden något mer splittrad. Ramverken skiljer sig mer åt även om vissa ”generella” principer, såsom uppdelningen i olika typer av arkitekturer (verksamhet, data/information, applikation och infrastruktur), är ett genomgående tema. Ovan beskrivs två vanligt förekommande ramverk i form av TOGAF och Zachman, men det finns en lång rad exempel på andra ramverk i den civila världen, se [17].

Som vi har sett i genomgången ovan skiljer sig dessa ramverk åt på flera sätt. Alla kallas EA-ramverk, men innehåll och användning är olika. I själva verket är det kanske i kombinationen av ramverk som den största nyttan går att finna. MODAF är genom sin metamodell, M3, fokuserat på att beskriva strukturen hos arkitekturmodeller på ett formellt sätt. Zachman kan snarare ses

som ett sätt att kategorisera artefakter (genom en taxonomi/ontologi), medan TOGAF framförallt sätter processen för utvecklingen av en arkitektur i första rummet (med stöd av ADM). Således kan ramverken med fördel användas i kombination, exempelvis MODAF som struktur för modeller vars framtagning drivs fram av TOGAF ADM.

Modellbaserad utveckling och EA har flera uppenbara fördelar. Samtidigt ska påpekas att det är komplext och framförallt tidskrävande att införa som arbetssätt i en organisation. Förutom att de tekniska aspekterna måste finnas på plats, i form av verktyg, referensmodeller, bibliotek gemensam terminologi etc., vilket ändå får anses vara förhållandevis lätt att åstadkomma, måste personal utbildas och tränas i att tillämpa det modellbaserade angreppssättet. Det sistnämnda kräver resurser, trots att det ibland kan vara svårt att se den omedelbara nyttan av EA.



### 3 EA-verktyg – en översikt

Detta kapitel ger en övergripande beskrivning av EA-verktyg, tillgängliga på marknaden idag. Det finns en lång rad olika verktyg för EA och det ligger utanför denna rapports syfte att ge en komplett bild. För att göra ett urval av verktyg användes Gartners Magic Quadrant [24] som grund. De verktyg som beskrivs i kommande avsnitt går samtliga att återfinna inom den kvadrant som Gartner benämner *leaders* inom branschen, se figur 3.1.

Dock ska påpekas att Gartner inte enbart ser på verktygen i fråga vid sin utvärdering, utan beaktar helheten. Därmed grundas positionen inom kvadranten även på faktorer som marknadsposition, möjlighet att tillhandahålla support, anpassningsförmåga efter kundens behov etc. Dvs. tar även hänsyn till företaget som helhet. Utöver detta tar Gartners utvärdering givetvis hänsyn till faktorer som stöd för modellering, analys, presentationsstöd, scenarioplanering, etc.

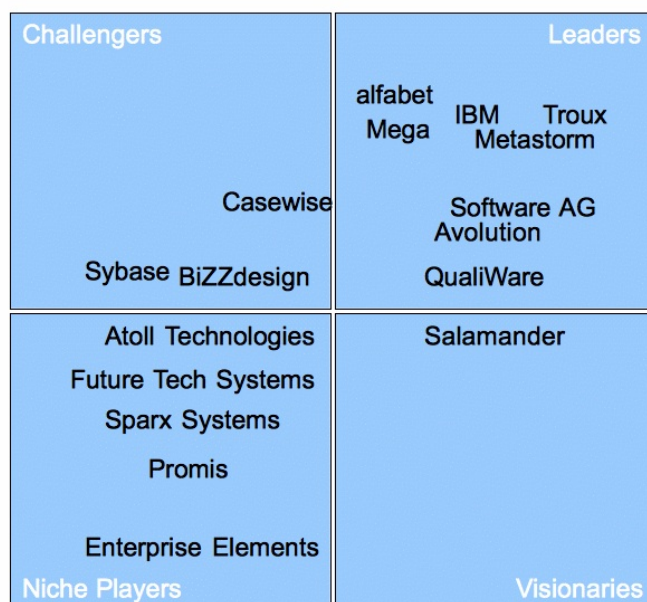
Som kompletterande källa har Enterprise Architecture Management Tool Survey 2008 från TU München använts [25]. Denna undersökning utvärderar ett antal av de ledande EA-verktygen (listan är lik, men inte identisk med Gartners) utifrån scenarier – forskarna vid TUM har tagit fram typanvändningsscenarier som de har testat verktygen mot. Verktygen har sedan mätts inte bara utifrån huruvida uppgifterna i scenariot var möjliga att utföra (*achievement of objectives*) i verktyget, utan också utifrån verktygets användarvänlighet (*tool handling*), konsistens (*procedure consistency*) och integration med andra processer (*procedure integration*). Eftersom rapporten från TUM är från 2008 skiljer sig en del produktnamn från de namn under vilka verktygen idag säljs. I kapitlet används alltid verktygens aktuella namn.

Observera att detta kapitel inte är en utvärdering av verktyg, utan syftet är att ge en översikt av vad som erbjuds på marknaden idag (från de företag som befinner sig i framkant enligt Gartner). Uppgifterna i texten nedan bygger i stor utsträckning på information från de enskilda företagen och de funktioner som beskrivs och dess kvalitet har *inte* verifierats av FOI genom praktisk användning av verktygen.

#### 3.1 Alfabet

Alfabet är ett företag baserat i Tyskland som tillhandahåller produkten planningIT. Enligt egen utsago har företaget en global kundkrets om ca 135 000 användare i 40 olika länder, inom bland annat finans-, telekom-, logistik- och IT-sektorerna.<sup>1</sup>

<sup>1</sup>Om Alfabet, <http://www.alfabet.com/en/about>



Figur 3.1: En något förenklad reproduktion av Gartners Magic Quadrant för leverantörer av EA-verktyg [24].

## planningIT

planningIT är utvecklat för att primärt stödja integrerad planering inom IT-relaterad verksamhet och sträcker sig därmed utöver vad som traditionellt avses med EA vad gäller funktionalitet. Utgångspunkten för det stöd som verktyget ger är att fokusering på helheten vid planering inför IT-relaterade investeringsbeslut och inte enbart på budget och resurser, såsom ofta sker. Mycket är, som namnet på verktyget antyder, fokuserat kring just planering. Verktygsstöd och en metod för arkitekturbaserad planering av IT, förväntas tillsammans ge en solid grund för beskrivning av framtida, önskvärda tillstånd (*to-be*) och hantering av nödvändiga förändringar för att nå eftersträfvade tillstånd (*change management*).<sup>2</sup>

Enligt Gartner [24] är styrkan med planningIT att det tillhandahåller en process-motor (*workflow engine*) som kan integreras med arkitekturarbetet på ett effektivt sätt. Genom detta kan planering och genomförande av ett arkitekturinitiativ effektiviseras genom datorbaserat stöd. Vidare inkluderar planningIT en omfattande metamodel som representerar *best practice* avseende hur information bör struktureras och relateras.

<sup>2</sup>Produktsida Alfabet, <http://www.alfabet.com/en/offering/product//main.aspx>

Enligt egen utsago [26] stöds TOGAF ADM (processen för arkitekturutveckling) fullt ut. Även Zachmans matris går att realisera inom ramen för planningIT. Slutligen stöds även DODAF. Det verkar dock som att det stödet främst avser att planningITs metamodel kan representera DODAFs vyer, snarare än att det har en dedikerad implementering av DODAF.

Enligt TUM:s utvärdering är planningITs starkaste sidor avseende verktygsfunktionalitet (i) konsekvensanalys och rapportering, (ii) stöd för stora datamängder samt (iii) stöd för kommunikation och samarbete. Den svagaste sidan är flexibiliteten i informationsmodellen [25].

## 3.2 Mega

MEGA är ett privatägt nordamerikanskt företag. Enligt egen utsago har företaget över 75 000 användare och finns i över 40 länder.<sup>3</sup>

### MEGA Enterprise Architecture

Genom att integrera EA med *Business Process Analysis* (BPA) samt it-planering tillhandahåller MEGA Enterprise Architecture en lösning som stödjer it-styrning, verksamhetsutveckling samt förändringsledning.

Plattformen tillhandahåller vyer för både verksamhetsarkitektur och it-arkitektur, med integrerade modellerings- och transformationsverktyg som möjliggör ett inkrementellt arbetssätt under arkitekturutveckling. Vidare tillhandahåller plattformen även en omfattande portallösning med vars hjälp de modeller som tas fram kan delas/kommuniceras på ett effektivt sätt inom en organisation, samt som även tillhandahåller beslutsfattare med analysmöjligheter och *dashboards*. Av de tidigare beskrivna ramverken stöds Zachman, TOGAF, DoDAF och NAF.<sup>4</sup>

Enligt Gartners analys [24] är MEGA Enterprise Architecture ett välutvecklat EA-verktyg med en omfattande meta-modell som sträcker sig över perspektiven verksamhet, information, teknik och lösning. Vidare tillhandahåller verktyget en bra mekanism för datautbyte, vilket förenklar integration med andra produkter. Slutligen satsar MEGA en hel del F&U mot just försvarsområdet, vilket eventuellt ger det en bra position inom den världen (genom en god domänförståelse).

Enligt TUM:s utvärdering är Megas starkaste sida avseende verktygsfunktionalitet flexibiliteten i informationsmodellen. Den svagaste sidan är möjligheten att interagera med, redigera och anteckna i visualiseringar [25].

<sup>3</sup>Om Mega, <http://www.mega.com/en/c/company>

<sup>4</sup>Mega EA-produkter, <http://www.mega.com/en/c/solution/p/enterprise-architecture>

### 3.3 Troux Technologies

Troux Technologies grundades 2001 och har sin bas i Austin, Texas. Företaget erbjuder lösningar för strategisk planering i it-sammanhang, samt EA, med kunder inom flera olika branscher såsom finans, telekom, hälso- och sjukvård och offentlig sektor.<sup>5</sup>

#### Troux 9

Troux 9 är en plattform för styrning och inriktning av it-utveckling genom genom tillhandahållande av en sammanhållen syn på resurser, projekt/program kopplat till strategi. Detta ger en bild av verksamheten där det är möjligt att snabbt identifiera aktiviteter och program som är, eller inte är, i linje med gällande verksamhetsstrategi (ytterst mål och vision). Plattformens applikationer kan användas för att stödja planering och exekvering av nödvändiga verksamhetsförändringar.<sup>6</sup> Troux tillhandahåller utökningar till den grundläggande plattformen som stödjer arkitekturutveckling i enlighet med DODAF och MODAF.<sup>7</sup> Det finns även ett tillägg som stödjer modellering, lagring (*repository*) och hantering av samtliga artefakter som TOGAF föreskriver.<sup>8</sup>

Enligt Gartner [24] tillhandahåller Troux en lång rad (avancerade) funktioner för datorbaserad samverkan vid arkitekturutveckling, samt styrning av en organisation utifrån en arkitekturbaserad grund. Värt att nämna är dess funktionalitet för att importera information från externa källor och framförallt förmågan att specificera regler som kan tillämpas för att lösa motsägelser i informationen som då kan uppstå. Genom samverkan med andra aktörer finns även möjlighet till integration med andra användbara verktyg såsom produkter för projektportföljhantering (PPM).

Enligt TUM:s utvärdering är Troux starkaste sidor avseende verktygsfunktionalitet (i) konsekvensanalys och rapportering samt (ii) stöd för stora datamängder. Den svagaste sidan är stöd för kommunikation och samarbete [25].

### 3.4 IBM

IBM är ett multinationellt företag med sin bas i Armonk, New York. Företaget grundades 1911 och har över 400 000 anställda. Idag tillverkar och säljer IBM såväl hårdvara som mjukvara och erbjuder konsulttjänster inom infrastruktur och lagring.<sup>9</sup>

<sup>5</sup>Om Troux Technologies, <http://www.troux.com/company/>

<sup>6</sup>Troux 9, <http://www.troux.com/products/troux\9/>

<sup>7</sup>Verktyg för MODAF, <http://www.modaf.com/Tools/>

<sup>8</sup>Troux stöd för TOGAF, [http://www.troux.com/products/troux\\\_togaf/](http://www.troux.com/products/troux\_togaf/)

<sup>9</sup>Om IBM, [http://www.troux.com/products/troux\\\_togaf/](http://www.troux.com/products/troux\_togaf/)

## Rational System Architect

Rational System Architect är en plattform som stödjer modellering, visualisering och analys av en organisations arkitektur och därtill relaterade verksamhetsprocesser. Verktøyet kan användas för arkitekturutveckling baserat på flera olika standarder, däribland TOGAF, DODAF, MODAF och NAF.<sup>10</sup> System Architect används i stor utsträckning inom det amerikanska försvaret för utveckling av arkitekturer baserat på DODAF [27]. Värt att notera är att IBM även tillhandahåller tillägg till System Architect som möjliggör simulering kopplat till de arkitekturmodeller som tas fram. Detta kan användas för test av en modells robusthet, hur modellen uppträder givet att förhållanden förändras, samt prediktion av effekten på en organisation utifrån föreslagna förändringar.

Gartner anser [24] att System Architect är en robust produkt med ett rikt utbud av funktionalitet som kan anpassas utifrån användarens behov. IBM har ett fokus på att göra resultatet av ett EA-initiativ till någon som verkligen används som grund för beslut. Genom de gränssnitt som System Architect erbjuder mot andra produkter inom portföljhantering, BPM (Business Process Modelling) etc. kan detta realiserats. Gartner pekar dock på att produkten i vissa stycken kan vara onödigt komplex, vilket kan utgöra ett hinder för vissa användare.

TUM:s utvärdering omfattar inte Rational System Architect. Skälet är att de flesta av utvärderingsscenarierna inte går att utföra i programmet, som mest är inriktat på enskilda mjukvaruprojekt [25].

## 3.5 Metastorm

Metastorm grundades 1996 i Baltimore, då huvudsakligen inriktat mot lösningar för BPM. Sedan 2011 är Metastorm del av OpenText, som huvudsakligen levererar lösningar inom ECM (Enterprise Content Management).<sup>11</sup>

### Metastorm Enterprise

Metastorm Enterprise är en programvarusvit inriktad mot stöd till verksamhetsförändring. Metastorm ProVision är en del av denna svit och används primärt för EA. I ProVision finns stöd för att översätta en organisations vision och mål till en modell med kopplingar till de resurser (människor, produkter, processer, system och teknologi) som tillsammans realiserar dessa mål. I ProVision kan arkitekturer baserat på ramverk som Zachman, TOGAF och DODAF utvecklas. Genom Metastorm BPA (Business Process Analysis) är det även möjligt att inkorporera stöd för att dokumentera och analysera komplexa processer i en verksamhet, bla. genom tillämpning av

<sup>10</sup><http://www-01.ibm.com/software/awdtools/systemarchitect/features/>

<sup>11</sup>Om Metastorm, [http://www.metastorm.com/company/overview/\\_index.asp](http://www.metastorm.com/company/overview/_index.asp)



simulering.<sup>12</sup>

Enligt Gartner [24] är Metastorm Provision ett förhållandevis användarvänligt verktyg jämfört med andra, liknande produkter på marknaden. En annan fördel är just dess möjlighet till integration med produkter för BPM och BPA.

Enligt TUM:s utvärdering är Metastorm Provisions starkaste sida avseende verktygsfunktionalitet användarvänlighet. Den svagaste sidan är möjligheten att interagera med, redigera och anteckna i visualiseringar [25].

### 3.6 Software AG

Software AG är ett börsnoterat, tyskt företag med omkring 5000 anställda. Dess produkter har globalt mer än 10000 användare i 70 länder och täcker in ett flertal områden såsom database management system (DBMS), Business Process Management System (BPMS), Service-Oriented Architecture (SOA) och EA. 2010 övertog Software AG IDS Scheer och därmed även produkten ARIS Plattform.<sup>13</sup>

#### ARIS Plattform

Lösningen för EA från SoftwareAg är ARIS-plattformen, som tillhandahåller funktioner för analys, optimering, planering och styrning av en verksamhet utifrån en arkitekturbaserad grund. Plattformen ger möjlighet till synkronisering av it-relaterade förändringar med verksamhetens behov och ger stöd för framtagning av planer som transformerar en verksamhet/organisation i enlighet med uppsatta mål. Vidare stöds EA-ramverk som TOGAF av ARIS.<sup>14</sup>

Enligt Gartner [24] är ARIS ett verktyg med god renommé bland användare och har länge haft en ledande position inom BPA/BMP-området med välutvecklad funktionalitet för modellering och analys. Vidare tillhandahåller plattformen ett brett utbud av funktioner för integration med andra produkter/plattformar såsom exempelvis SAP Solution Manager och Enterprise Service Repository. Slutligen tillhandahålls en omfattande metamodell som ger möjlighet till strukturering och relatering av information som är relevant för en EA-ansats.

Enligt TUM:s utvärdering får ARIS samma betyg avseende verktygsfunktionalitet i alla kategorier utom den svagaste sidan: möjligheten att interagera med, redigera och anteckna i visualiseringar [25].

<sup>12</sup>Metastorm Enterprise, [http://www.metastorm.com/products/overview\\\_index.asp](http://www.metastorm.com/products/overview\_index.asp)

<sup>13</sup>Om SoftwareAG, <http://www.softwareag.com/corporate/company/companyinfo/overview/default.asp>

<sup>14</sup>Aris Plattform, [http://www.softwareag.com/corporate/products/aris\\\_platform/default.asp](http://www.softwareag.com/corporate/products/aris\_platform/default.asp)

### 3.7 Avolution

Avolution grundades 2001 som en avknoppning från universitetsvärlden och har idag huvudkontor i Sydney. Företaget har en global kundkrets i över 70 länder. Företagets produkter är främst inriktade mot EA, verksamhets- och processmodellering.<sup>15</sup>

#### Abacus

Abacus är ett kraftfullt verktyg för modellering, analys och förståelse av komplexa verksamheter utifrån dess människor, processer och teknologi. Genom funktionalitet för analys av arkitekturer (trade-off) ges stöd till generering av planer inför investeringsbeslut. Genom verktyget kan nytta, effektivitet och kostnad predikteras för alternativa strategier. Abacus har även avancerade former av kommunikationsstöd, genom grafer, 2-dimensionella bilder och 3-d visualisering.<sup>16</sup> ABACUS stödjer arkitekturarbete enligt TOGAF, DODAF, MODAF och Zachman.<sup>17</sup>

Enligt Gartner [24] har Avolution, genom Abacus, en unik XML-baserad ansats som ger en flexibilitet som få andra produkter kan erbjuda. Abacus skapades med EA som utgångspunkt och har inte som många andra verktyg vuxit fram från modelleringsverktyg tidigare inriktade mot andra nischer. Därmed har det få begränsningar. Företaget har som nämnts ovan en bakgrund inom forskningsvärlden och genom detta har det fortfarande ett visst fokus mot forskning, framförallt inom nya och innovativa sätt att analysera och presentera data.

TUM:s utvärdering omfattar inte Abacus.

### 3.8 QualiWare

QualiWare är ett privatägt Danskt bolag som grundades 1991 med huvudkontor i Köpenhamn. Företaget erbjuder konsulttjänster och ett flertal olika modelleringsverktyg, däribland EA-verktyg.<sup>18</sup>

#### Qualiware Enterprise Architecture – QEA

QEA ger stöd för att skapa en holistisk vy av en verksamhet utifrån grafiska och textuella beskrivningar av strategier, information, system, resurser, processer och relationerna mellan dessa. Genom att modeller som tas fram, mha av verktyget, är objektbaserade är det möjligt att skapa relationer mellan godtyckliga objekt i en modell. Förutom att verktyget ger

<sup>15</sup>Om Avolution, [http://www.avolution.com.au/about\\\_us.html](http://www.avolution.com.au/about\_us.html)

<sup>16</sup>Avolution Abacus, <http://www.avolution.com.au/products.html>

<sup>17</sup>Arkitekturramverk i Abacus, <http://www.opengroup.org/togaf/cert/showcase/participants/avolution/index.html>

<sup>18</sup>Om QualiWare, <http://www.qualiware.com>

ett kraftfullt modelleringsstöd, har det även funktionalitet för gemensam lagring (*repository*), publicering av modeller via intranät/webbläsare, vilket tillsammans ger bra möjligheter för kommunikation av organisationens mål, vision och pågående förändringsarbeten. QEA stödjer ett flertal arkitekturramverk, däribland Zachman och TOGAF.<sup>19</sup>

Enligt Gartner [24] tillhandahåller QEA en omfattande meta-modell som täcker in verksamhets-, informations-, teknologi- och lösningsarkitektur på ett sätt som är tillräckligt detaljerat för att kunna representera information av intresse för en (verksamhets-)arkitekt. Vidare är företagets svit av produkter starka inom portföljhantering och stöd till hantering av investeringar.

TUM:s utvärdering omfattar inte QualiWare.

### 3.9 Sammanfattning

Det finns idag en mogen marknad för EA-verktyg, där ett flertal aktörer konkurrerar med varandra. Många verktygsförsäljare har under 2011 börjat ompositionera sig och talar mer om affärsplanering och mindre om *enterprise architecture*, en term som är belastad i mångas ögon. Konkurrensen har också skärpts av att verktygsköpare allt mer noggrant behöver demonstrera vilket affärsvärde som deras EA-satsningar faktiskt tillför. Slutligen har branschen under 2011 påverkats en del av uppköp och förvärv, såsom SAP:s köp av Sybase och OpenTexts köp av Metastorm [28].

En stor nytta med EA-verktyg är deras möjlighet att versionshantera och säkert lagra all den dokumentation som en fullständig arkitekturbeskrivning av en stor organisation ofrånkomligen medför. Denna *repository*-tankegång framhålls ofta i verktygssammanhang.

En annan stor nytta med EA-verktyg är möjligheten att göra automatiserade analyser på all den data som finns i en fullständig arkitekturbeskrivning av en stor organisation. Detta är dock väsentligen ett tämligen utforskat område, som sällan lyfts fram. De stora kommersiella verktygen har i själva verket – med undantag av Abacus – sällan särskilt kvalificerade analysmöjligheter av den typ som beskrivs i kapitel 6.

---

<sup>19</sup>Om QualiWare, <http://www.qualiware.com>

## 4 Kraftfulla simuleringsverktyg för MBFU

### 4.1 Introduktion

När det gäller angreppssätt för förmågeutveckling infinner sig Försvarmakten i ett övergångsskede från dokumentbaserat till modellbaserat. Enbart en tillämpning av ett modellbaserat angreppssätt räcker dock inte för att utveckla förmågor som uppfyller de krav som ställs på Försvarmakten, utan ytterligare verktyg och metoder behövs som stöd till arbetet. Inom förmågeutvecklingsprocessen finns t.ex. behov av stöd vad gäller värdering av olika lösningar, identifiering av konsekvenser av olika handlingsalternativ, synliggörande av beroenden, visualisering av arkitekturmodeller och förbandskonfigurationer m.m.

För många av dessa behov, exempelvis värdering av olika lösningar och konsekvensanalyser av olika handlingsalternativ (s.k. "what if"-analyser) brukar simulering omnämnas som ett utmärkt verktyg. Simulering kan också vara ett viktigt hjälpmedel under ett arkitekturarbete för att närmare studera komplexa skeenden och frågeställningar. Resultatet från en simulering kan bland annat ge värdefull feedback till utvecklingen av själva arkitekturen. Genom att utsätta en arkitekturmodell för dynamisk analys m.h.a. simulering kan man få stöd för konsekvens-, risk- och sårbarhetsanalys, men även som stöd vid dimensionering.

Det är ofta svårt att förstå en arkitekturmodell och därför är simuleringar av stor nytta för att kunna visualisera arkitekturmodellen på ett sätt som underlättar användarens förståelse för sammanhanget. Genom att modellera både system och deras interaktioner i dimensionerande scenarier kan simuleringar användas för att studera dimensioneringsaspekter. Det blir då möjligt att studera exempelvis logistikfunktioner såsom drivmedelsförbrukning givet olika kontexter eller studera hur länge ett förband kan nyttjas innan försörjningskedjan haltar, etc. Att även kunna spela upp ett scenario, pausa, zooma in och studera detaljer för att bättre illustrera och studera den verksamhet som modellerna representerar, är ovärderlig.

För att tillmötesgå alla dessa behov krävs avancerade och kraftfulla simuleringssystem som kan snabbt och effektivt producera lämpliga simuleringar. Om byggstenar och vyer från MODAF ska användas för att strukturera upp och modellera försvarsverksamheter och -förmågor så måste det finnas ett lika flexibelt och kraftfullt simuleringssystem som kan simulera dessa konstruerade modeller. Försvarmaktens modellbaserade förmågeutveckling begränsar sig inte till en viss verksamhet inom försvaret utan innefattar potentiellt alla dagens och framtida försvarsverksamheter. Att försöka införskaffa ett simuleringsverktyg för varje specifik syfte är kostsamt och orealistiskt. Man måste kunna anskaffa ett generellt simuleringsverktyg

som kan svara mot de flesta (och med smärre modifieringar nästan mot alla möjliga) försvarsapplikationer. Ett sådant generellt simuleringsverktyg brukar kallas *simuleringsramverk*.

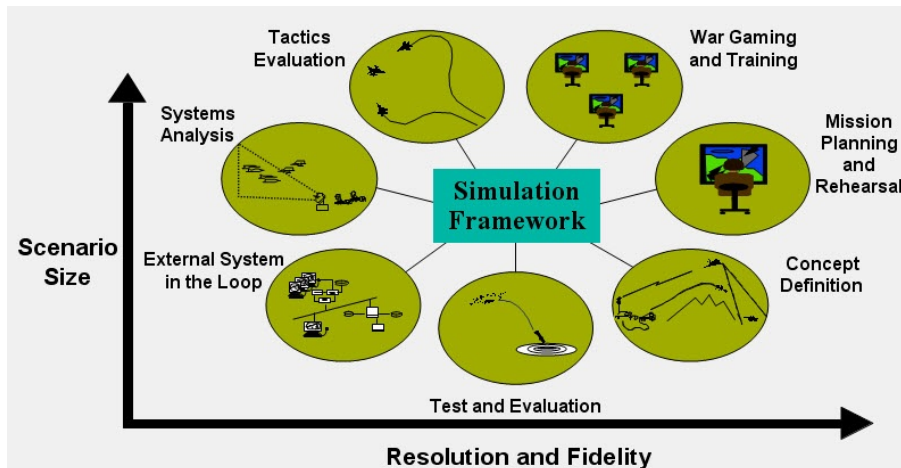
## 4.2 Vad är ett simuleringsramverk?

Ett ramverk för modellering och simulering definieras som en infrastruktur, tillhandahållande tjänster för modellutveckling, scenarioupbyggnad, exekvering, analys och visualisering. Vissa simuleringsramverk stödjer även en komponentbaserad modelleringsmetodik där större modeller byggs upp av mindre modeller som kommunicerar med varandra. I dessa ramverk kan man relativt enkelt koppla ihop mindre modellobjekt till större modeller och snabbt studera deras beteende i en simulering, egenskaper som ej uppnås med andra koncept som *High-Level Assembler* (HLA) och distribuerad simulering. I många fall medföljer även ett modellbibliotek till dessa simuleringsramverk, vilket gör det möjligt att direkt efter anskaffning komma igång att använda verktyget. Men det är ändå inte denna aspekt av simuleringsramverk vi är främst intresserade av, då våra modeller utvecklas i enlighet med MBFU/MODAF.

Den aspekt av simuleringsramverk som är intressant i detta sammanhang är att det tillhandahåller en infrastruktur innehållande en simuleringskärna med ett antal simulerings-tjänster samt ett antal standard applikationer som stöd för scenariogenerering, exekvering, analys och visualisering. I denna rapport när vi pratar om simuleringsramverk så syftar vi specifikt till militära simuleringsramverk som är avsedda att stödja militära tillämpningsapplikationer (motsvarande funktionalitet finns nämligen paketerat som stödverktyg även för andra domäner, exempelvis telekommunikation [29] eller medicin [30].)

Exempel på tjänster som stöds av en simuleringskärna och infrastruktur ett simuleringsramverk erbjuder är tidsstyrning, databashantering, minnesallokering, schemaläggning, terrängrepresentation, dataloggning, återuppspelning, grafik för resultatpresentation och mycket mera. Genom detta skapas förutsättningar för att simuleringsstudier (inom nästan alla för försvaret viktiga tillämpningsområden såväl luft-, sjö-, mark- som informationsarenan) underlättas och påskyndas.

Vad Figur 4.1 försöker illustrera är att simuleringsramverk är ett simuleringsverktyg som kan användas för många olika syften, med varierande scenariostorlek och omfattande antal deltagare. Ett simuleringsramverk tillåter att scenariodeltagarna (aktörer, aktiviteter, system) modelleras med olika upplösningar och trovärdighetsgrader. Detta gör att verktygen går att använda för skilda syften, exempelvis såväl taktikutveckling som analytiska studier. Det är möjligt att göra batch-körningar med viss parametervariation mellan varje körning, och det finns ofta stöd för aggregering eller disaggregering i ett scenario. Det finns ofta möjlighet att på ett enkelt sätt koppla in verkliga



Figur 4.1: En generell och illustrativ beskrivning av ett simuleringsramverk. Figuren försöker illustrera att ramverken i sig inte sätter några begränsningar vad det gäller antalet deltagare i ett scenario samt tillåter att deltagarna modelleras med valfria upplösnings- och trovärdighetsgrader. Därmed kan ramverken användas för många olika syften.

system till en pågående simulering och låta det verkliga systemet agera som en deltagare i scenariot, styra en virtuell aktör eller blir styrd av en sådan. En simulering kan exekveras i realtid, snabbare än realtid eller långsammare. Den mängd av data som produceras vid simuleringar loggas genom kanalisering till ett antal resultatfiler som senare kan användas för olika analyser.

### 4.3 Nyttan med simuleringsramverk

Huvudsyftet med användningen av simuleringsramverk är att få en gedigen bas där en infrastruktur som tillgodoser behov för många tillämpningsprojekt redan finns inbyggd. Detta möjliggör att tillämpningsprojekten kan koncentrera sina resurser på kärnverksamheten och inte på simuleringsutvecklingsverksamheten.

Vid en analys av fördelar respektive nackdelar med att använda militära simuleringsramverk framträder ett antal iakttagelser, varav de viktigaste är [31]:

- Användningen av simuleringsramverk kan avsevärt minska tiden som krävs för att skapa komplexa simuleringsystem för ett specifikt militärt tillämpningsområde.
- Simuleringsramverk är effektiva och flexibla; de ger möjlighet till ett snabbt scenario- och prototypskapande och det är oftast lätt att förbättra eller byta ut inbyggda modeller.

- Den objektorienterade och strikt komponentbaserade konstruktionen som de flesta ramverk har bidrar till ett objektorienterat och modulärt arbetssätt.
- Simuleringsramverk uppfattas användarvänliga eftersom det är samma miljö för alla simuleringar och användaren alltid har samma välkända gränssnitt.
- Ett simuleringsramverk optimerar tekniska resurser eftersom mycket av det repetitiva och administrativa arbetet redan ingår.
- Programvaran är i allmänhet relativt väl avlusad, och merparten av kostnaderna för underhåll och förvaltning drabbar tillverkaren av ramverket och delas därigenom av alla kunder.
- Det är en stor fördel, om inte ett måste, att simuleringsramverket i fråga stöds av ett stabilt företag som även garanterar framtida underhåll och vidareutveckling av programvaran.

Några av de viktigaste nackdelarna som också bör nämnas i sammanhanget är:

- Simuleringsramverk kan innehålla mycket mer än vad som behövs för en specifik frågeställning, vilket kan leda till att det blir kontraproduktivt.
- Simuleringsramverk kan ha en hög utbildningströskel och kräva en längre inskolningstid.
- Många simuleringsramverk erbjuder en hög grad av flexibilitet och frihet vilket resulterar att det ställs ett stort ansvar på användaren.

#### 4.4 Några exempel på simuleringsramverk

Det finns nog hundratals företag på marknaden som gärna skulle vilja (alternativt redan försöker) kalla sina produkter för simuleringsramverk. Det finns dock få av dessa produkter som klarar av alla de krav som ställs på ett sådant system (se avsnitt 4.2).

Vid ett antal marknadsundersökningar som tidigare genomförts vid FOI [32] och [33] har några simuleringsramverk identifierats och vissa av dem utvärderats. Baserat på erfarenhet från dessa arbeten samt en ny genomgång av marknaden har några kandidater (med hänsyn taget främst till företagets stabilitet och produktens mognadsgrad men även modifierbarhet, flexibilitet, funktionalitet, etc.) valts ut:

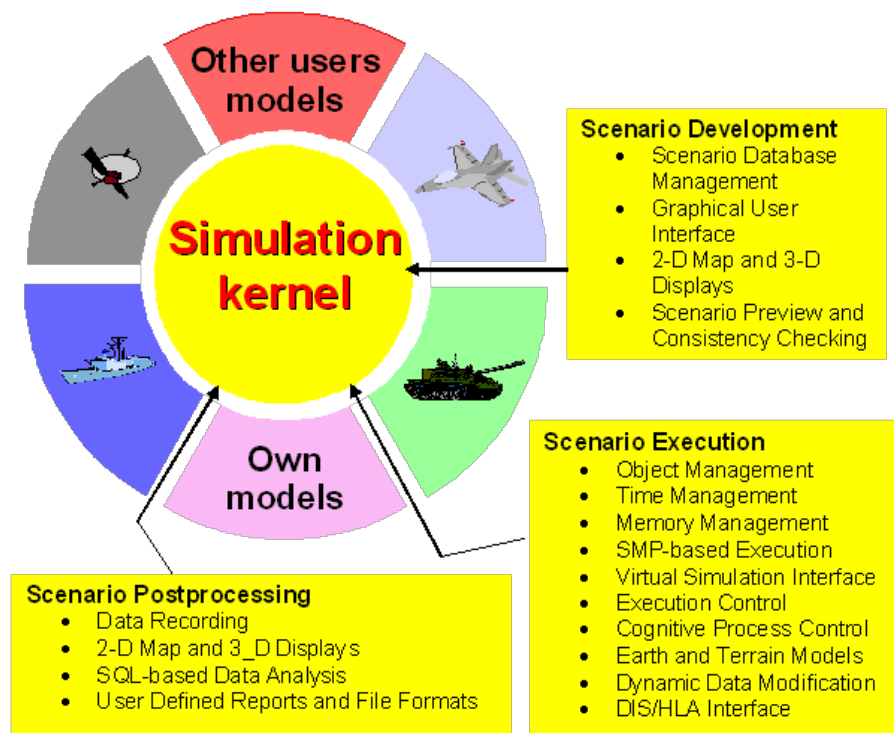
- *Flames från Ternion*
- *Strive, Aeria, Stage och AI.implant från CAE*

- *MASA Sword och MASA Life från MASA Group*
- *VR-Forces från MÅK*

Dessa beskrivs översiktligt nedan. Värt att nämna är att vi i denna genomgång inte har tagit hem någon av dessa produkter för utvärdering, men vi gör ändå den kvalitativa bedömningen – med avseende på ovannämnda faktorer – att dessa hör till de mest intressanta simuleringsramverken på marknaden idag ifall man vill satsa på en sådan utvärdering under 2012.

#### 4.4.1 Flames från Ternion

Flames (Flexible Analysis Modeling and Exercise System) är ett simuleringsramverk utvecklat av Ternion Inc. i USA under flera årtionden. Flames levereras med en uppsättning av exempelmodeller inom alla de fyra traditionella försvarsområden; mark, sjö, luft och rymd. Flames har använts på FOI mellan 1997 och 2006 för såväl luft- och markstridssimuleringar som för undervattenssimuleringar. (I Sverige används Flames idag på SAAB.)



Figur 4.2: Exempel på tjänster som ingår i Flames simuleringskärna [34].



Flames är en familj av COTS<sup>1</sup>-produkter som tillsammans skapar ett ramverk med vars hjälp en användare kan enkelt sätta samman komplexa så kallade LVC-simuleringar (live, virtual, and constructive simulations). Flames har en öppen, objektorienterad arkitektur som ger möjlighet att skapa egna simuleringar anpassade efter behov, samt enkelt ändra simuleringarna då behovet förändras. Trots detta är Flames klar för användning direkt vid anskaffning, dvs. det kräver i sig inte stora modellutvecklingsarbeten innan verktyget kan användas. Detta gör Flames till ett flexibelt simuleringsramverk för modellering, analys, träning och utbildning [35].

Verktyget stödjer utöver modellering av fysiska egenskaper för olika materielsystem även beteendemodellering. Ett antal fördefinierade modeller av mänskligt beteende, så kallade kognitiva modeller följer med verktyget. Dessa fördefinierade modeller kan vara till stor hjälp om deras beteende är relevanta för frågeställningen man vill studera, men dessa måste ändå ses som exempelmodeller vars mönster man följer när man skall skapa egna beteendemodeller.

Flames är designat för att vara utbyggbart i bemärkelsen att det finns en väl dokumenterad process för att bygga egna modeller som nyttjar de tjänster som tillhandahålls av ramverket. Utöver tjänsterna tillhandahåller ramverket även applikationer för scenariogenerering, exekvering, visualisering och analys. Scenarier skapas normalt genom att konfigurera ett antal aktörer bestående av olika tekniska utrustningar och kognitiva modeller, vilka antingen kan styras av beteendemodeller eller av mänskliga operatörer [31].

Enligt Ternion är Flames ett simuleringsverktyg som kan användas för många olika syften, med varierande scenariostorlekar och omfattande antal deltagare med olika upplösningar och trovärdighetsgrader. De främsta fördelarna med att använda Flames enligt företaget omfattar ökad produktivitet och återanvändning samt minskade kostnader i hela simuleringens livscykel (design, utveckling, underhåll, användning och förbättring) [35].

#### 4.4.2 Strive, Aeria, Stage och AI.implant från CAE

CAE (Canadian Aviation Electronics) är en ledande leverantör av modellerings- och simuleringsteknik till försvarsmakter över hela världen. CAE har utvecklat avancerade simuleringsprodukter samt integrerade tränings- och utbildningstjänster i decennier. Under 2002 annonserade CAE ett samarbete med Boeing, världens ledande leverantör av missilförsvar”. Detta var ett genombrott för det kanadensiska företaget eftersom Boeing inte ville använda CAEs senaste produkter enbart till utbildning av soldater för användning av missilförsvaret, utan även för konstruktion, provning, utvärdering och utveckling av dessa system. CAEs viktigaste bidrag till Boeings missilförsvarsarbete kretsade kring en ny simuleringsprodukt, kallad

---

<sup>1</sup>Commercial Off The Shelf

Strive, vilket Boeing ville använda för utveckling och utvärdering av alla luftförsvars relaterade system, dvs sensorer, stridsplan, ledningssystem, kommunikationssystem, etc. [36].

## Strive och Aeria

Strive introducerades då som ett simuleringsramverk som skulle ge systemutvecklarna möjlighet att enkelt konstruera komplexa och interoperabla simuleringar. Det marknadsfördes därmed som ett komponentbaserat, distribuerat och HLA-kompatibelt utvecklings- och exekveringsmiljö för modellering och simulering. Strive skulle förkorta utvecklingscykeln för M&S avsevärt genom att hjälpa utvecklaren att fokusera på att bygga modeller och inte bry sig om hur modellerna kommunicerar, interagerar och exekverar i realtid [37]. Strive levererades med ett bibliotek av färdiga modeller. Även en CGF<sup>2</sup>-programvara följde med Strive som kunde automatiskt producera trovärdiga datorgenererade aktörer som hade användardefinierade beteenden och realistiska interaktioner med andra aktörer [38]. Det finns dock inget underlag som styrker att Strive salufördes till någon annan aktör på marknaden än Boeing.

Däremot skapade CAE under 2007 ett nytt företag med namnet Presagis, vars främsta produkt skulle vara ett nytt kraftfullt COTS-verktyg för modellering och simulering, förmodligen Strive. CAE hade då just förvärvat några av världens främsta företag inom branschen (*TERREX, Engenuity teknik och MultiGen-Paradigm*) som levererade avancerade tredjepartsprodukter till simuleringsverktyg. Men Strive som en fristående produkt marknadsfördes aldrig av Presagis heller, utan delar av det ingick i flertalet av Presagis produkter. Dock har Presagis i skrivande stund (mot slutet av november 2011) annonserat att på IITSEC-2011 (går i Orlando, USA under v 48 2011) kommer att introducera och ge en första demonstration av Presagis senaste produkt, simuleringsramverket Aeria [39]. Mycket talar om att Aeria bygger på marknadsens en gång mest lovande simuleringsramverk i början av 2000-talet, Strive. Därför utfärdas en stark rekommendation till intresserade av avancerade militära simuleringsramverk att följa den tråden.

Presagis är förövrigt idag en ledande industri inom militär M&S med tusentals kunder i 60 länder över hela världen, och är främst känd för sina ledande verktyg, bl.a. *AI.implant, Creator, SeaWind, Stage, Terra Vista, Vega Prime and VAPS XT*. Två av dessa verktyg är intressanta i detta sammanhang, vilka vi ska titta närmare på.

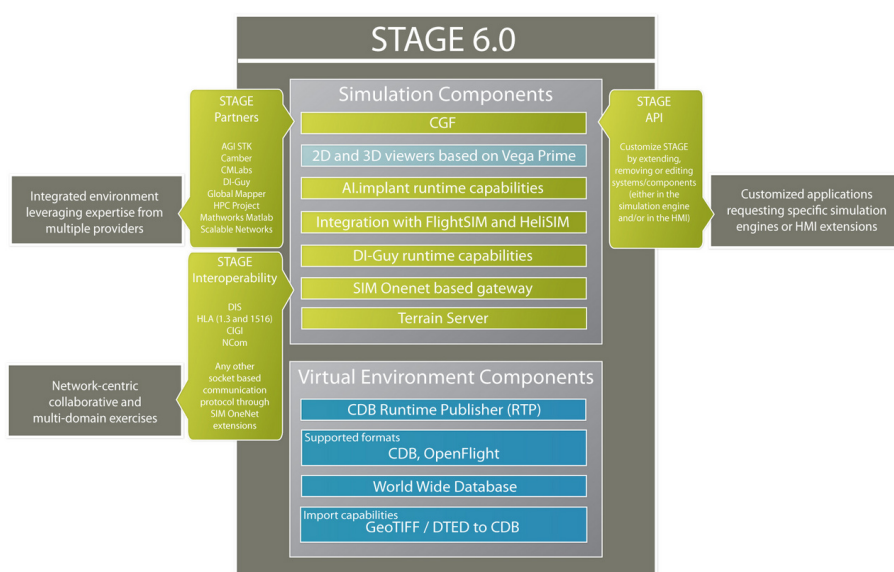
## Stage och AI.implant

Stage (Scenario Toolkit And Generation Environment) är ett simuleringsystem som ger användarna möjlighet att skapa och exekvera komplexa scenarier

---

<sup>2</sup>Computer Generated Forces

främst för utbildning och analys. Systemet möjliggör för både programmerare och icke-programmerare att arbeta i en flexibel och integrerad simuleringsmiljö för att bygga dynamiska och interaktiva taktiska och operativa scenarier. Scenarier i Stage byggs genom att konfigurera luft-, mark- och sjöplattformar som kan utrustas med vapen och andra utrustningar. Dessa plattformar interagerar sedan genom upptäckt, kommunikation, engagemang och destruktion [40]. Figur 4.3 illustrerar en högnivårepresentation av Stagekomponenter.



Figur 4.3: En högnivårepresentation av Stage-komponenter [41].

Stage har använts och används fortfarande på FOI som simuleringsmiljö för både träning och analytiska tillämpningar. Som koncept har Stage stora likheter med andra simuleringsramverk t.ex. Flames. Möjligheterna att integrera befintlig och hemmabyggda programvaror i Stage har undersökts av tidigare FOI-projekt, och det har visat sig att det finns goda möjligheter till integration av befintliga plattformsmodeller i Stage [33]. FLSC (Flygvapnets Luftstridssimuleringscentrum) på FOI är en världsledande simuleringsanläggning för bemannad luftstrid som använt Stage i flera år. Så på FOI finns det idag en mer heltäckande förståelse av verktyget och dess användningsområden. Av Presagis marknadsförs Stage som ett idealiskt verktyg som kan användas för att lösa kritiska problem i vitt skilda simuleringsapplikationer, från utbildning och analys till simuleringsbaserade anskaffning (SBA) och forskning [41].

AI.implant modellerar både mänskliga rörelsen och intelligenta beteenden inom en simulering. Verktyget är mycket effektivt då man snabbt önskar

generera realistiska folkmassor för urbana miljöer samt för att ge kropparna ett någorlunda trovärdigt mänskligt beteende.



Figur 4.4: AI-agenter i ett upploppsscenario [42].

Stage och AI.Implant kan visserligen användas som fristående verktyg, men de kan även användas integrerat, exempelvis kan en entitet i Stage kontrolleras av AI.Implant. Så integrationen mellan Stage och AI.implant låter användaren att installera artificiell intelligens för utvalda Stage-plattformar. Presagis påstår att man med AI.Implant lätt kan skapa autonoma plattformaktörer (exempelvis ett fordon) och mänskliga agenter med olika uppdrag som lätt kan skala från en enda intelligent enhet till stora folkmassor med grupp beteenden [42].

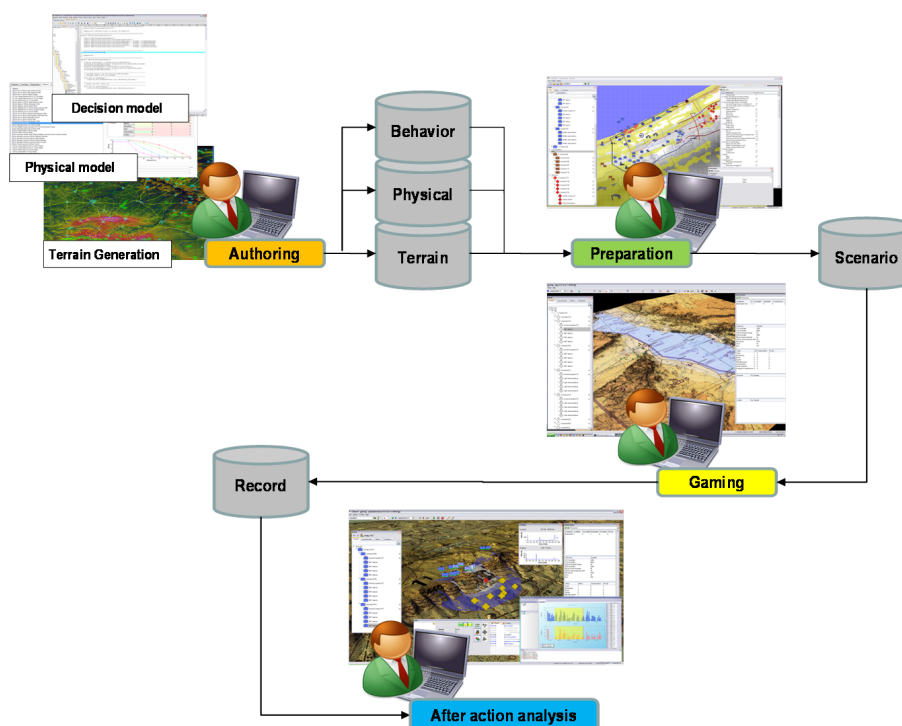
FOI har sedan tio år tillbaka haft kontakter med CAE om Strive och med Presagis om Stage, AI.implant och deras andra verktyg, men idag används enbart Stage på FOI.

#### 4.4.3 MASA Sword och MASA Life från MASA Group

MASA är en fransk produkt som även är yngst bland alla simuleringsramverk som går igenom i detta kapitel. Även företaget med tre huvudkontor i tre länder (Frankrike, USA och Singapore) och totalt ca 60 anställda lär vara minst bland alla företag som omnämns i detta sammanhang.

MASA har i princip två produkter; MASA Sword och MASA Life. MASA Sword finns i två varianter, dels *Defence* för försvarsapplikationer med fokus på träning och utbildning av befälhavare på olika nivåer, och dels *Security* för

säkerhetsapplikationer med fokus på träning och utbildning av staber, chefer och andra ledare i krishanteringsapplikationer. Medan MASA Life å andra sidan består av tre moduler; *Brain*, *Form*, och *World*. Där *Brain* tillhandahåller ett modelleringsramverk för utveckling av strukturerade AI-modeller samt tillhörande beteendebibliotek. *Form* står för ett ramverk för skapande av virtuella karaktärer samt tillhörande bibliotek. Och *World* tillhandahåller ett ramverk för scenarioutveckling och 3D-visualisering [43].



Figur 4.5: Den funktionella arkitekturen hos MASA Sword [44].

I MASA utför simulerade agenter sina uppdrag på samma sätt som soldater skulle ha gjort i verkligheten, dvs:

- i enlighet med doktrinen
- reagerar på dynamiska händelser
- ger lämpliga order till sina underlydande
- rapporterar om den aktuella situationen till sina överordnade

Som fördelar med att använda MASA-produkter räknar företaget upp bland andra, (i) det är lätt att skapa övertygande träningsupplevelser (ii) det går att

skapa professionella och kvalitativa visuella applikationer utan större kostnad eller risk (iii) man kan fokusera resurserna på det egentliga målet, exempelvis träningen eller analysen.

#### 4.4.4 VR-Forces från MÄK

MÄK är ett företag inom koncernen VT Systems Inc., beläget i Cambridge, Massachusetts, USA, som är nischad på att utveckla avancerade simuleringssystem. Genom kombination av sofistikerade COTS-programvaror och egenutvecklade systemkomponenter bygger MÄK trovärdiga virtuella 3D-miljöer i vilka kunderna kan träna, planera, analysera, experimentera, prototyputveckla och demonstrera. MÄK har flera av de stora internationella försvarsindustrier, bland andra *Lockheed Martin*, *Boeing*, *Raytheon*, *BAE* som kunder [45].

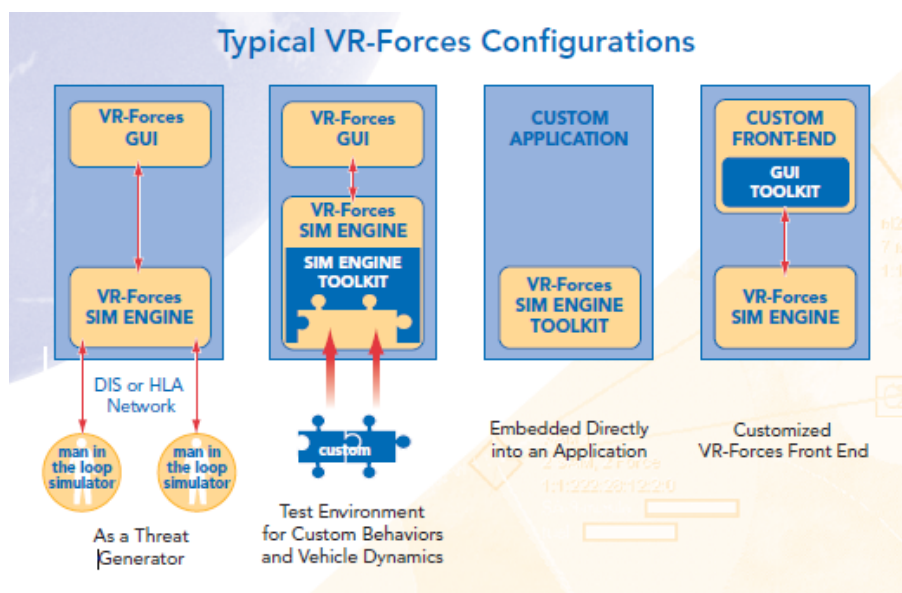
VR-Forces är ett kommersiellt simuleringsramverk från MÄK som är allmänt känt för sina sofistikerade CGF-er (Computer Generated Forces, en sorts AI-modeller). Samtidigt har MÄK lagt ganska mycket kraft på grafiken och nyttjar till synes påkostade verktyg för 3D-visualisering, så att det ska vara lätt att bygga scenarier. Man får därför intrycket att fokuset på ramverket har lagts på det att man snabbt och enkelt skall kunna implementera ett tänkt scenario och sedan visualisera förloppet för att se hur det gick. Typiskt skulle detta kunna användas för att pröva planer och nya koncept utan alltför stora krav på detaljupplösning [32].

MÄK har en stor portfölj av simuleringsprodukter och försöker erbjuda sina kunder en komplett simuleringlösning. VR-Forces uppfattas som ett kraftfullt och flexibelt simuleringsramverk i MÄKs portfölj så till vida att det tillåter användaren att integrera sina specifika modeller i simuleringssmiljön, för att skapa och exekvera militära mark-, sjö- och luftstridsscenarier efter eget behov [32].

Ramverket kan användas för att simulera allt från några få entiteter upp till större taktiska scenarier med många inblandade aktörer. Det finns dessutom stöd för aggregering och disaggregering vilket gör att man kan växla från mer detaljerade studier till att få en övergripande bild av händelseförloppet. Precis som de tidigare nämnda simuleringsramverken, så även med VR-Forces, följer en uppsjö av modeller för en mängd olika aktörer och vapensystem. VR-Forces har en kraftfull simuleringssmotor som under scenarioexekveringen ser till att människor och plattformar interagerar med terrängen, följer vägar, rör sig i konvojer, undviker hinder, kommunicera via simulerade radioapparater, upptäcker och engagera sig i fienden, beräkna skador, etc. [46].

## 4.5 Sammanfattning

Försvarsmaktens modellbaserade förmågeutveckling begränsar sig inte till en viss verksamhet inom försvaret utan innefattar potentiellt alla dagens och



Figur 4.6: En typisk konfiguration av VR-Forces [46].

framtida försvarsverksamheter. Att försöka införskaffa ett simuleringsverktyg för varje specifik försvarsverksamhet är kostsamt och orealistiskt. Att försöka utveckla stora, komplexa och generella simuleringsystem från grunden är inte heller längre försvarbart, det skulle ta alltför lång tid, vara alltför kostsamt och t.o.m. medföra en stor risk att simuleringsprojektet havererar [47]. Om MODAF ska användas för att strukturera upp och modellera försvarsverksamheter och -förmågor så måste det finnas ett lika flexibelt och kraftfullt simuleringsystem, *ett simuleringsramverk*, som kan simulera dessa konstruerade modeller. Viktigt är därför att inhämta och utnyttja den senaste kunskapen från omvärlden om det aktuella utbudet av kommersiella simuleringsramverk, deras möjligheter och begränsningar att användas i ett MBFU-sammanhang. Mångårig utvecklingstid är ofta satsad och mångårig erfarenhet och kunskap från olika tillämpningsområden finns ofta inbyggd redan från början i ett simuleringsramverkskoncept.

Ett väl designat och välfungerande simuleringsramverk tillsammans med ett ackrediterat modellbibliotek skulle kunna erbjuda en stabil grundplattform. Man skulle då snabbt och effektivt kunna prova sina idéer, snabbt skaffa sig en uppfattning om en viss situation, värdera olika handlingsalternativ, generera beslutsunderlag, se resultat av en simulering m.m.

Det simuleringsramverk som väljs bör även ha egenskaper som tillåter bl.a. anpassning av verktyget till användarens behov. För att kunna lösa mer specialiserade uppgifter bör såväl modifiering av befintliga modeller som

modifiering av standardapplikationer som följer med ramverket tillåtas. Som nämnt, vissa simuleringsramverk stödjer även en modelleringsmetodik där större modeller byggs upp m.h.a. mindre som kommunicerar med varandra via väldefinierade gränssnitt. Detta underlättar en evolutionär modellutveckling då delmodellerna enkelt kan ersättas av nya sådana med önskad funktionalitet. Med ett bra simuleringsramverk bör man således snabbare och effektivare kunna nå fram till önskat resultat.

Givet att man bestämmer sig för att nyttja konceptet simuleringsramverk inom M&S-stöd för MBFU så blir det främst tre frågeställningar intressanta att utforska inför en eventuell anskaffning:

1. Vilka simuleringsramverk finns som innehåller det beredaste stödet inom ramen för FMs frågeställningar?
2. Hur mappar vi modeller utvecklade i enlighet med MODAF/MBFU till det valda simuleringsramverket?
3. Hur utformas och driftsätts en sådan infrastruktur så att man med dess hjälp snabbt och kostnadseffektivt tillgodoser det mesta av simuleringsbehoven inom MBFU?





## 5 Modelleringsmetod för exekverbar MODAF

### 5.1 Introduktion

MODAF har stött på ett visst motstånd inom Försvarmakten. MODAF-modeller anses svåra att

1. utveckla;
2. förstå, i synnerhet för beställare, kunder, användare och andra intressenter;
3. transformera, det vill säga omvandla till artefakter för nästa steg i utvecklingsprocessen så som att omvandla systemvyer till källkod.
4. analysera, till exempel använda som indata till (automatiserat) beslutsstöd.

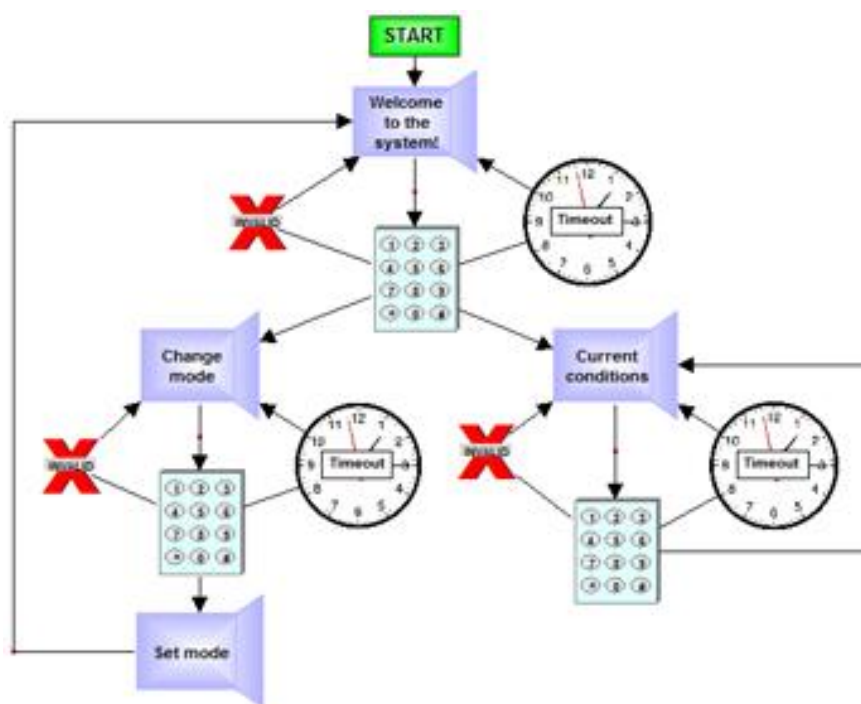
Alla fyra invändningar drabbar i själva verket även andra former av traditionell modellbaserad utveckling, i synnerhet verksamhetsmodellering. Inom senare år har det dock vuxit fram en framgångsrik lösning på dessa problem inom modellbaserad mjukvaruutveckling: domänspecifik modellering, förkortat DSM [48].

MODAF är ett generellt modelleringspråk avsett för att modellera allt inom försvarsdomänen. Ett *domänspecifikt språk* är däremot anpassat för en smal domän; modelleraren uttrycker sin lösning med välbekant terminologi och notation hämtad direkt ur den aktuella problemdomänen. Upprepade studier visar konsekvent på att utveckling med domänspecifika språk är 5 - 10 gånger effektivare än manuell programmering [48]. Det saknas däremot dokumenterade produktivitetsökningar för utveckling med generella modelleringspråk, även inkluderat väletablerade industristandarder som UML [49].

I detta kapitel beskriver vi först (avsnitten 5.2 och 5.3) hur domänspecifik modellering löser de fyra ovannämnda problemen, och visar sedan (avsnitt 5.4) hur domänspecifik modellering kan bedrivas inom ramen för MODAF på ett sätt som gör det möjligt att återanvända MODAF-verktyg, expertis och modeller i den domänspecifika modelleringen.

### 5.2 Domänspecifik modellering (DSM)

Ett *domänspecifikt språk* är ett modelleringspråk med en begränsad uttryckskraft anpassad för en smal domän, så som en viss produktlinje i en viss organisation [50]. Domänspecifika språk används framförallt som gränssnitt till specialiserade ramverk eller komponent-bibliotek; varje modell kompileras till



Tabell 5.1: Domänspecifik modell [48]

en fil som konfigurerar ramverket. Ramverket utvecklas ofta tillsammans med det domänspecifika språket, till exempel genom att anpassa och specialisera redan existerande ramverk.

**Exempel** Med ett domänspecifikt språk för mobil-appar kan apparna specificeras i termer av *query*, *pop-up*, *SMS* och andra begrepp specifika för mobilplattformen. Slutprodukten, dvs körbar källkod (figur 5.2), genereras automatiskt från en hög-nivå design (figur 5.1).

**Exempel** Ett domänspecifikt språk för C2-applikationer skulle på motsvarande vis kunna tänkas innehålla prefabricerade konstruktioner för *HeadQuarter*, *MilitaryUnit*, *AirForce*, *MaritimeForces*, *Operations*, *Mission*, etc. samt kodgeneratorer som skapar uppdragsscenarier.

Domänspecifika modelleringspråk tillåter således utvecklaren att fokusera på att specificera den avsedda funktionaliteten snarare än omständliga implementationsdetaljer. Den automatiserade kodgenereringen är möjlig just därför att modelleringspråket är specialiserat för en smal problemdomän.

```

    Say 0x07 2 'select...'
    Say 0x08 3 '...another...'
    Say 0x04 1 '...mode...'
    Load A 00
:10_663
    Add A 01
    Say 0x09 1 'for...'
    SayMode A
    Say 0x00 1 '...press...'
    SayNumber A
    Test A < 05
    If
    Jump 10_663
:10_222
    Say 0x17 11 'The current temperature is ... degrees'
    Say 0x16 11 'The air filtration system is in ... mode'
    Say 'The cooling system is in ... mode'
    Say 'The heating system is in ... mode'
    Say 'Press 1 to repeat or 2 to return to main menu'
:10_187
    SaveDTMF A
    SetMode A
    Jump 10_774

```

Tabell 5.2: Genererad källkod [48]

### Domänspecifika språk är lättare att modellera med

Domänspecifika språk är mer lättanvända i och med att de har en högre och mer naturlig abstraktionsnivå [50, 51, 48]. Språkkonstruktionerna är hämtade direkt ur en konkret domän och följer därmed en befintlig informell eller semi-formell terminologi.

Fokus på en smal domän så som en produktlinje innebär dessutom att det räcker med små, enkla modeller för att fullständigt specificera en lösning – det mesta i lösningen kan lämnas implicit och endast variationerna (så som det karakteristiska för en viss produkt inom en produktlinje) behöver anges explicit i modellen.

### Domänspecifika språk är lättare för intressenter att förstå

Domänspecifika modelleringsspråk innehåller inte konstruktioner och detaljer som är orelaterade till problemdomänen, till exempel låg-nivå generiska konstruktioner som *Class*, *Attribute* och *Method*. Den mer domännära abstraktionsnivån underlättar för intressenter – domän-expertter, beställare, kunder, slutanvändare, etc. – att läsa, förstå och modifiera modeller och därmed

engageras i utvecklingsprocessen och bättre förmedla krav [50, 48].

### Domänspecifika språk genererar komplett kod

Kodgeneratorer för generella modelleringsspråk producerar endast kodstubbar som sedan måste editeras manuellt. Utvecklaren tvingas därmed att hantera samma information på två platser, i koden och i modellen, en notoriskt besvärlig uppgift och en viktig anledning till att systemutvecklare upplever modellering som ett störande extramoment [48].

Med DSM å andra sidan genereras körbar och fullständig kod direkt ur modellerna. Om kodgeneratoren är skriven av en expert kan man rent av vänta sig att den genererade koden är av bättre kvalitet än vad en genomsnittlig programmerare skulle åstadkomma manuellt [48].

Kodgeneratoren behöver inte heller alltid skapas manuellt. Om ramverket/kodbiblioteket har ett väldefinierat gränssnitt kan en generisk syntetiseringsalgoritm – en algoritm som syntetiserar en optimal konfiguration av komponenter givet komponenternas gränssnitt – fungera som kodgenerator (se avsnitt 5.3.3).

### Domänspecifika språk möjliggör automatiserad analys

Tack vare den smala problemdomänen och den begränsade uttryckskraften i domänspecifika språk är det möjligt att generera *optimerad simuleringskod*<sup>1</sup> direkt ur *enkla modeller* på hög abstraktionsnivå, inte bara simuleringsstubbar som måste editeras manuellt [48, 52]. Eftersom simuleringsverktyg ofta är extremt resurskrävande kan det vara avgörande att simuleringskoden just är anpassad och optimerad för den specifika frågeställningen [52].

Den smala problemdomänen tillåter dessutom *business rules* (domänregler) som mer effektivt utesluter otillåten eller oönskad design [48]. Ju smalare domän, desto starkare och mer precisa domänregler kan formuleras, vilket i sin tur innebär att fler designfel fångas upp (se avsnitt 5.3.1).

System som modelleras med domänspecifika språk kan sålunda analyseras och i viss mån verifieras redan på designstadiet då specifikationerna (det vill säga modellerna) formuleras. Detta är av stort värde eftersom att det är mindre kostsamt att rätta till designfel ju tidigare de upptäcks.

## 5.3 Automatiska analysmetoder

På ett abstrakt plan är domänspecifika språk formella system, ofta variationer på välbekanta formella system från bla automatateori, processalgebra och logik. Naturligt nog lånar därför DSM många analysmetoder just från formella metoder. I detta avsnitt tittar vi lite närmare på några av dessa.

<sup>1</sup>Med 'simuleringskod' avses här input till alla slags verktyg som analyserar icke-funktionella egenskaper.

### 5.3.1 Ontologi och automatisk inferens

Ett domänspecifikt språk formaliserar en domänspecifik begreppsapparat och kan därför betraktas som en *ontologi*, det vill säga en explicit, formell specifikation av en gemensam begreppsbildning [53].

Ontologier är 'formella' i den betydelsen att notationen följer en formell syntax, och 'explicita' i det att den avsedda tolkningen (meningen) reflekteras i domänsregler. Den formella syntaxen gör det möjligt för maskiner att *läsa* modeller som uttrycks med ontologin, medan domänreglerna gör att maskinerna kan *förstå* modellerna och dra slutsatser ur dem.

Syntaxen är oftast någon variation på entitet-attribut-relation (EAR) modellering, en populär form av objektorienterad modellering som är naturlig och lätt att ta till sig för personer med erfarenhet av objektorienterad programmering. Domänreglerna anger hur nya entiteter, attribut och relationer kan härledas ur redan fastställda sådana.

**Exempel** En ontologi för sociala relationer med entiteterna `Ungkarl` och `Man` samt relationen `giftMed` kunde till exempel innehålla domänreglerna:

$$\begin{aligned} x \text{ är } \text{Ungkarl} &\longrightarrow x \text{ är } \text{Man} \\ x \text{ är } \text{Ungkarl} &\longrightarrow x \text{ är inte } \text{giftMed något } y \end{aligned}$$

Domänreglerna ovan speglar den avsedda tolkningen av `Ungkarl`, `Man` och `giftMed` i det att reglerna är giltiga för all tänkbar instansdata, det vill säga giltiga vilka instanser vi än substituerar för  $x$ .

I DSM används domänreglerna till att automatiskt verifiera att en modell är förenlig med den avsedda betydelsen av entiteter, attribut och relationer.

**Exempel** Den andra domänregeln i exemplet ovan till exempel skulle varningsflagga om en modell sade:

$$\begin{aligned} \text{adam} &\text{ är } \text{Ungkarl} \\ \text{adam} &\text{ är } \text{giftMed } \text{eva} \end{aligned}$$

Domänregler kan även användas för att automatiskt klassificera objekt, fylla i attributvärden (datafält) samt koppla samman objekt med relationer.

**Exempel** Den första domänregeln i exemplet ovan skulle automatiskt klassificera `adam` som en `man` ifall modellaren anger att `adam` är en `ungkarl`, dvs automatiskt härleda `adam är Man` ur `adam är Ungkarl`.

Domänregler uttrycks oftast med något fragment av första ordningens predikatlogik – hela första ordningens predikatlogik är allt för komplext för att tillåta automatisk inferens med rimlig prestanda. Inom ontologiforskningen har man länge arbetat på att identifiera fragment som hittar den rätta balansen (för något visst syfte) mellan uttryckskraft och prestanda samt som har en enkel och naturlig notation.

### 5.3.2 Model checking: automatisk tillståndsrymdssökning

Dynamik specificeras i de flesta modelleringsspråk – inkluderat MODAF – med tillståndsmaskiner, aktivitetsdiagram och andra variationer på klassiska automater. En *model checker* är ett verktyg som automatiskt verifierar att en tillståndsmaskin (eller annat dynamiskt system) inte kan uppvisa oönskat beteende så som att en fil läses innan den öppnas, eller att två trådar samtidigt får skrivrättigheter till samma fil [54]. Model checkern utforskar alla logiskt möjliga exekveringar av tillståndsmaskinen, vilket gör verifikationen ekvivalent med uttömmande testning. Model checkern returnerar en exekvering som uppvisar det oönskade beteendet ifall en sådan exekvering finns, vilket är värdefullt vid felsökning.

Model checking verktyg används i stor omfattning i hårdvaruindustrin [55]. Fram till nyligen har dock model checking av *mjukvarusystem* inte vunnit framgång inom industrin. Det viktigaste hindret har varit det merarbete som krävts för att skapa modeller som kan analyseras av en model checker – en model checker tar inte källkod direkt som input utan snarare abstrakta modeller. Utvecklarna har varit tvungna att manuellt konstruera modeller (abstraktioner av källkoden) i det inputspråk som model checkern förstår, vilket har inneburit ett omfattande merarbete: så fort källkoden uppdateras lite måste modellen uppdateras manuellt för att förbli synkroniserad.

Detta hinder försvinner med DSM (se figur 5.3). Modellerna som skapas med DSM kan automatiskt kompileras till ett format som förstås av en model checker; modellern skapar de modeller som han eller hon ändå skulle skapa och behöver bara trycka på en knapp för att få verifikationsresultatet. Inget merarbete läggs på modellern.

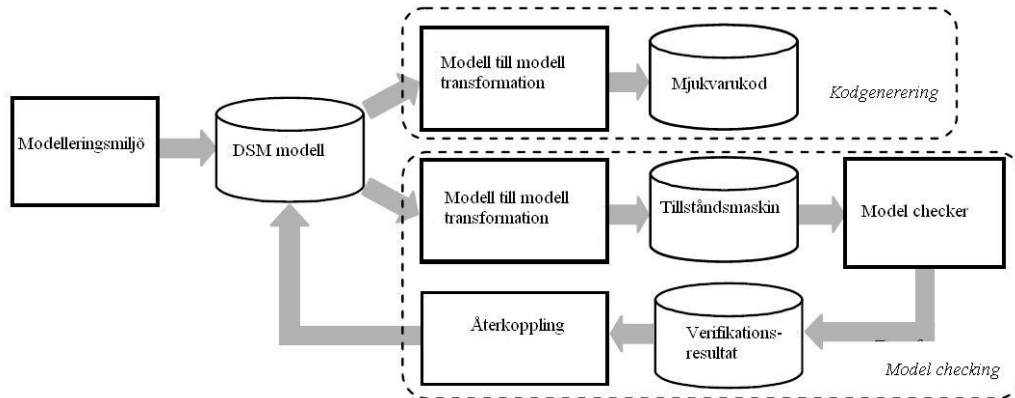
#### Jämförelse med simulering

Model checking kan ses som en form av uttömmande simulering i vilken samtliga möjliga exekveringar undersöks, inte bara en begränsad delmängd som slumpas fram. En model checker kan därmed fånga fel i obskyra exekveringar som sannolikt inte skulle dyka upp ens vid omfattande testning [55], [52].

Traditionell simulering (så som diskret händelsestyrd simulering) kan endast verifiera rent probabilistiska modeller och passar därför ofta inte riktigt för att analysera icke-deterministiska system, det vill säga system med delvis obestämt beteende. En model checker däremot kan verifiera även icke-deterministiska modeller.

### 5.3.3 Automatisk planering

DSM kan ses som en form av automatisk programmering: utvecklaren specificerar den avsedda funktionaliteten och implementationen – i form av en konkret konfiguration av det bakomliggande kodbiblioteket – genereras sedan automatiskt ur specifikationen. Kodgeneratorer i DSM lösningar



Tabell 5.3: Model checking kräver inget extra modelleringsarbete under DSM

använder ibland planeringsalgoritmer från AI för att mappa specifikationer till implementationer, det vill säga resonera sig fram till en konfigurationsfil som på ett optimalt sätt uppfyller specifikationen [56].

Problemet att automatiskt syntetisera en optimal orkestrering av befintliga komponenter (tjänster, processer, objekt, procedurer, etc.) som tillsammans levererar ett önskat beteende är ett klassiskt problem inom AI [57]. Den automatiska syntetiseringen förutsätter dock att varje komponent i biblioteket har ett formaliserat gränssnitt, till exempel i form av en tillståndsmaskin, som specificerar komponentens observerbara beteende.

## 5.4 Domänspecifik MODAF

Domänspecifik modellering hanterar komplexitet genom att bygga mindre, avgränsade och specifika modelleringsspråk, som därigenom blir mer begripliga och användbara. EA hanterar typiskt komplexitet genom att abstrahera och förenkla bort ”onödiga” lågnivåaspekter, vilket möjliggör företagsövergripande arkitekturbeskrivningar där de viktigaste dragen på hög abstraktionsnivå framträder. Det kan tyckas som att detta är två rakt motsatta strategier. Hur kan de förenas?

I detta avsnitt visar vi hur DSM kan rymmas inom ramen för MODAF (avsnitt 5.4.1 och avsnitt 5.4.2) och skissar på hur DSM kan lyftas från mjukvarumodellering till verksamhetsmodellering (avsnitt 5.4.3).



### 5.4.1 MODAF-profiler

UML innehåller en profileringsmekanism som är till för att definiera domän-specifika språk baserade på UML. När domän-specifika språk baseras på UML blir det möjligt att återanvända UML-verktyg, expertis och modeller i den domän-specifika modelleringen [58].

En *UML-profil* kan ses som en ontologi som specialiserar UML. Varje profil inför en samling nya begrepp (entiteter, attribut och relationer) som specialiserar begrepp från UML eller från redan befintliga profiler. Profiler kan och bör även införa domänregler (i form av villkor uttryckta i OCL, *Object Constraint Language*) som preciserar meningen hos de nya, specialiserade begreppen [58].

Relativt UML utgör MODAF i sig en specialisering till försvarsdomänen – MODAF är i själva verket definierat som en UML-profil (se avsnitt 2.4.1). Samma utbyggnadsmekanism kan utnyttjas till att skapa mer riktade domän-specifika språk inom ramen för MODAF i form av MODAF-profiler, det vill säga UML-profiler som specialiserar MODAF. Vi illustrerar med ett litet exempel.

**Exempel** Vi skapar en MODAF-profil som inför relationerna **hosts** och **has subsystem** som specialiseringar av den mer generiska MODAF relationen **has part**. Den avsedda tolkningen av de nya relationerna preciseras med ett antal domänregler. Först två domänregler som säger att **hosts** är en relation mellan plattformar och system:

$$\begin{aligned}x \text{ hosts } y &\longrightarrow x \text{ is Platform} \\x \text{ hosts } y &\longrightarrow y \text{ is System}\end{aligned}$$

Två liknande regler sedan som säger att **has sub-system** är en relation mellan system:

$$\begin{aligned}x \text{ has sub-system } y &\longrightarrow x \text{ is System} \\x \text{ has sub-system } y &\longrightarrow y \text{ is System}\end{aligned}$$

Slutligen en regel som säger att **has part** är en transitiv relation:

$$x \text{ has part } y, y \text{ has part } z \longrightarrow x \text{ has part } z \quad (5.1)$$

Genom att formulera domän-specifika språk som MODAF-profiler bevaras interoperabiliteten med MODAF och det blir möjligt att återanvända verktyg, expertis och modeller. Följande exempel illustrerar hur domän-specifika vyer synkroniseras automatiskt med generiska MODAF-vyer.

**Exempel** Antag att vi under den specialiserade MODAF-profilen från exemplet ovan anger att plattformen *Airframe-1* är värd för systemet *Sensor-1*

som innehåller ett delsystem *Scanner-1*:

$$\textit{Airframe-1} \text{ hosts } \textit{Sensor-1} \quad (5.2)$$

$$\textit{Sensor-1} \text{ has sub-system } \textit{Scanner-1} \quad (5.3)$$

Om vi byter till en generisk MODAF vy (som inkluderar relationen **has part**) ser vi att plattformen *Airframe-1* innehåller systemet *Scanner-1* som del:

$$\textit{Airframe-1} \text{ has part } \textit{Scanner-1} \quad (5.4)$$

Sambandet 5.4 i den generiska MODAF-vyn härleds automatiskt via domän-reglerna i exemplet ovan ur sambanden 5.2 och 5.3 som lades in under den specialiserade MODAF-profilen. Om vi försöker att plocka bort sambandet 5.4 varnas vi för att detta strider mot data 5.2 och 5.3 som lagts in under den specialiserade profilen.

#### 5.4.2 Arbetsdelning mellan MODAF och MODAF-profiler

DSM kan således bedrivas inom ramen för MODAF med automatisk synkronisering mellan domänspecifika modeller och MODAF-vyer. Men hur ska arbetsfördelningen se ut mellan domänspecifik modellering (i MODAF-profiler) och verksamhetsmodellering (i standard MODAF)?

[59] beskriver hur man kan bygga företagsövergripande arkitekturbeskrivningar utifrån mindre domänspecifika modeller. Det grundläggande problemet är korsberoenden mellan olika domäner: processer på affärssidan ska realisera företagets produkter, system på mjukvarusidan ska stödja dessa affärsprocesser, och det krävs teknisk infrastruktur för att köra mjukvaran. Systemen behandlar den information som processerna skapar. Arkitekter som arbetar med dessa olika områden måste kommunicera med varandra. Den kommunikationen måste inte bestå i att var och en använder sina fullständiga domänspecifika modeller. Vad som behövs är någon form av dokumentation av korsberoendena, uttryckta i ett språk som blir en minsta gemensam nämnare. Först när begränsningarna från övriga områden är dokumenterade och klarlagda har var och en frihet att utforma sina lösningar inom dessa ramar.

Utifrån detta resonemang drar Lankhorst et al. slutsatsen att ett språk för att modellera verksamhetsövergripande *enterprise*-arkitekturer bör fokusera på relationerna mellan olika domäner. Specifikt är det två beskrivningsområden som ska täckas [59]:

- Den globala strukturen inom vare domän, med en beskrivning av de *huvudsakliga* delarna och deras beroenden, på ett sätt som är *lätt att förstå* även för icke-expertter inom den aktuella domänen.
- Relationerna *mellan* domäner.

Om det sker en arbetsdelning så att övergripande EA-modellering sker med ett språk med hög abstraktionsnivå enligt dessa principer – MODAF – medan detaljerna inom varje domän hanteras av olika domänspecifika språk – i form av MODAF-profiler – så försvinner den skenbara motsättningen mellan domänspecifik modellering och EA.

### 5.4.3 MODAF-profiler för verksamhetsmodellering

I avsnitt 5.4.2 antog vi att verksamhetsmodelleringen sker med MODAF. Men skulle man kunna tänka sig att även verksamhetsmodelleringen sker med MODAF-profiler? Verksamhetsmodellering inom FM kan tjäna två olika syften:

1. dokumentation/konfigurationskontroll av befintliga förmågor för exempelvis budgetarbete eller
2. upphandling/kravställning/utveckling av framtida förmågor.

Verksamhetsmodellering som syftar till dokumentation (syfte 1 ovan) kunde lämpligen ske med en för FM gemensam MODAF-profil som enbart preciserar redan existerande MODAF-begrepp – som ju avsiktligt lämnats vaga i MODAF-standarderna – med hjälp av domänregler som ringar in den av FM föreskrivna tolkningen. Detta skulle göra det möjligt att automatiskt konsistenskontrollera MODAF-modeller i UML-kompatibla modelleringsmiljöer.<sup>2</sup>

För verksamhetsmodellering som syftar till kravställning (syfte 2 ovan) kunde ytterligare ett steg mot domänspecialisering vara lämpligt: en MODAF-profil som är anpassad för verksamhetsmodellering inom ett enskilt förmåge-utvecklingsprojekt och hämtar begrepp och business rules (domänregler) direkt ur domänen för förmågan som skall utvecklas. Den domännära abstraktionsnivån kan förväntas underlätta för beställare och andra intressenter att förmedla krav och engageras i utvecklingsprocessen.

De förmågespecifika MODAF-profilerna skulle även kunna fungera som gränssnitt till simuleringsramverk – att använda verksamhetsmodeller som indata till simuleringar på operativ/taktisk nivå är ju redan en populär tanke (se kapitel 4, 5, 7 och 8 i denna rapport). Avgränsningen till en specifik förmåga gör att simuleringen av verksamhetsmodeller inte kräver något merarbete av verksamhetsarkitekten: det blir möjligt att generera komplett simuleringskod direkt ur verksamhetsmodeller som inte behöver vara mer komplicerade och detaljerade än syftet med verksamhetsmodelleringen kräver – simuleringen ställer inte extra krav på detaljer.

<sup>2</sup>Inom ontologier är det best practice att specialisera ('subklassa') ett vagt begrepp innan dess mening preciseras med hjälp av domänregler. Enligt denna best practice skulle MODAF-profilen specialisera standardbegrepp ur MODAF och sedan formulera domänreglerna på de specialiserade begreppen.

## 5.5 Sammanfattning

MODAF är ett generellt modelleringspråk avsett för att modellera allt inom försvarsdomänen. I likhet med andra generella modelleringspråk kritiseras MODAF för att dess modeller är svåra att utveckla, svåra att förstå, svåra att transformera och svåra att analysera.

Domänspecifika språk – modelleringspråk med en begränsad uttryckskraft anpassad för en smal domän, såsom en viss produktlinje i en enskild organisation – drabbas däremot betydligt mindre av denna typ av kritik. Ledande industriföretag (Microsoft, Nokia, Motorola, JetBrains, med flera) satsar nu stort på domänspecifika språk.

I det här kapitlet har vi visat hur domänspecifik modellering kan rymmas inom ramen för MODAF på ett sätt som gör det möjligt att återanvända MODAF-verktyg, expertis och modeller.

Vi har även skissat på hur domänspecifik modellering kan lyftas till verksamhetsmodellering, till exempel i form av en MODAF-profil som endast preciserar redan existerande MODAF-begrepp – som ju avsiktligt lämnats vaga i MODAF-standarden. En sådan MODAF-profil skulle göra det möjligt att automatiskt konsistenskontrollera MODAF-modeller med UML-baserade standardverktyg.



## 6 Analyismetoder för MBFU

### 6.1 Introduktion

Enterprise Architecture (EA) kan användas i många syften [60]. Dokumentation (av *as-is*-läge) och kommunikation (av *as-is*- eller *to-be*-läge) tillhör de vanligaste. Emellertid kan en mogen organisation använda EA till mer än så. Ett område som fått ökad uppmärksamhet på senare år är EA-analys, det vill säga användning av den information som finns lagrad i en arkitekturbeskrivning som indata till analysmodeller i syfte att stödja beslutsfattande [3].

Traditionellt EA-arbete inskränker sig ofta till att man gör modeller som olika intressenter kan titta på och därmed bilda sig en uppfattning om olika sakförhållanden. En sådan gemensam lägesuppfattning är mycket värd och ska på intet sätt nedvärderas. EA-analys går dock ett steg längre och handlar om att behandla och visualisera informationen i arkitekturbeskrivningen på ett sådant sätt att även icke-triviala egenskaper, som inte går att bedöma vid första anblicken, framkommer. Därmed skulle EA bli ett mycket mer kraftfullt verktyg för beslutsstöd än vad det är i dag.

### 6.2 Bakgrund

Enterprise Architecture (EA) har under de senaste decennierna vuxit fram som en metod för att hantera komplexiteten i stora organisationer. EA gör anspråk på att täcka både (i) affärsverksamhetens strategier, processer och mål och (ii) de informationssystem som stödjer dessa. Kärnan i metoden är så kallade arkitekturmodeller; diagram som visar hur olika delar av organisationen hänger ihop med varandra i termer av till exempel dataflöden, processbeskrivningar, ekonomiska transaktioner, ansvar och befogenheter etc.

Som framgår av kapitel 2 finns det gott om etablerade ramverk för EA. Inget av dem stödjer emellertid kvalificerad analys. För att förklara detta ska vi betrakta arkitektur-analogin lite närmare.

### 6.3 Arkitektur vs. arkitektur

Begreppet *enterprise architecture* innebär en analogi mellan organisationer ("arkitekturens" nya domän) och byggnader (arkitekturens traditionella domän). Professor Pontus Johnson har analyserat skillnaderna och likheterna mellan dessa arkitektur-begrepp närmare [61]. Han identifierade en mognadsskala för arkitekturritning:

1. **As-is-arkitektur:** Beskriva den redan byggda konstruktionen. Syftet med en sådan ritning är huvudsakligen att hjälpa användaren förstå hur

konstruktionen kan nyttjas.

2. **To-be-arkitektur:** En ritning av den planerade konstruktionen. Syftet med en sådan ritning skiljer sig från as-is-arkitekturen. Fokus här är inte så mycket användning utan snarare design: Ritningen hjälper arkitekten att tänka kring sin konstruktion.
3. **Governance:** Styra konstruktionsprojektet. Syftet med Governance skiljer sig från de två tidigare stegen. Fokus här är inte på vare sig användning eller design utan på implementation: Ritningen styr byggprojektet.
4. **Modelleringspråk:** Standardiserade representationsmanér. Syftet med modelleringspråk är att säkerställa att modellen ger en riktig bild av verkligheten, oavsett vem som skapat modellen och oavsett vem som betraktar den. Detta åstadkoms genom att en standardiserad regeluppsättning används för att skapa modellen.
5. **Modelleringsramverk:** Standardiserade ritningsuppsättningar. Syftet med dessa är att hjälpa modellaren att representera de relevanta aspekterna av sin design.

Så här långt har Enterprise Architecture också kommit, men byggnadsarkitekturen har tagit ytterligare ett mycket viktigt steg:

6. **Arkitekturanalys**, vars syfte är att förutsäga konstruktionens beteende redan under designfasen.



(a) Misslyckad traditionell arkitektur.



(b) Misslyckad enterprise architecture.

Figur 6.1: En analogi.

### 6.3.1 Arkitekturanalys

Arkitekturanalys i den traditionella arkitekturen handlar bland annat om mekanik, hållfasthet, akustik och optik. Möjligheten att kunna göra sådana analyser har revolutionerat den traditionella arkitekturritningen. Med dagens verktyg för CAD (computer aided design) och CAE (computer aided engineering) kan man redan på ritbordet testa viktiga egenskaper hos slutkonstruktionen. Håller bron för trafiken? Faller ljuset vackert i vardagsrummet? Hör publiken längst bak i salen vad som sägs från podiet?

Det är viktigt att inse att sådana frågor *inte* kan besvaras av arkitekturritning som befinner sig på de någon av de fem första mognadsnivåerna på Johnsons skala. På dessa mognadsnivåer finns det förvisso kunskap om sådant, men det är tyst kunskap (*tacit knowledge*) som erfarna byggmästare har i huvudet – inte formaliserad kunskap som låter sig uttryckas explicit eller implementeras i verktyg för automatisk uträkning. En duktig arkitekt eller byggnadsingenjör kan se på en ritning om huset kommer att rasa i en storm som i Figur 6.1a, men det är först med CAE-verktyg som man automatiskt under designfasen kan räkna på hur tjocka balkar och bärande väggar man behöver ha. På samma sätt är det i stor utsträckning inom enterprise architecture idag. Kanske kan en duktig arkitekt se på en arkitekturritning gjord enligt Zachman, TOGAF, MODAF eller DoDAF huruvida en internetbank är stabil eller kommer att krascha som i Figur 6.1b, men redan det är tveksamt. Motsvarigheten till CAE – automatiserad analys av huruvida systemen och verksamheten kommer att bli effektiva, om informationen kommer att vara säker, i vilken utsträckning systemen kommer att vara tillgängliga, hur interoperabla och flexibla systemen blir – är det inte tal om i dagens ledande arkitekturmodelleringspråk.

Den potentiella nyttan av sådan analys är emellertid oomtvistlig. Därför finns det en del lovande forskning kring modeller och metoder för att möjliggöra EA-analys och därmed lyfta Enterprise Architecture till den sjätte mognadsnivån.

## 6.4 Vikten av mål

En viktig aspekt av EA-analys är dess fokus på beslutsstöd och mål. En *as-ís*-beskrivning av en arkitektur kan man göra ganska mekaniskt, utan att reflektera över vad man egentligen ska ha arkitekturbeskrivningen till efteråt. För att kunna göra en bra EA-analys krävs emellertid lite mer. På samma sätt som det krävs olika information för att analysera hållfasthet, ljussättning eller akustik i en byggnad så krävs det olika information för att analysera affärsnytta, informationssäkerhet eller tillgänglighet i en enterprise architecture.

Detta ställer krav på genomförandet av enterprise architecture-satsningar. För att lyckas måste sådana satsningar börja med att definiera tydliga mål – vad är det som ska analyseras, vilka beslut är det egentligen som arkitekturbeskrivningarna ska stödja? Utan en sådan tydlig målbild från första



början blir det omöjligt att gå iland med projektet. Då sitter man med fel sorts beskrivningar för den analys man vill göra – en lika omöjlig uppgift som om man vill göra en elsäkerhetsanalys, men har tagit fram en ritning över fastighetens avloppsledningar.

En organisation bör emellertid inte bara se detta krav som en pålaga: det är också en viktig generell framgångsfaktor. Många EA-satsningar har satts igång med oklara mål, gjort av med sin tid och budget och sedan inte riktigt kunnat förklara nyttan med den framtagna *as-is*-arkitekturen. Om klara analysmål sätts upp från början – och stäms av med sina avnämare – så har EA klart större chanser att överleva och ses som värdeskapande i organisationen.

## 6.5 Behovet av att hantera osäkerhet

EA-analys skiljer sig från traditionell EA bland annat genom att den måste kunna hantera olika sorters *osäkerhet*. Alltför ofta tas det bara för givet att de modeller som man bygger är korrekta och tillförlitliga till 100%. Verkligheten är emellertid mer komplex. Den vetenskapliga litteraturen har identifierat åtminstone fyra sorters osäkerhet som måste hanteras för framgångsrik EA-analys [3]:

**Definitionsosäkerhet:** Det är inte alltid självklart vad som ska analyseras. Om man är intresserad av exempelvis tillgänglighet, är det då server-tillgänglighet, tjänstetillgänglighet, användarupplevd tillgänglighet eller något annat som avses?

**Heterogen teoribas:** Det är inte alltid självklart vilka teorier som ska användas för att genomföra analysen. Om man är intresserad av exempelvis tillgänglighet så kan man, med stöd hos forskare och praktiker, använda felträd, Failure Modes and Effects Analysis (FMEA), mögnadsmodeller eller andra metoder.

**Kausal osäkerhet:** I vissa vetenskaper finns det otvetydiga teorier som förutsäger exempelvis hur mycket en balk böjs ned när den utsätts för en viss last. I EA-analys finns det ofta så många olika faktorer som påverkar resultatet att även bra teorier är behäftade med osäkerhet. Svaret på frågan om hur mycket tillgängligheten höjs av en bättre change management-process lämpar sig för att formuleras som en sannolikhetsfördelning snarare än ett enda deterministiskt svar.

**Empirisk osäkerhet:** Något som ofta glöms bort är att en organisation inte har perfekt kunskap om sig själv. Det är få CIO:er förunnat att kunna svara på exakt hur många servrar man har, eller om det finns någon affärsprocess som fortfarande använder ett visst (gammalt) system. Detta betyder i sin tur att arkitekturmodeller kommer att vara imperfekta representationer av verkligheten – vissa fel kommer ofrånkomligen att smyga sig in. Det finns forskning som syftar till att ta fram en optimal

samplings-frekvens för hur ofta modeller borde uppdateras, givet hur långa livscyklerna för IT-system är [62].

Hanteringen av dessa sorters osäkerhet begränsar vilka formalismer för analys som är lämpliga för EA-analys.

## 6.6 Formalismer för analys

Det finns några olika förslag på formalismer lämpliga för EA-analys i litteraturen. En gemensam nämnare är att de använder sannolikheter för att kunna hantera osäkerheterna ovan.

### 6.6.1 Probabilistiska relationsmodeller

Så kallade probabilistiska relationsmodeller (PRM) [63] har tagits fram av en forskargrupp vid Stanford och har tillämpats inom EA-analys av en forskargrupp vid KTH [64, 65, 66].

I en PRM kompletteras den traditionella arkitekturmodellen med en matematisk sannolikhetsmodell. Denna sannolikhetsmodell gör att man kan räkna på hur värdena hos intressanta attribut på entiteterna påverkas av arkitekturmodellens utseende. I den matematiska sannolikhetsmodellen kan man exempelvis koda in kunskap om hur prestandan hos en IT-tjänst påverkas av antalet servrar, antalet processer som använder tjänsten och redundansen i kommunikationsinfrastrukturen. Exakt hur dessa samband ser ut kan vara en forskningsfråga i sig – men när det sambanden väl är klarlagda och inkodade i sannolikhetsmodellen så kan arkitekten utföra prestanda-analys utan att vara prestanda-expert.

#### 6.6.1.1 P<sup>2</sup>AMF

Predictive, Probabilistic Architecture Modeling Framework (P<sup>2</sup>AMF) liknar PRM i det att man även här kompletterar arkitekturmodellen med matematiskt beskrivna sannolikhetsmodeller. Syftet är också likartat – formalismen ska möjliggöra analys av komplexa systemegenskaper hos arkitekturen. Skillnaden är att P<sup>2</sup>AMF bygger på Object Constraint Language (OCL), ett formellt språk som typiskt används för att uttrycka villkor på UML-modeller [67]. Sammanfattningsvis erbjuder P<sup>2</sup>AMF ett kraftfullt ramverk för EA-analys genom att (i) vara objektorienterat, hantera (ii) predikatlogik, (iii) aritmetik och (iv) mängdlära, samt osäkerhet på (v) objekt- och (vi) klassnivå.

## 6.7 Tillämpningsområden

EA-analys, baserad på formalismerna ovan, har tillämpats på ett flertal områden:

### 6.7.1 Analys av systemegenskaper

**Säkerhet** för industriella styrsystem (cyber security) har analyserats med hjälp av PRM-formalismen [66].

**Datakvalitet** har analyserats med hjälp av PRM-formalismen [68].

**Tillgänglighet** har analyserats med hjälp av PRM-formalismen [69] och P2AMF-formalismen [70].

**Interoperabilitet** har analyserats med hjälp av P2AMF-formalismen [71]

**Modifierbarhet** har analyserats med hjälp av PRM-formalismen [64, 65]

### 6.7.2 Analys av militära förmågor

I den vetenskapliga litteraturen finns det också exempel på EA-analys i militär kontext. I [72] används formalismerna bayesianska nätverk och felträäd tillsammans med DoDAF-ramverket för att analysera organisationsöverskridande beroenden. Målet är att strukturerat och formellt sett analysera och synliggöra hur högnivåkoncept såsom operativa förmågor hänger ihop med tekniska lösningar såsom IT-system och fordon. Artikeln innehåller en metod för hur formella modeller baserade på bayesianska nätverk och felträäd kan tas fram ur DoDAF-modeller av en domänexpert och illustrerar metoden med hjälp av ett målinvisningsexempel.

En vidareutveckling och tillämpning av detta arbete återfinns i [73], som använder metoden att kombinera DoDAF med bayesianska nätverk för att ta fram kvantitativa tillförlitlighetsestimat för ett missilförsvarssystem. Artikeln använder modellen för att analysera systemets tillförlitlighet i olika scenarier.

I [74] beskrivs hur MODAF-ramverket kan utvidgas för att möjliggöra likartad formaliserad och spårbar analys av beroenden. Artikeln tar fram en metamodell – en delmängd av MODAF:s metamodell M3 – som lämpar sig för felträdsanalys och ger metodanvisningar för hur denna kan användas för att analysera beroenden i en arkitektur.

## 6.8 Verktygsstöd

Som framgår av kapitel 3 är de flesta kommersiella verktyg inte inriktade på EA-analys i den bemärkelse som begreppet används i detta kapitel. Det enda av de kommersiella verktygen som kan sägas ha god analysförmåga är Abacus (som är sprunget ur en akademisk forskningsmiljö). I Abacus kan man undersöka arkitekturmodeller med avseende på flera egenskaper såsom prestanda, flexibilitet och tillförlitlighet. I bakgrunden körs Monte-Carlo- och diskreta händelsesimuleringar. Vad Abacus inte förmår är att hantera osäkerhet (se ovan) på ett kvalificerat sätt.

För att fullt ut möjliggöra EA-analys enligt de principer som dras upp i detta kapitel pågår ett utvecklingsprojekt på KTH där ett verktyg –

Enterprise Architecture Analysis Tool (EA<sup>2</sup>T) – tas fram.<sup>1</sup> Målsättningen är att skapa ett kvalificerat verktyg som möjliggör arkitekturbaserat beslutsstöd för hanteringen av stora, organisationsövergripande informationssystem.

I EA<sup>2</sup>T finns de bägge formalismerna probabilistiska relationsmodeller och P<sup>2</sup>AMF implementerade. Olika versioner av verktyget har presenterats i ett flertal vetenskapliga publikationer [75, 76], testats i ett antal fallstudier (se även avsnitt 6.7 ovan) och till och med länkats ihop med nätverksscannern för att automatiskt skapa arkitekturbeskrivningar utan tröttsam manuell modellering. Verktyget illustreras i figur 6.2.

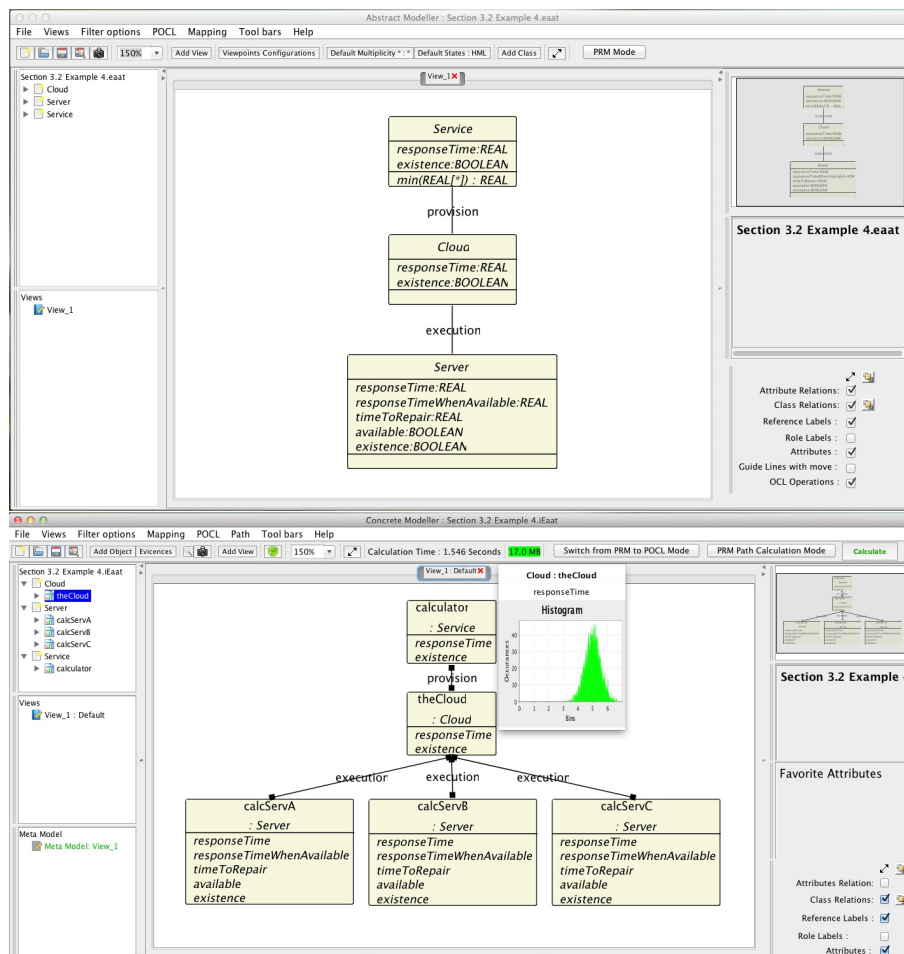
## 6.9 Sammanfattning

EA har potential att användas inte bara för beskrivning och kommunikation, utan även för analys. EA-analys karakteriseras av att icke-triviala egenskaper hos en arkitektur beräknas och presenteras som stöd för en beslutsfattare. En framgångsfaktor är väldefinierade beslutssituationer med väldefinierade mål som analysen ska stödja.

Även om EA-analys bland praktiker är i sin linda så finns det ett flertal exempel i forskningslitteraturen på ramverk och formalismer med stor potential. Fallstudier på företag och organisationer har varit framgångsrika, och litteraturen rymmer även exempel på analys i militär kontext. För PRM- och P<sup>2</sup>AMF-formalismerna finns det idag gratis verktygsstöd.

---

<sup>1</sup>Kan laddas ner på <http://www.ics.kth.se/eat>



Figur 6.2: EA<sup>2</sup>T-verktyget. Den abstrakta modelleraren (övre bilden) definierar en metamodell. Den konkreta modelleraren (nedre bilden) definierar instansmodeller. Notera hur osäkerhet illustreras i form av staplar i en fördelning i den nedre bilden.

## 7 Verksamhetssimuleringsmetoder för MBFU

### 7.1 Introduktion

Verksamheter kan ses som komplexa socio-tekniska system där ett antal sammankopplade, och förhoppningsvis välfungerande, komponenter interagerar med varandra för att uppnå verksamhetens strategiska målsättning [77]. Kombinationen av personal (med olika bakgrund, kultur och erfarenhet) med diverse tekniska system och processer gör det mycket svårt för verksamhetens ledning att identifiera vilka kritiska faktorer som påverkar verksamhetens strategiska målsättning. Alltför ofta genomförs förbättringsarbete i delkomponenter inom verksamheter utan att först analysera vilka konsekvenser och effekter, såväl positiva som negativa, som potentiellt kan uppstå för verksamheten i sin helhet [78]. Detta förfarande leder till att kostsamma problem, istället för att lösas, enbart ”flyttas runt” inom verksamheten och därmed förblir kostsamma.

Verksamhetssimulering, eller Enterprise Simulation (ES)<sup>1</sup>, är en metod som används för att analysera verksamhetens dynamiska egenskaper och beteende [80, 81, 77]. ES skiljer sig från Enterprise Architecture (EA) i den mån att EA modellerar verksamhetens statiska egenskaper i form av diagram, organisationshierarkier och flödesscheman medan ES fokuserar på dynamiska egenskaper som verksamhetens prestanda, skalbarhet och stabilitet. Det absolut främsta användningsområdet för ES är exekvering och analys av s.k. ”what if” scenarier för beslutsstöds- och optimeringsändamål. Nedan ges några motiverande exempel där ES kan appliceras:

- *Beslutsstöd*: Beslutsfattare kan avgöra om en åtgärd (till exempel förändring av en affärsprocess) skall införas genom att först simulera åtgärden i en virtuell miljö. Beslutsfattaren kan därefter analysera simuleringen för att säkerställa att den tänkta åtgärden faktiskt uppfyller önskad funktion på holistisk nivå och att den uppfylls till rätt kostnad.
- *Insikt*: Verksamheter är, som nämnts ovan, komplexa system som består av flera sammankopplade komponenter vars enskilda beteende i kombination med dess interaktion med andra komponenter ger upphov till verksamhetens holistiska beteende. I verksamheter där det holistiska beteendet är okänt kan ES användas för att ge insikt i systemet genom att besvara frågor som: Vilka parametrar (förmågor, processer, etc.) har

---

<sup>1</sup>ES är i sig inget nytt, redan på 50-talet simulerades dynamiken i industriella verksamheter [79].

stort inflytande på verksamhetens strategiska mål? Finns det parametrar som förändrar beteendet på ett oförklarligt sätt?

Man kan grovt dela upp ES i följande nivåer<sup>2</sup>: 1) operativ nivå; och 2) strategisk nivå. På den operativa nivån simuleras enskilda verksamhetskomponenter medan den strategiska nivån fokuserar på simulering av verksamheter i sin helhet. De allra flesta ES-implementationer är operativa, då det är relativt lätt att modellera isolerade komponenter med hjälp av traditionella modellerings- och simuleringstekniker som till exempel diskret-händelsestyrd simulering (DHS).

På strategisk nivå finns det färre implementationer. Detta beror till stor del på att verksamhetens holistiska beteende är svårdefinierat och i vissa fall okänt för modelleraren och/eller ämnesrådesexperten. För strategisk ES används normalt hybridlösningar där systemdynamik (SD) och DHS kombineras med agent-baserad (AB) modellering [2, 83].

I takt med att processorprestanda och minneskapacitet ökat i våra datorer har också möjligheterna att skapa bättre verksamhetssimuleringar ökat. Nya simuleringsverktyg och förfinade metoder har genom åren utvecklats av industriella och akademiska samfund. Vi kommer i detta kapitel presentera de vanligaste simuleringsmetoderna inom ES samt ge exempel på implementationer av dessa.

## 7.2 Simuleringsmetoder

Innan vi fortsätter med att beskriva de simuleringsmetoder som använts för simulering av verksamheter är det viktigt att understryka varför och när det är lämpligt att använda sig av simulering. Buckl et al. [81] och Glazner [2] har identifierat följande:

- ES bör användas när analys av verklig verksamhet är alltför kostsam, tids- och/eller resurskrävande.
- ES bör användas för att analysera verksamhetskonfigurationer för att så tidigt som möjligt identifiera oförutsedda eller oönskade effekter.
- ES bör användas för att analysera förändringar i verksamhetskonfigurationer som kan vara direkt oetiska att genomföra i verkligheten.

Generellt sett så bör man *inte* använda sig av simulering när de uppskattade kostnaderna för att utveckla modellerna överstiger nyttan med simuleringen. Man ska också vara medveten om att simulering i många fall inte kan användas för att generera punktprediktioner. Detta gäller främst vid simulering av

<sup>2</sup>Liknande uppdelning förekommer även i moderna EA ramverk som till exempel MODAF (Ministry of Defence Architecture Framework) där information om verksamheten är uppdelad i strategisk samt operativ nivå med flera [82].

komplexa system där beteenden är olinjära och kaotiska (små förändringar i systemets konfiguration genererar okontrollerbara följder/effekter). Här kan man i bästa fall använda simulering för att få *insikt* i systemets dynamik.

Det finns ett flertal modellerings- och simuleringsmetoder där de mest kända och välanvända metoderna inom ES inkluderar [2]:

- DHS: Diskret händelsestyrd simulering (se kapitel 7.2.1)
- SD: Systemdynamik (se kapitel 7.2.2)
- AB: Agent-baserad modellering och simulering (se kapitel 7.2.3)

En stor fördel med dessa metoder är att det finns färdiga, kommersiella och icke-kommersiella, simuleringsverktyg, till exempel AnyLogic [84] eller Repast [85], som kan användas för att snabbt skapa modeller, exekvera simuleringar samt för att analysera resultat. Ovannämnda metoder, främst DHS och SD, har nått en hög mognadsgrad och används redan flitigt inom industrin. AB är en metod vars idéer utvecklades redan på 40-talet men som inte applicerats i praktiken förrän på senare år eftersom metoden vanligtvis kräver stor beräknings- och minneskapacitet.

### 7.2.1 Diskret händelsestyrd simulering

Diskret händelsestyrd simulering (DHS) är en simuleringsmetod som används för att simulera flöden i verksamheter och system. Vanligtvis använder man DHS för att: 1) identifiera flaskhalsar i flöden; 2) för att optimera flöden; eller 3) för att stress-testa flöden. Utöver ovannämnda användningsområden så är DHS ett utmärkt verktyg för att kommunicera och beskriva dynamiken i till exempel en verksamhets affärsprocess.

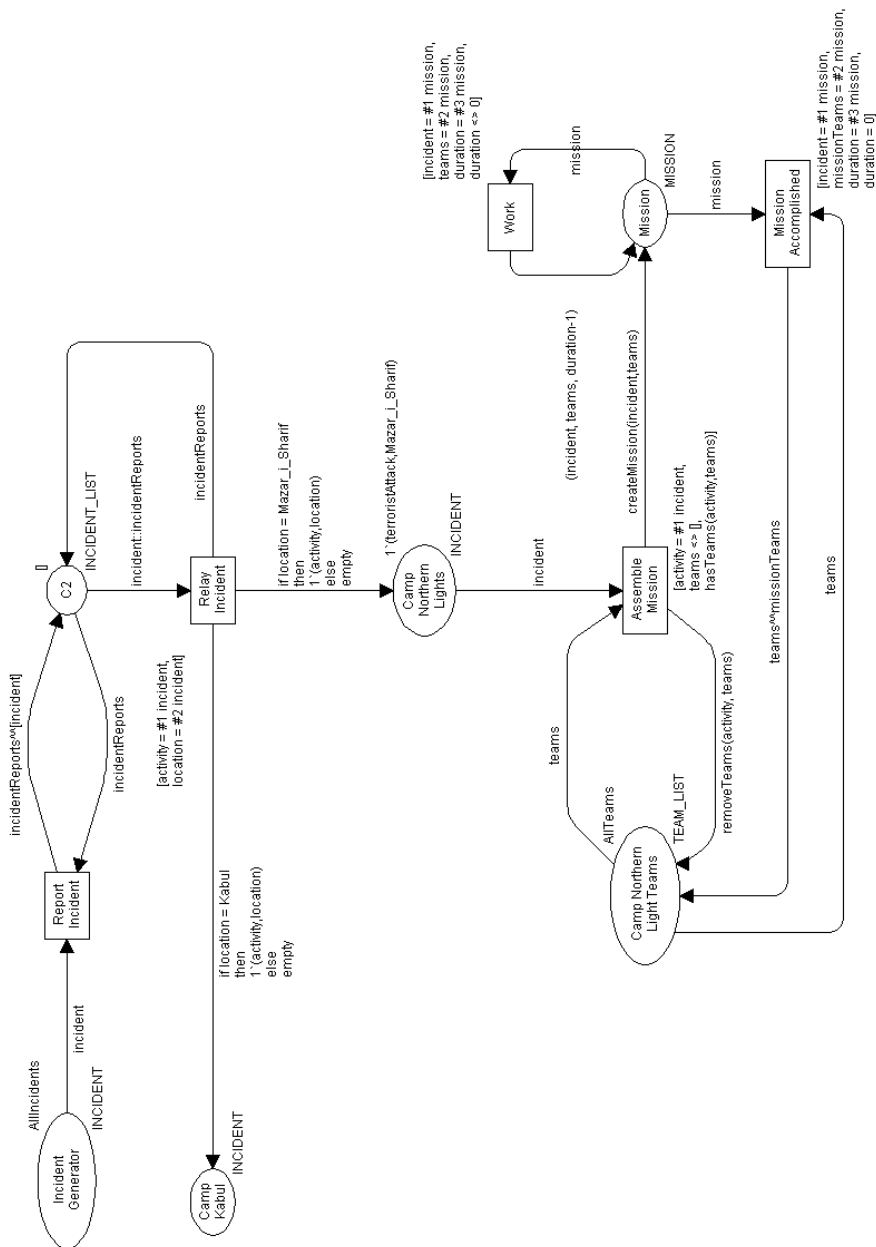
En DHS-modell består av ett antal entiteter som via transitioner/händelser överförs från ett tillstånd till ett annat. Ett entitetsflöde uppstår när flera transitioner tillåts exekvera i följd. Transitioner kan modelleras så att de aktiveras först när vissa villkor är uppfyllda. Transitionsvillkor kan vara regelbaserade (entiteten har de egenskaper/resurser som krävs), temporala (entiteten har existerat i tillståndet under en viss simulerings-tid) eller slumpmässigt styrda enligt någon känd sannolikhetsfördelning.

Ofta modellerar man DHS-system i grafer där noder representerar tillstånd för entiteterna och riktade länkar mellan noderna representerar möjliga transitioner. För att konkretisera resonemanget ovan visar vi i figur 7.1 en förenklad DHS modell, implementerad som ett "Colored Petri Net" (CPN) [86], där incidenter i form av terroristangrepp, illegal opiumhandel, demonstrationer, etc. hanteras av lokala enheter bestående av ett antal kompetenser (soldater, läkare, ingenjörer och så vidare). I CPN-diagrammet representerar ovaler (och cirklar) tillstånd där entiteter kan existera och riktade pilar, tillsammans med rektanglar, representerar transitioner mellan två tillstånd. Flödet i modellen är relativt enkelt att följa. Incidenter rapporteras till högkvarterets



ledningscentral (C2), som sedan vidarebefodrar incidenterna till lämplig lokal enhet. De lokala enheterna sätter sedan samman grupper av kompetenser som tillåts ingå i ett uppdrag där uppdraget går ut på att lösa de problem som orsakats av incidenten. Entiteter i DHS-modellen består alltså, i detta fall, av incidenter, kompetenser och uppdrag.

Modellen är parametriserad så att incidentflödet, tillgängliga kompetenser samt uppdragens kompetens- och tidskrav med enkelhet kan modifieras. Även om modellen representerar en kraftigt förenklad verklighet så kan den till exempel användas för att identifiera det antal incidenter som en lokal enhet maximalt kan hantera innan det uppstår resursbrist eller för att kommunicera dynamiken i incidenthantering på en hög abstraktionsnivå.



Figur 7.1: DHS-modell för simulering av incidenthantering.

Det är värt att notera att modellen i figur 7.1 är icke-deterministisk i bemärkelsen att flera transitionstillstånd kan vara aktiva samtidigt. Samtidighet kan i vissa fall leda till att oönskade dödlägen uppstår där till exempel en transition beror av en annan transition som i sin tur beror av den första transitionen, vilket resulterar i att entitetsflödet i simuleringen ”fastnar”. Om liknande situationer uppstår i verkligheten kan konsekvenserna vara fatala. Lyckligtvis så finns det verktyg som är speciellt utvecklade för tillståndsanalys (”state space analysis”) där dödlägen automatiskt kan identifieras. Ett exempel på ett sådant verktyg är CPN Tools [86] som för övrigt användes för att skapa modellen i figur 7.1.

## 7.2.2 Systemdynamik

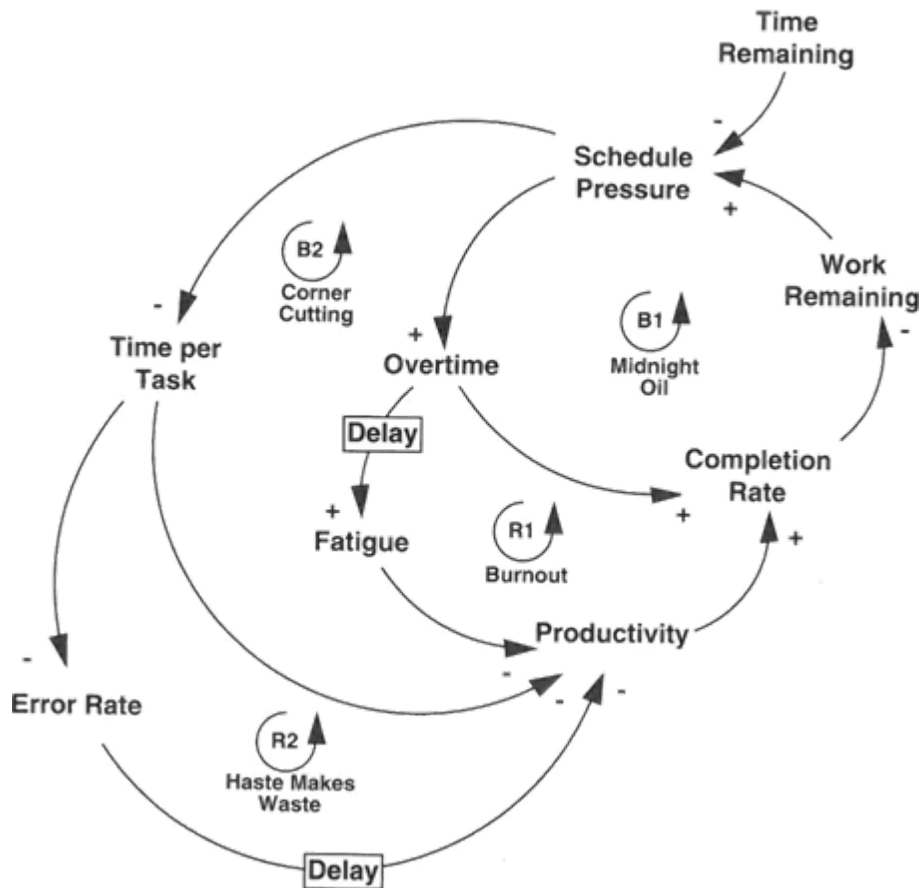
Systemdynamik (SD) är en simuleringsmetod som utvecklades i slutet av 50-talet av Jay Forrester [79]. SD används normalt för att analysera hur förändringar påverkar system eller verksamheter i sin helhet. SD är därför ett väl anpassat verktyg för simuleringar av verksamheter på strategisk nivå. SD kan även, likt DHS, användas som ett kommunikationsverktyg, för att beskriva hur verksamhetskomponenter direkt eller indirekt påverkar varandra.

En SD modell kräver att man har god kunskap om systemet som skall modelleras. Det är därför viktigt att utveckling av SD modeller sker i nära samarbete med de personer som vet, eller tror sig veta, hur verksamheten fungerar i sin helhet. I SD använder man sig av kausaldiagram, eller ”causal loop diagrams” (CLDs), för att överföra experters mentala modeller av systemets struktur till en överskådlig diagramform. I figur 7.2 visas ett enkelt kausaldiagram, vars ursprung är från [87], som beskriver hur produktivitet i en ingenjörsverksamhet påverkas av faktorer som övertid, press, tillgänglig tid, och så vidare.

Länkar i form av riktade pilar representerar påverkansrelationer mellan parvisa tillståndsvariabler. Pilar med minustecken indikerar att variablerna påverkar varandra i motsatt riktning medan plustecken indikerar att variablerna påverkar varandra i samma riktning. till exempel så kan man i diagrammet utläsa att en ökad trötthet minskar produktivitet (minustecken) medan minskad övertid resulterar i minskad trötthetsgrad (plustecken). Notera att plus- och minustecken inte nödvändigtvis betyder att variabler ökar respektive minskar, de beskriver enbart relationen mellan två variabler som både kan öka och minska.

I kausaldiagrammet återfinns fyra återkopplingslingor, eller ”feedback loops”, där varje slinga antingen är positiv eller negativ. En slinga uppstår när en variabel påverkar en annan variabel som i sin tur, indirekt eller direkt, påverkar den ursprungliga variabeln. En positiv slinga karakteriseras av att den resulterar i växande tillståndvariabler medan en negativ slinga har en balanserande (målsökande) effekt. I kausaldiagram är det enkelt att särskilja mellan positiva och negativa slingor genom att räkna antal minustecken i slingan: är antalet udda är det en balanserande slinga, annars är slingan

växande.

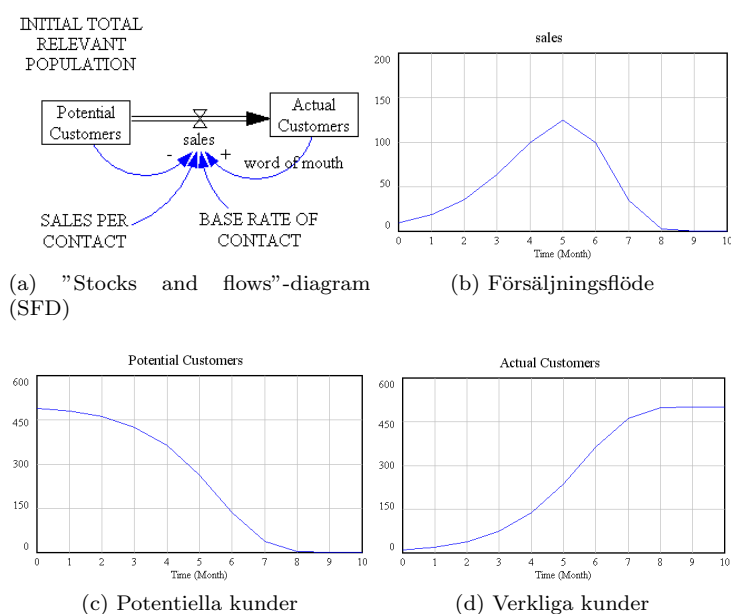


Figur 7.2: Exempel på ett kausaldiagram som representerar hur produktivitet påverkas av faktorer som utbrändhet, tidspress, arbetsbelastning etc. [87].

Ett kausaldiagram är väl anpassat för att representera experters mentala modeller på en relativt hög abstraktionsnivå. För att skapa exekverbara SD modeller översätts de initiala kausaldiagrammen till "stocks and flows"-diagram (SFD) där tillståndsvariabler, "stocks", och flöden, "flows", är associerade med integral- respektive differentialekvationer. Ekvationerna bildar tillsammans med diagrammets struktur och återkopplingslingor systemets holistiska beteende.

Figur 7.3a är ett klassiskt exempel, taget från [78], på ett SFD diagram där ett försäljningsflöde påverkas av tillståndsvariablerna potentiella kunder och verkliga kunder samt två hjälpvariabler som beskriver sannolikheten att potentiella och verkliga kunder träffas ("base rate of contact") samt att en

potentiell kund blir övertalad ("word of mouth") att övergå till en verklig kund ("sales per contact"). Modellen resulterar i ett S-format beteende hos tillståndsvariablerna (se figur 7.3c och figur 7.3d). Till exempel så kan man i figur 7.3c notera att antalet potentiella kunder initialt är många för att sedan sjunka och slutligen stabiliseras runt noll. På liknande sätt ser man i figur 7.3d att verkliga kunder initialt är få, men att antalet ökar med tiden för att slutligen stabiliseras runt det totala antalet kunder. Dynamiken i försäljningsflödet som ger upphov till detta beteende visas i figur 7.3b.



Figur 7.3: SD modell för simulering av försäljningsdynamik [78].

### 7.2.3 Agent-baserad modellering och simulering

Agent-baserad (AB) modellering är liksom SD en metod som kan användas för att simulera verksamheter på strategisk nivå. Den främsta skillnaden mellan AB- och SD-metoden är att det holistiska beteendet inte nödvändigtvis måste vara väldefinierat och känt före implementation av en AB-modell. (Däremot måste beteendet på komponentnivå vara känt, ty det är detta som bygger upp helheten.) AB är den enda kända metoden som har denna egenskap. Den grundläggande idén bakom AB är att det holistiska beteendet uppstår via fenomen, som i litteraturen benämns emergens eller själv-organisation, genom att låta enskilda agenter samverka och kommunicera med varandra i ett system av agenter. Det finns alltså ingen centralt styrande enhet i AB utan metoden är helt decentraliserad.

Agenter i AB är helt eller delvis autonoma och variationen på dess implementation är stor. Agenter är ofta utrustade med en förmåga att resonera som används för att fatta rationella beslut baserat på agentens kunskap. Dessutom kan agenter tilldelas förmågan att kontinuerligt utveckla och anpassa sig via inlärningsalgoritmer såsom ”reinforcement learning” eller med hjälp av andra algoritmer, till exempel neurala nätverk, genetisk programmering eller beslutsträd, som är vanliga inom maskininlärnings- och mönsterigenkänningsläran [88].

Inom ES representerar agentsystemet verksamheten i sin helhet och de enskilda agenterna representerar komponenter inom verksamheten. AB kan användas, och har tidigare använts, inom ES i hybrid-simuleringar tillsammans med DHS och SD [2, 83, 84]. I DHS så är det vanligt att entiteter i tillstånden (se kapitel 7.2.1) istället representeras av intelligenta agenter som autonomt kan avgöra om de skall gå från ett tillstånd till ett annat. Resultatet av att kombinera AB med DHS och SD är att modelleraren, där det är lämpligt, kan införa centraliserad styrning men ändå bibehålla AB-metodens naturliga sätt att modellera holistiska beteenden.

Eftersom AB är en beräkningsintensiv metod så har inte heller den applicerats i praktiken i någon större skala förrän på senare år. AB har därför inte nått samma höga mognadsgrad som DHS- och SD-metoderna och generella ES-verktyg för modellering, exekvering och analys av verksamheter är fåtaliga<sup>3</sup>. En annan nackdel med AB är att modeller är svåra att validera mot verkligheten eftersom verkligheten ofta är okänd. Därför är det viktigt att poängtera att AB enbart bör ses som en metod för att bättra ge insikt i komplexa system. Att använda AB för att generera prognoser eller för punktprediktion resulterar sannolikt i felaktigheter som kan vara direkt skadliga [90].

### 7.3 Sammanfattning

Syftet med verksamhetssimulering, eller ES, är att analysera verksamhetens dynamiska egenskaper och beteenden. ES är därför ett komplement till dagens EA-ramverk där verksamhetens statiska egenskaper och uppbyggnad modelleras. Vi har i detta kapitel identifierat och presenterat tre vanliga simuleringsmetoder inom ES: 1) Diskret händelsestyrd simulering (DHS); 2) Systemdynamik (SD); och 3) Agent-baserad (AB) modellering och simulering.

Det är viktigt att val av metod tar hänsyn till syftet med simuleringen. Till exempel så är det lämpligt att använda sig av DHS i studier där syftet är att identifiera flaskhalsar i flöden inom en verksamhet. SD är bra på att simulera helheten i verksamheter när det holistiska beteendet är känt, medan AB kan användas för att simulera verksamheter där det holistiska beteendet initialt är okänt.

---

<sup>3</sup>För den intresserade presenteras i [89] en sammanställning av de vanligaste AB-verktygen som finns idag.



## 8 Effektiva planeringsmetoder för MBFU

### 8.1 Introduktion

På senare år har intresset vuxit för att se militära operationer ur en effektbaserad synvinkel, *Effects Based Approach to Operations* (EBAO). Detta har bland annat berott på att befintliga synsätt och metoder varit dåligt anpassade för dagens mer komplexa militära operationsmiljö, inte minst som den ter sig i internationella fredsbevarande och -framtvängande insatser [91].

En effektbaserad syn ger beslutsfattaren möjligheten att systematiskt kunna testa och jämföra olika operativa planer och scenarier. På så sätt kan man dra slutsatser kring vilka insatser (planer) som ger bästa möjliga effekt [92]. Tanken är att man då ska kunna optimera sina resurser så att man med så liten insats som möjligt ska uppnå en så stor önskad effekt som möjligt. Denna beslutsoptimering är viktig, för annars riskerar man att förbruka onödiga resurser (både materiella och mänskliga) vilket kan bli kostsamt för Försvarsmakten. Inom FOI finns det olika projekt där man utvecklar och värderar M&S-baserade beslutsstöd för de olika planeringsprocesser som förekommer i planeringsdelen. En annan nytta är att man genom M&S skulle kunna bygga effektiva förmågor baserade på redan existerande delkomponenter inom FM.

### 8.2 Förmågor, operationsplanering och arkitekturbeskrivningar

Arkitekturbeskrivningar kan stödja militär förmågeutveckling och operationsplanering på flera sätt. Arkitekturramverket MODAF (se avsnitt 2.4) är konstruerat för att möjliggöra beskrivningar av ett flertal viktiga aspekter (vy-beteckningar inom parentes):

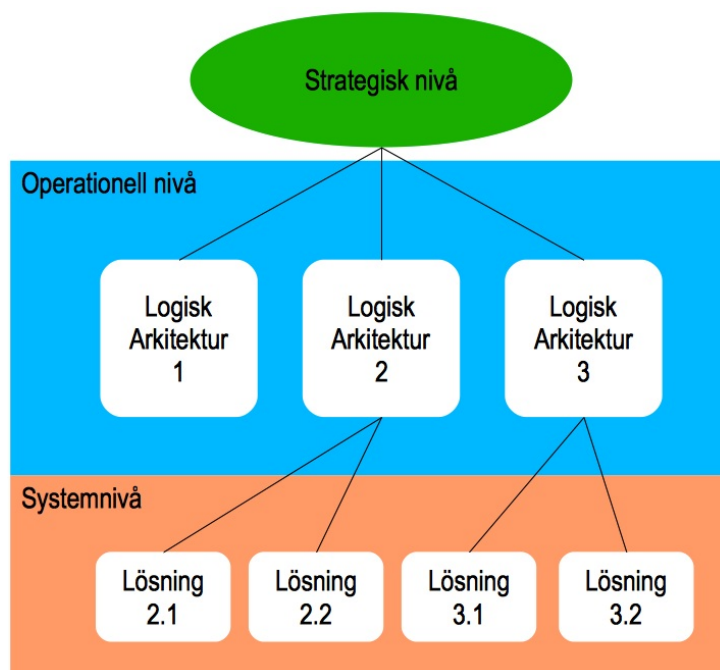
**De strategiska vyerna** beskriver bland annat militära förmågors inbördes beroenden (StV-4), förmågornas koppling till befintlig organisation (StV-5) och hur förmågorna realiserar av operationer (StV-6).

**De operationella vyerna** beskriver bland annat operationsplaner (*Concept of operations*, CONOPS) på en övergripande nivå (OV-1a och OV1-b), relationer mellan olika förmågor i operationen (OV-2), behoven av informationsutbyte (OV-3), organisatoriska förhållanden (OV-4) och hur de aktiviteter som utförs bidrar till att uppnå de mål som satts upp (OV-5).

**Systemvyerna** beskriver bland annat hur resurser sätts samman och interagerar (SV-1), hur de bidrar till att uppnå operativa mål (SV-5) och vilken prestanda de har (SV-7).



Den information som dessa beskrivningar rymmer kan användas för att ta fram olika sorters lösningar och utvärdera dem mot varandra, vilket konceptuellt illustreras i Figur 8.1.



Figur 8.1: Arkitekturer på logisk- och systemnivå.

Det är viktigt att inse att framtagning av olika sorters lösningar enligt ovan kan ske på två olika plan:

**Förmågeutveckling av framtida förmågor** är det användningsområde som MODAF vanligen anses vara till för. Här återfinns aktiviteter som projektstyrning, upphandling, materielutveckling etc.

**Planering för insats med befintliga förmågor** kan dock också dra nytta av den information som finns lagrad i arkitekturbeskrivningar.

Relationen mellan dessa två är inte helt okomplicerad. Användning av arkitekturramverk för förmågeutveckling kan ofta vara förvirrande lik operativ planering: på sätt och vis gör man hypotetisk operativ planering i utvärderingssyfte. Det som ska utvärderas kan vara hypotetiska förmågor, hypotetiska sammansättningar av befintliga förmågor, hypotetiska arkitekturlösningar på systemnivå etc. Även utvärderingen kan vara av olika slag: från kvalitativa bedömningar av sakkunniga experter till

kvantitativa resultat baserat på matematiska beräkningar eller simuleringar. En gemensam nämnare är dock att användningen av operativ planering i förmågeutvecklingssyfte alltid är en väsentlig förenkling av en fullständig operativ plan, där endast viktigare karakteristika bibehålls, renodlas och analyseras.

Ett exempel på analys av en framtida förmåga är [73], som med hjälp av en DoDAF-modell försöker uppskatta tillförlitligheten hos ett framtida (hypotetiskt) missilförsvarssystem. I [93] finns fler exempel på hur olika designval för förmågor – beskrivna som MODAF-modeller – kan utvärderas i scenarier.

### 8.3 Algoritmer

Att utifrån arkitekturbeskrivningar av förmågor, förband och system automatiskt sätta samman optimala lösningar innebär att göra en sökning i ett abstrakt *sökrum* av möjliga alternativ. Vissa alternativ är *otillåtna*, exempelvis eftersom de överskrider en personalram eller budget eller inte uppfyller ett förmågekrav. Bland de tillåtna alternativen är vissa *bättre* än andra, exempelvis eftersom de är billigare, inbegriper färre caveats eller levererar fler förmågor. Bland de tillåtna alternativen kan ett vara *optimalt*, det vill säga bästa tänkbara alternativ. Om det finns många önskvärda egenskaper som inte kan reduceras till varandra så får man en speciell typ av problem som behandlas i teorier om så kallad *multiple criteria decision making* [94].

Det finns många olika sökalgoritmer som är intressanta för att kunna identifiera optimala alternativ i ett sökrum. Om problemet kan formuleras som ett grafsökingsproblem så är Dijkstras<sup>1</sup> algoritm och A\*-algoritmen<sup>2</sup> två intressanta kandidater. Dijkstras algoritm är en s.k. "girig" algoritm som alltid försöker välja den bästa vägen i en graf. A\*-algoritmen är en förbättrad version av Dijkstras algoritm som uppnår bättre prestanda genom att använda sig av heuristiker.

Dijkstras algoritm kan ses som ett specialfall av *dynamisk programmering*, en algoritm för att bryta ner stora komplexa problem i små och mer lätthanterliga delproblem. En lösning av ett förmågesammansättningsproblem med dynamisk programmering kan enkelt uttryckt beskrivas som att man börjar med att betrakta den förmåga man vill uppnå och sedan arbetar sig baklänges genom de beslut som måste tas för att uppnå den.

Många problem lämpar sig för optimeringsmetoder som *linjärprogrammering* och *heltalsprogrammering* – problemformuleringar som uttrycker en målfunktion och dess bivillkor i matematisk form som system av ekvationer och olikheter. Det finns ett stort antal algoritmer för att effektivt lösa sådana problem. Metoderna har fått stort genomslag i planering av militär flygverksamhet, där det finns gott om heltals- och

<sup>1</sup>[http://sv.wikipedia.org/wiki/Dijkstras\\_algoritm](http://sv.wikipedia.org/wiki/Dijkstras_algoritm)

<sup>2</sup>[http://en.wikipedia.org/wiki/A\\*\\_search\\_algorithm](http://en.wikipedia.org/wiki/A*_search_algorithm)

linjärprogrammeringsmodeller för optimal sammansättning av flygförband och utförande av flygföretag [95, 96]. Heltals- och linjärprogrammering skulle på samma sätt kunna användas för att med hjälp av information från arkitekturbeskrivningar optimalt sätta samman förmågor. Ett exempel på användning av heltalsprogrammering kopplat till arkitekturbeskrivningar för beslutsstöd är [97].

## 8.4 Sammanfattning

Att beskriva militära förmågor är intressant i många olika sammanhang, både på lång sikt vid utvecklingen av nya förmågor och på kort sikt vid användningen av befintliga. Ofta är det fruktbart att betrakta förmågorna som uppbyggda av komponenter: resurser såsom förband och system, konfigurerade på ett visst sätt.

En naturlig fråga att ställa sig blir då: hur ska man sätta ihop sina komponenter för att så billigt och effektivt som möjligt uppnå de förmågor som eftersträvas? Eftersom resurser och system alltid är begränsade i antal och beroende av varandra blir problemet svårt att överblicka och lösa. Arkitekturmodeller och liknande formella beskrivningar av komponenterna kan underlätta både visualiseringen och sökandet efter optimala lösningar. Givet att arkitekturmodellerna är konstruerade på ett smart sätt så finns det ett flertal algoritmer som skulle kunna bidra till att automatiskt ta fram lösningsalternativ.

## 9 Summering

### 9.1 Modellbaserad förmågeutveckling

Försvarsmakten siktar på att realisera modellbaserad förmågeutveckling, MBFU. En hörnsten i detta sammanhang är Enterprise Architecture, EA, med Ministry of Defence Architecture Framework, MODAF, som grund. Fördelarna med EA utifrån en modellbaserad grund är flera, relativt det dokumentbaserade angreppssätt för förbands- och materielutveckling som oftast används idag. För att bedriva utveckling med EA som grund krävs ett ramverk. EA-ramverk är dock ett heterogent begrepp där innehåll och användning skiljer sig åt. I själva verket är det kanske i kombinationen av ramverk som den största nyttan går att finna. Norska försvaret har försökt kombinera NAF och TOGAF, där NAF fungerar som struktur för modeller som tas fram och underhålls enligt TOGAF ADM. Försvarsmakten bör ta vara på de norska erfarenheterna, inte minst avseende behovet av förenkling av såväl metamodeller som processer.

Modellbaserad utveckling och EA har flera uppenbara fördelar. Samtidigt ska påpekas att det är komplext och framförallt tidskrävande att införa som arbetssätt i en organisation. Förutom att de tekniska aspekterna måste finnas på plats, i form av verktyg, referensmodeller, bibliotek gemensam terminologi etc., vilket ändå får anses vara förhållandevis lätt att åstadkomma, måste personal utbildas och tränas i att tillämpa det modellbaserade angreppssättet. Det senare kräver resurser och det kan ibland vara svårt att se den omedelbara nyttan av EA.

Det finns idag en mogen marknad för EA-verktyg, där ett flertal aktörer konkurrerar med varandra. Många verktygsförsäljare har under 2011 börjar kompositionera sig och talar med om affärsplanering och mindre om *enterprise architecture*, en term som är belastad i vissas ögon. Konkurrensen har också skärpts av att verktygsköpare allt mer noggrant behöver demonstrera vilket affärsvärde som deras EA-satsningar faktiskt tillför. Slutligen har branschen under 2011 påverkats en del av uppköp och förvärv, såsom SAP:s köp av Sybase och OpenTexts köp av Metastorm.

### 9.2 Metoder för kvalificerat beslutsstöd

Försvarsmaktens modellbaserade förmågeutveckling inbegriper potentiellt alla dagens och morgondagen försvarsverksamheter. Denna mycket omfattande ambitionsnivå ställer krav på systematisk inhämtning och återanvändning av befintlig kunskap och kommersiella simuleringsramverk. Egenutveckling är inte ett försvarbart alternativ. Ett väldesignat och välfungerande simuleringsramverk tillsammans med ett ackrediterat modellbibliotek skulle kunna erbjuda en stabil grundplattform, där man snabbt och effektivt

kan prova idéer, värdera handlingsalternativ, generera beslutsunderlag etc. Det simuleringsramverk som väljs bör även ha egenskaper som tillåter bl.a. anpassning av verktyget till användarens behov.

I nära anslutning till frågan om simuleringsramverk finns ett antal andra metoder som kan användas för att nyttja arkitekturbeskrivningar för beslutsstöd till fullo:

**EA-analys** karakteriseras av att icke-triviala egenskaper hos en arkitektur beräknas och presenteras som stöd för en beslutsfattare. En framgångsfaktor är väldefinierade beslutssituationer med väldefinierade mål som analysen ska stödja. Analysformalismer som probabilistiska relationsmodeller och probabilistisk OCL har med framgång använts i fallstudier på företag och organisationer. För PRM- och p-OCL-formalismerna finns det idag gratis verktygsstöd från en forskargrupp på KTH.

**Verksamhetssimulering** kan till den traditionella användningen av arkitekturbeskrivningar lägga analys av verksamhetens dynamiska egenskaper och beteenden. Tre tänkbara simuleringsmetoder är diskret händelsestyrd simulering (DHS), systemdynamik (SD) och agent-baserad (AB) modellering och simulering.

**Domänspecifika språk** skall fungera som intuitiva gränssnitt till ramverk där varje modell kompileras till en körbar konfiguration av ramverket. Förmågespecifika MODAF profiler skulle kunna tänkas tjänstgöra som gränssnitt till simuleringsbibliotek på operativ eller taktisk nivå. Med förmågespecifika MODAF profiler skulle analysen (simuleringen) ske med blott en knapptryckning – avgränsningen till en specifik försvarsförmåga skulle göra det möjligt att generera körbar simuleringskod ur arkitektur på en hög och naturlig abstraktionsnivå.

**I effektbaserad planering** kan arkitekturbeskrivningar av delförmågor användas för att finna optimala konfigurationer av sammansatta förmågor.

## Litteraturförteckning

- [1] Vahid Mojtahed Martin Eklöf. FOI Memo 3600: M&S-MBFU - Behovsanalys, 2011.
- [2] Christopher G. Glazner. *Understanding Enterprise Behavior using Hybrid Simulation of Enterprise Architecture*. PhD thesis, Massachusetts Institute of Technology, Engineering Systems Division, Cambridge, Massachusetts, 2009.
- [3] P. Johnson, R. Lagerström, P. Närman, and M. Simonsson. Enterprise architecture analysis with extended influence diagrams. *Information Systems Frontiers*, 9(2):163–180, 2007.
- [4] Wikipedia. Enterprise Architecture — Wikipedia, The Free Encyclopedia, November 2011.  
[http://en.wikipedia.org/w/index.php?title=Enterprise\\_architecture&oldid=456424459](http://en.wikipedia.org/w/index.php?title=Enterprise_architecture&oldid=456424459).
- [5] Ministry of Defence. MODAF Glossary v1.2, November 2011.  
[http://www.mod.uk/NR/rdonlyres/D4CEF7F5-B008-4734-9D05-7DD746B0F166/0/20090304\\_MODAF01\\_2Glossary\\_V1\\_0\\_\\_1.pdf](http://www.mod.uk/NR/rdonlyres/D4CEF7F5-B008-4734-9D05-7DD746B0F166/0/20090304_MODAF01_2Glossary_V1_0__1.pdf).
- [6] CIO Council. Federal Enterprise Architecture Framework (FEAF). Technical report, Office of Management and Budget, USA, 1999.
- [7] R.E. Giachetti. *Design of Enterprise Systems: Theory, Architecture and Methods*. CRC Press, 2010.
- [8] R. Hilliard. IEEE-Std-1471-2000 Recommended Practice for Architectural Description of Software-Intensive Systems. *IEEE*, 2000.  
<http://standards.ieee.org>.
- [9] The Open Group. TOGAF Online, November 2011.  
<http://pubs.opengroup.org/architecture/togaf9-doc/arch/index.html>.
- [10] S. Westin. Planerat, alltför planerat: En perspektivistisk studie i stadsplaneringens paradoxer. 2010. Doktorsavhandling, Kulturgeografiska institutionen, Uppsala universitet.
- [11] The Open Group. *TOGAF 9*. Van Haren Publishing, Zaltbommel, Netherlands, 2009.
- [12] Vasilis Boucharas, Marlies Steenbergen, Slinger Jansen, and Sjaak Brinkkemper. The Contribution of Enterprise Architecture to the Achievement of Organizational Goals: A Review of the Evidence. In

- Erik Proper, Marc M. Lankhorst, Marten Schönherr, Joseph Barjis, Sietse Overbeek, Wil Aalst, John Mylopoulos, Michael Rosemann, Michael J. Shaw, and Clemens Szyperski, editors, *Trends in Enterprise Architecture Research*, volume 70 of *Lecture Notes in Business Information Processing*, pages 1–15. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-16819-2. [http://dx.doi.org/10.1007/978-3-642-16819-2\\_1](http://dx.doi.org/10.1007/978-3-642-16819-2_1).
- [13] John A. Zachman. A framework for information systems architecture. *IBM Syst. J.*, 26(3):276–292, 1987. ISSN 0018-8670.
- [14] NATO NC3 Board. Nato Architecture Framework NAF v.3. Technical report, NATO, 2009. <http://www.nhq3s.nato.int/ARCHITECTURE/>.
- [15] Department of Defense Architecture Framework Working Group. DoD Architecture Framework, version 1.5. Technical report, Department of Defense, USA, 2007.
- [16] Ministry of Defence. MOD Architecture Framework version 1.2.003. Technical report, Ministry of Defence, UK, September 2008.
- [17] Wikipedia. Enterprise architecture framework — Wikipedia, The Free Encyclopedia, November 2011. [http://en.wikipedia.org/w/index.php?title=Enterprise\\_architecture\\_framework&oldid=459995069](http://en.wikipedia.org/w/index.php?title=Enterprise_architecture_framework&oldid=459995069).
- [18] UPDM Group. Unified Profile for DoDAF and MODAF (UPDM), November 2011. <http://www.updm.com/index.htm>.
- [19] Ministry of Defence. MODAF Views & Viewpoints, November 2011. <http://www.mod.uk/DefenceInternet/AboutDefence/WhatWeDo/InformationManagement/MODAF/ViewpointsAndViews.htm>.
- [20] Ministry of Defence. What is the MODAF Meta-Model?, November 2011. [http://www.mod.uk/NR/rdonlyres/0360559B-3D6B-4198-BD92-D2B783A7E831/0/20090310\\_MODAF\\_Meta\\_Model\\_V1\\_0\\_U.pdf](http://www.mod.uk/NR/rdonlyres/0360559B-3D6B-4198-BD92-D2B783A7E831/0/20090310_MODAF_Meta_Model_V1_0_U.pdf).
- [21] John P. Zachman. The Zachman Framework Evolution, November 2011. <http://zachman.com/ea-articles-reference/54-the-zachman-framework-evolution>.
- [22] R. Sessions. A comparison of the top four enterprise-architecture methodologies. 466232(May):1–28, 2007. <http://msdn.microsoft.com/en-us/library/bb466232.aspx>.

- [23] H.D. Jørgensen, T. Liland, and S. Skogvold. Aligning TOGAF and NAF-Experiences from the Norwegian Armed Forces. *The Practice of Enterprise Modeling*, pages 131–146, 2011.
- [24] Chris Wilson and Julie Short. Magic Quadrant for Enterprise Architecture Tools. Technical report, 2010.
- [25] Florian Matthes, Sabine Buckl, Jana Leitel, and Schweda Christian M. Enterprise architecture management tool survey 2008. Technical report, Software Engineering for Business Information Systems (sebis), Technische Universität München, 2008.
- [26] Alfabet. planningIT Support for Architecture Frameworks. Technical report, Alfabet, 2007.
- [27] SJ Ring and M. Johnson. Architecture development and analysis survey: The state of dod architecting. In *Winter 2005 Command Information Superiority Architectures (CISA) Worldwide Conference*, 2005.
- [28] Chris Wilson and Julie Short. Magic Quadrant for Enterprise Architecture Tools. Technical report, 2011.
- [29] SOFA - Simulation Open Framework Architecture, November 2011.  
<http://www.sofa-framework.org/>.
- [30] OMNeT++, Network Simulation Framework, November 2011.  
<http://www.omnetpp.org/>.
- [31] Vahid Mojtahed, Göran Bergström och Mattias Karlsson. En jämförelse av ansatserna SaSS-UV och UV-strid. Technical Report FOI-R–1568-SE, Totalförsvarets Forskningsinstitut, Januari 2005.
- [32] Göran Bergström, Johan Pelo, Mattias Karlsson och Vahid Mojtahed. SaSS-UV, System-av-System Simulering med simuleringsramverk - inriktning UV. Technical Report FOI-R–1769-SE, Totalförsvarets Forskningsinstitut, December 2005.
- [33] Johan Pelo. M&S Infrastruktur för Rapid prototyping. Technical Report FOI-R–2626-SE, Totalförsvarets Forskningsinstitut, December 2008.
- [34] Gunnar Holm. *Systemmodellering och Simulering med särskild inriktning mot Försvarsmakten*. Totalförsvarets Forskningsinstitut, 2007. ISBN 978-91-7056-124-5.
- [35] FLAMES - FLExible Analysis and Mission Effectiveness System, November 2011.  
<http://www.ternion.com/>.
- [36] Canada’s role in so called “missile defense”. *Press for Conversion - Canada*, pages 32–37, (Issue no. 56), June 2005.



- [37] Strive - simulation development and application environment, November 2011.  
[http://www.defence.gov.au/teamaustralia/simulation\\_development\\_and\\_application\\_environment\\_\(STRIVE\).htm](http://www.defence.gov.au/teamaustralia/simulation_development_and_application_environment_(STRIVE).htm).
- [38] Simon Hallé and Nicolas Bernier. Analysis and review of synthetic simulation environment - CAE STRIVE Suite 2.0. Technical Report RDDC Valcartier CR 2005-555, DRDC - Defence Research and Development Canada, December 2005.
- [39] Aeria, November 2011.  
[http://www.presagis.com/products\\_services/aeria/](http://www.presagis.com/products_services/aeria/) and  
<http://enter-aeria.com/>.
- [40] David Wilson, Nigel Allen and Brian Topping. Use of stage modelling to quantify survivability effectiveness. *EUROPEAN SURVIVABILITY WORKSHOP*, 2008.
- [41] Stage, November 2011.  
[http://www.presagis.com/products\\_services/products/modeling-simulation/simulation/stage/](http://www.presagis.com/products_services/products/modeling-simulation/simulation/stage/) and  
[http://www.presagis.com/files/product\\_brochures/2011\\_11\\_DS\\_SIM\\_STAGE\\_.pdf](http://www.presagis.com/files/product_brochures/2011_11_DS_SIM_STAGE_.pdf).
- [42] AI.implant, November 2011.  
[http://www.presagis.com/products\\_services/products/modeling-simulation/simulation/aiimplant/](http://www.presagis.com/products_services/products/modeling-simulation/simulation/aiimplant/).
- [43] MASA Group, November 2011.  
<http://www.masagroup.net/>.
- [44] MASA Sword, November 2011.  
<http://www.masagroup.net/products/masa-sword.html>.
- [45] MÄK, November 2011.  
<http://www.mak.com/>.
- [46] VR-Forces, November 2011.  
<http://www.mak.com/products/simulate/computer-generated-forces.html>.
- [47] Pelo Johan, Ohlsson Kjell och Pers Lisbeth. Simuleringsramverk och logistiksimulering - en översiktsstudie. Technical Report FOI-R-1318-SE, Totalförsvarets Forskningsinstitut, 2004.
- [48] Steven Kelly and Juha-Pekka Tolvanen. *Domain-Specific Modeling: Enabling Full Code Generation*. Wiley, 2008. ISBN 978-0-470-03666-2.

- [49] Martin Fowler. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Addison-Wesley, Boston, MA, 3. edition, 2003. ISBN 978-0-321-19368-1.
- [50] Martin Fowler. *Domain-Specific Languages*. Addison-Wesley Professional, 2010. ISBN 0321712943.
- [51] Jack Greenfield, Keith Short, Steve Cook, and Stuart Kent. *Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools*. Wiley, 2004.
- [52] Steven P. Miller, Michael W. Whalen, and Darren D. Cofer. Software model checking takes off. *Commun. ACM*, 53:58–64, February 2010. ISSN 0001-0782.  
<http://doi.acm.org/10.1145/1646353.1646372>.
- [53] T. R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition Academic Press Inc. 5(2)*, 1993.
- [54] Jr. Edmund M. Clarke, Orna Grumberg, and Doron A. Peled. *Model Checking*. MIT Press, 1999.
- [55] Limor Fix. Fifteen years of formal property verification in intel. In Orna Grumberg and Helmut Veith, editors, *25 Years of Model Checking*, volume 5000 of *Lecture Notes in Computer Science*, pages 139–144. Springer, 2008. ISBN 978-3-540-69849-4.  
<http://dblp.uni-trier.de/db/conf/spin/lncs5000.html#Fix08>.
- [56] Ethan K. Jackson, Eunsuk Kang, Markus Dahlweid, Dirk Seifert, and Thomas Santen. Components, platforms and possibilities: towards generic automation for mda. In Luca P. Carloni and Stavros Tripakis, editors, *EMSOFT*, pages 39–48. ACM, 2010. ISBN 978-1-60558-904-6.  
<http://dblp.uni-trier.de/db/conf/emsoft/emsoft2010.html#JacksonDSS10>.
- [57] Moshe Y. Vardi. From church and prior to psl. In Orna Grumberg and Helmut Veith, editors, *25 Years of Model Checking*, volume 5000 of *Lecture Notes in Computer Science*, pages 150–171. Springer, 2008. ISBN 978-3-540-69849-4.  
<http://dblp.uni-trier.de/db/conf/spin/lncs5000.html#Vardi08>.
- [58] Bran Selic. A systematic approach to domain-specific language design using uml. In *ISORC*, pages 2–9, 2007.
- [59] Marc M. Lankhorst, René van Buuren, Diederik van Leeuwen, Henk Jonkers, and Hugo ter Doest. Enterprise architecture modelling—the issue of integration. *Advanced Engineering Informatics*, 18(4):205 – 216, 2004. ISSN 1474-0346. doi: 10.1016/j.aei.2005.01.005. Special issue on

Enterprise Modelling and System Support.  
<http://www.sciencedirect.com/science/article/pii/S1474034605000054>.

- [60] S. Kurpjuweit and R. Winter. Viewpoint-based meta model engineering. In *Proceedings of EMISA*, pages 143–161. Citeseer, 2007.
- [61] Pontus Johnson. Ritningens roll i arkitekturens historia. Föredrag på KTH:s Enterprise Architecture Symposium, October 2010.
- [62] S. Aier, S. Buckl, U. Franke, B. Gleichauf, P. Johnson, P. Närman, C.M. Schweda, and J. Ullberg. A survival analysis of application life spans based on enterprise architecture models. In *3rd International Workshop on Enterprise Modelling and Information Systems Architectures, Ulm, Germany*, pages 141–154, 2009.
- [63] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *Proc. of the 16th International Joint Conference on Artificial Intelligence*, pages 1300–1309. Morgan Kaufman, 1999.
- [64] R. Lagerström, U. Franke, P. Johnson, and J. Ullberg. A method for creating enterprise architecture metamodels—applied to systems modifiability analysis. *International Journal of Computer Science and Applications*, 6(5):89–120, 2009.
- [65] R. Lagerström, P. Johnson, and D. Höök. Architecture analysis of enterprise systems modifiability-Models, analysis, and validation. *Journal of Systems and Software*, 83(8):1387–1403, 2010. ISSN 0164-1212.
- [66] Teodor Sommestad, Mathias Ekstedt, and Pontus Johnson. A probabilistic relational model for security risk analysis. *Computers & Security*, 29(6): 659–679, September 2010.
- [67] OMG. Object constraint language, version 2.2. Technical report, Object Management Group, OMG, feb 2010. OMG Document Number: formal/2010-02-01.  
<http://www.omg.org/spec/OCL/2.2>.
- [68] P. Närman, H. Holm, P. Johnson, J. König, M. Chenine, and M. Ekstedt. Data accuracy assessment using enterprise architecture. *Enterprise Information Systems*, 5(1):37–58, 2011. ISSN 1751-7575.
- [69] Per Närman, Ulrik Franke, Johan König, Markus Buschle, and Mathias Ekstedt. Enterprise architecture availability analysis using fault trees and stakeholder interviews. *Enterprise Information Systems*, 2011. Accepted with revisions, to appear.
- [70] Ulrik Franke and Pontus Johnson. Enterprise IT service availability modeling using probabilistic OCL. 2011. Submitted manuscript.

- [71] Johan Ullberg, Ulrik Franke, Markus Buschle, and Pontus Johnson. A tool for interoperability analysis of enterprise architecture models using pi-OCL. In *Proceedings of The international conference on Interoperability for Enterprise Software and Applications (I-ESA)*, April 2010.
- [72] Ulrik Franke, Pontus Johnson, Evelina Ericsson, Waldo Rocha Flores, and Kun Zhu. Enterprise architecture analysis using fault trees and MODAF. In *Proc. CAiSE Forum 2009*, volume 453, pages 61–66, June 2009. ISSN 1613-0073.
- [73] B. Liu and X. Wu. Mission reliability analysis of missile defense system based on DODAF and Bayesian networks. In *Quality, Reliability, Risk, Maintenance, and Safety Engineering (ICQR2MSE), 2011 International Conference on*, pages 777–780. IEEE.
- [74] Ulrik Franke, Waldo Rocha Flores, and Pontus Johnson. Enterprise architecture dependency analysis using fault trees and bayesian networks. In *Proc. 42nd Annual Simulation Symposium (ANSS)*, pages 209–216, March 2009. ISBN 1-56555-331-4.  
<http://www.scs.org>.
- [75] M. Buschle, J. Ullberg, U. Franke, R. Lagerström, and T. Sommestad. A tool for enterprise architecture analysis using the prm formalism. *Information Systems Evolution*, pages 108–121, 2011.
- [76] Markus Buschle, Hannes Holm, Teodor Sommestad, Mathias Ekstedt, and Khurram Shahzad. A tool for automatic enterprise architecture modeling. In *Proceedings of the CAiSE Forum 2011*, volume 734, pages 25–32, June 2011.
- [77] Milind M. Datar. Enterprise simulation: framework for a strategic application. In *Proceedings of the 32nd conference on Winter simulation, WSC '00*, pages 2010–2014, San Diego, CA, USA, 2000. Society for Computer Simulation International. ISBN 0-7803-6582-8.  
<http://portal.acm.org/citation.cfm?id=510378.510672>.
- [78] Craig W. Kirkwood. *System Dynamics Methods: A Quick Introduction*. College of business, Arizona State University, 1998.
- [79] Jay Forrester. Industrial dynamics: A major breakthrough for decision makers. *Harvards business review*, 36(4):37–66, 1958.
- [80] Ostic J.K. and Cannon C.E. An introduction to enterprise modeling and simulation. Technical report, Los Alamos National Lab, Technology Modeling and Analysis Group, September 1996.
- [81] Sabine Buckl, Florian Matthes, Wolfgang Renz, Christian M. Schweda, and Jan Sudeikat. Towards simulation-supported enterprise architecture.

- Fachtagung Modellierung betrieblicher Informationssysteme (MobIS2008)*, pages 131–145, 2008.
- [82] Ministry of Defence. MOD Architecture Framework version 1.2.003. Technical report, Ministry of Defence, 2008.
- [83] Rabelo Luis, Magdy Helal, Albert Jones, and Hyeung-Sik Min. Enterprise simulation: a hybrid system approach. *International Journal of Computer Integrated Manufacturing*, 18(6):498–508, 2005.  
<http://dx.doi.org/10.1111/j.1468-2486.2009.00911.x>.
- [84] Andrei Borshchev and Alexei Filippov. From system dynamics and discrete event to practical agent based modeling: Reasons, techniques, tools. *The 22nd International Conference of the System Dynamics Society*, July 2004.
- [85] Michael J. North, Nicholson T. Collier, and Jerry R. Vos. Experiences creating three implementations of the repast agent modeling toolkit. *ACM Trans. Model. Comput. Simul.*, 16:1–25, January 2006. ISSN 1049-3301.  
<http://doi.acm.org/10.1145/1122012.1122013>.
- [86] Kurt Jensen and Lars M. Kristensen. *Coloured Petri Nets - Modelling and Validation of Concurrent Systems*. Springer Publishing Company, 2009.
- [87] John D. Sterman. *Business dynamics: systems thinking and modeling for a complex world*. Irwin/McGraw-Hill, 2000.
- [88] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000. ISBN 0471056693.
- [89] Cynthia Nikolai and Gregory Madey. Tools of the Trade: A Survey of Various Agent Based Modeling Platforms. *Journal of Artificial Societies and Social Simulation*, 12(2):2, 2009.  
<http://jasss.soc.surrey.ac.uk/12/2/2.html>.
- [90] Eric Bonabeau. Agent-Based Modeling: Methods and Techniques for Simulating Human Systems. *Proceedings of the National Academy of Sciences of the United States of America*, 99(10):7280–7287, 2002. ISSN 00278424. doi: 10.2307/3057854.  
<http://dx.doi.org/10.2307/3057854>.
- [91] J. Schubert, H. Asadi, F. Harrysson, P. Hörling, F. Kamrani, B. Kylesten, A. Linderhed, F. Moradi, E. Sjöberg. Real-time Simulation Supporting Effects-Based Planning 2008-2010 - Final Report. Technical Report FOI-R-3060-SE, Swedish Defence Research Agency, 2010. ISSN 1650-1942.

- [92] J. Schubert, G. Holm, P. Hörling, F. Kamrani, B. Kylesten, F. Moradi, E. Sjöberg, P. Svensson. Realtidssimulering som stöd för effektbaserad planering 2008. Technical Report FOI-R-2616-SE, Swedish Defence Research Agency, 2008. ISSN 1650-1942.
- [93] E.N. Urwin, C.C. Venters, D.J. Russell, L. Liu, Z. Luo, D.E. Webster, M. Henshaw, and J. Xu. Scenario-based design and evaluation for capability. In *System of Systems Engineering (SoSE), 2010 5th International Conference on*, pages 1–6. IEEE, 2010.
- [94] M. Zeleny. *Multiple criteria decision making*, volume 25. McGraw-Hill New York, 1982.
- [95] B.J. Griggs, G.S. Parnell, and L.J. Lehmkuhl. An air mission planning algorithm using decision analysis and mixed integer programming. *Operations Research*, pages 662–676, 1997.
- [96] K.A. Yost and A.R. Washburn. Optimizing assignment of air-to-ground assets and bda sensors. *Military Operations Research*, 5(2):77–91, 2000.
- [97] Ulrik Franke, Oliver Holschke, Markus Buschle, Per Närman, and Jannis Rake-Revelant. It consolidation: An optimization approach. In *Enterprise Distributed Object Computing Conference Workshops (EDOCW), 2010 14th IEEE International*, pages 21 –26, oct. 2010. doi: 10.1109/EDOCW.2010.11.