KUNGL TEKNISKA HÖGSKOLAN
Royal Institute of Technology
Numerical Analysis and Computing Science

TRITA-NA-D9907 • CID-49, KTH, Stockholm, Sweden 1999

# IT-baserade Matematikverktyg
# några tidigare och några pågående KTH-projekt

Ambjörn Naeve

CID
Centre for
User Oriented IT Design

**Ambjörn Naeve**

It-baserade Matematikverktyg - några tidigare och några pågående KTH-projekt

**Report number:** TRITA-NA-D9907, CID-49
**ISSN number:** ISSN 1403-073X
**Publication date:** June 1999
**E-mail of author:** amb@nada.kth.se
**URL of author:** http://www.nada.kth.se/~amb

**Reports can be ordered from:**
CID, Centre for User Oriented IT Design
Nada, Dept. Computing Science
KTH, Royal Institute of Technology
S-100 44 Stockhom, Sweden
telephone: + 46 8 790 91 00
fax: + 46 8 790 90 99
e-mail: cid@nada.kth.se
URL: http://www.nada.kth.se/cid/

## INNEHÅLLSFÖRTECKNING


1 Bakgrund 1

2 Tidiga utvecklingsprojekt av IT-stödda matematikverktyg 2
- 2.1 MapCon 2
- 2.2 MapAnalyze 2
- 2.3 MacTapet 3
- 2.4 Drawboard 3
  - 2.4.1 De grundläggande idéerna 3
  - 2.4.2 Den första prototypen 4
  - 2.4.3 MacDrawboard - en efterföljare 5
  - 2.4.4 Projective Drawing Board - en modern vidareutveckling 5
- 2.5 MacFlow - en grafisk programmeringsmiljö 6
- 2.6 PrimeTime 6

3 Geometriska IT-verktyg - utvecklingsarbetet på CVAP 6
- 3.1 Bakgrund 6
- 3.2 Reflections - ett system för simulering av vågfronter 8
- 3.3 Dynamisk Projektiv Geometri 8
- 3.4 Geometrisk Algebra 9

4 Förstklassig Matematik - ett projekt i min dotters klass. 9
- 4.1 Bakgrund 9
- 4.2 Matematiken som logiskt uttestade fantasifoster 9
- 4.3 Det matematiska smörgåsbordet 10
- 4.4 En variabel som en låda med ett namn och ett innehåll 11

5 Kunskapens Trädgård 13
- 5.1 Bakgrund - CID 13
- 5.2 Att vända ut och in på lärandet 14
- 5.3 Komponenter och läroupplevelser - två olika synsätt 15
- 5.4 Relationer mellan begrepp 16
- 5.5 Lingvistiskt baserad begreppsbildning 16
- 5.6 Begreppet Kunskapstäppa 17
- 5.7 Dynamisk omstrukturering av kunskap 19
- 5.8 Kunskapens Koloniträdgård - en mångfald av länkade täppor 19
- 5.9 Att strukturera kunskapskomponenter 20
- 5.10 Övergripande designprinciper för kunskapskomponenter 22
  - 5.10.1 Innehållet skall: 22
  - 5.10.2 Metadata, dvs. information om innehållet, skall: 22
  - 5.10.3 Strukturen, dvs. systemet av relationer mellan komponenter, skall: 23
- 5.11 Helpdesk - människa till människa via maskin 23
- 5.12 FrågeBaserad Inlärning (FBI) 23
- 5.13 Att designa sin egen lärostrategi 26
- 5.14 Kunskapsvaruhuset - ett arkiv av kunskapskomponenter 27

6 Begreppsorganisation och innehållspresentation 27
6.1 Bakgrund 27
6.2 Grundläggande designprinciper för begrepps-browsers 28
6.3 Fördelar med en begreppskarta 28
6.4 Conzilla - en första prototyp till begrepps-browser 28
6.4.1 Begrepps-editorn 29
6.4.2 Begrepps-browsern 29

7 Begrepps-browsning av komponentarkiv i multipel skala 30
7.1 Introduktion 30
7.2 Surfa kontexten 30
7.3 Visa innehållet under 1-dimensionell aspektfiltrering 30
7.4 Multipelt berättande 31
7.5 Två-dimensionell aspektfiltrering 31
7.6 Multipelt berättande baserat på förklaring och fördjupning 32
7.7 Ett längre sammanhängande exempel 32
7.8 Att manipulera kontexten för ett begrepp 36
7.9 Att tilldela innehåll till en kombination av olika begrepp 37
7.10 Matematiska begreppsrelationer på universitetsnivå 38

8 Referenser 41

APPENDIX

9 Notation för begreppsmodellering 44

10 IMS-projektet 45
10.1 Bakgrund 45
10.2 Intressenter, aktörer och grundläggande syfte 45
10.3 Marknadskrafternas uppgift 48
10.4 IMS-projektets framtida roll 49

# IT-baserade matematikverktyg
# några tidigare och några pågående KTH-projekt

Ambjörn Naeve
CID (Centrum för användarorienterad IT Design)
NADA (Institutionen för Numerisk Analys & Datalogi)
KTH (Kungliga Tekniska Högskolan)
100 44 Stockholm

[amb@nada.kth.se]

ABSTRACT: This paper presents a number of projects at KTH with the common aim to develop prototypes for IT supported tools with the potential of being useful in mathematics education and research. Some of these tools are included for historical reasons, while some represent ongoing work. The main tools of the latter category is *pdb*, which is a program for interactive studying of geometry, and *Conzilla*, which is a tool for conceptual browsing of archives of information components.

## 1 Bakgrund

De spektakulära framstegen med datorer - på både hårdvaru- och mjukvaruområdet - har skapat kraftfulla grafiska arbetsstationer med nya möjligheter att studera matematik på dator. Idag är det tydligt hur den datorbaserade animationsevolutionen har påverkat de flesta områden som har behov av att hantera matematik i olika sammanhang. Visualisering av strukturella förhållanden och förlopp har vuxit fram som ett område i sig - med tillämpningar inom såväl naturvetenskap och teknik som inom ekonomi och humaniora.

Under de senaste 15 åren har jag bedrivit ett arbete på KTH (Matematik & NADA) i syfte att utnyttja datorernas möjligheter att levandegöra matematiken. Arbetet har dels tagit formen av egen programmering, dels av handledning av ett antal projekt - framför allt olika elevprojekt inom högre årskurser av civilingenjörsutbildningen på KTH samt matematikerlinjens datalogigren på SU (Naeve [(30)]), men även examensarbeten för civilingenjörsexamen (Appelgren [(2)]), Nilsson & Palmér[pågående examensarbete på CID och Uppsala Universitet], samt forskningsprojekt som resulterat i doktorsexamen (Winroth[(38)]).

Jag har formulerat både visioner och specifikationer för olika verktyg som skulle vara användbara i dessa sammanhang och som samtidigt inte skulle vara orealistiskt komplicerade att designa och implementera. Gemensamt för samtliga projekt har varit deras syfte att presentera matematiska idéer på ett sätt som möjliggör en ökad förståelse av dem genom interaktion och experiment av olika slag. Verksamheten var från början fokuserad på geometri - vilket har resulterat i program som MapCon (1986), MacTapet (1987), Drawboard (1987) och MacDrawboard (1988) - men även funktionsbegreppet [HyperFlow (1987), MapAnalyze (1989)] och den elementära aritmetiken [PrimeTime 1992] har behandlats.

Arbetet har resulterat i ett antal prototypverktyg som kan användas som stöd inom både utbildning och forskning i matematik. Många av dessa prototyper är fortfarande i körbart skick, men somliga - t.ex. *MapCon*, *Drawboard* och *MacDrawboard* - tillhör numera historien och kan enbart köras på mer eller mindre utdöda datortyper och operativsystem.

Under senare år har arbetet kommit att inriktas mot att bygga datorstödda matematikverktyg som kan fungera tillsammans i en modulariserad och distribuerad konceptuell och experimentell läromiljö. Här kan framförallt nämnas följande projekt:

• Kunskapens Trädgård (interdisciplinärt projekt, pågår sedan 1996 på CID)
• Reflections (utvecklingsarbete vid CVAP, 1991-94)
• Projective Drawing Board (utvecklingsarbete vid CVAP, 1995-99)
• Begreppsnavigation (pågående utvecklingsarbete vid CID sedan 1998)

Nedan kommer dessa projekt - och de tidigare nämnda - att kortfattat beskrivas.

# 2 Tidiga utvecklingsprojekt av IT-stödda matematikverktyg

## 2.1 MapCon

Programmet *Mapcon* skapades genom ett teknologprojekt år 1986 som ett verktyg för interaktiva studier av s.k. konforma avbildningar, vars teori utgör en för tekniska tillämpningar synnerligen viktig del av teorin om komplexa funktioner. Programmet gav möjlighet att rita en kurva i ett fönster (det komplexa $z$-planet) och betrakta effekten på denna kurva när den avbildas till ett annat fönster (det komplexa $w$-planet) genom att applicera funktionen $w = f(z)$ på var och en av kurvans punkter. Programmet förutsatte alltså att kurvan approximeras med ett s.k. ändligt polygontåg (= sammanhängande ändlig följd av raka streck). Det hanterade även ett antal på förhand givna teststrukturer - som t.ex. rektangulära och cirkelsektorsformade rutnät - på motsvarande sätt.

*Mapcon* skrevs i Interlisp och snurrade på en (numera pensionerad) Lisp-maskin av typ Xerox-1108. Eftersom det inte har portats till någon annan plattform, så befinner det sig numera inte i körbart skick. En mer detaljerad beskrivning av programmets funkionalitet presenteras i projektrapporten [(5)].

På grund av den knappa tillgången på Lisp-maskiner (vi hade 2 stycken!) blev det aldrig möjligt att testa *Mapcon* som ett verklig hjälpmedel i matematikundervisningen. De enskilda tester som gjordes på speciellt intresserade ‘försökskaniner’ visade dock att programmet väsentligt kunde bidra till att öka förståelsen av hur olika typer av komplexa avbildningar uppför sig, och vad som händer i närheten av deras s.k. singulariteter - där de inte är definierade. Ett mycket populärt sätt att skapa överblick över en avbildnings verkningar var att lägga in det rektangulära respektive det concentriskt cirkelformade testnätet i $z$-planets fönster och se efter hur det avbildades i $w$-planets fönster.

## 2.2 MapAnalyze

*MapAnalyze* var ett liknande typ av verktyg som *Mapcon*, och skapades 1989 som ett studentprojekt av elever på matematikerlinjens årskurs 3 vid SU. *MapAnalyze* designades med avsikt att studera effekten av mångdimensionella reella matematiska transformationer på olika typer av geometriska objekt. Med sin tonvikt på studiet av linjära avbildningar (uttryckta på matrisform) var *MapAnalyze* tänkt som ett stödverktyg för kurserna i Linjär Algebra. Programmet skrevs i Scheme för Macintosh. Mera detaljer om programmets funktion finns i projektrapporten [(10)].

Tyvärr har även *MapAnalyze* gått ur tiden, och befinner sig numera - som nämndes ovan - i icke-körbart skick. På grund av en del problem med gränssnittet blev den framtagna prototypen för komplicerad att hantera för att kunna användas av “icke invigda”. *MapAnalyze* innehöll dock ett sätt att hantera avbildningar och deras sammansättningar som betonade fundamentala och notoriskt svårbegripliga aspekter av detta område. Med hjälp av enkla testobjekt - som t.ex. linjesegment, kvadrater och kuber - kunde programmet förmedla en visuell upplevelse av avbildningarnas verkan. Ett liknande program - med ökad tonvikt på gränssnittsproblematiken - borde därför vara av stort värde, exempelvis som IT-stöd för de grundläggande kurserna i Linjär Algebra vid universitet och högskolor.

## 2.3 MacTapet

*MacTapet* är ett program som ger möjligheter att studera tapetmönster på et interaktivt sätt. Programmet utvecklades 1988 av elever vid matematikerlinjens dataloggren, som skapade ett grafiskt gränssnitt till de 17 olika existerande tapetsymmetrierna (se [(28)]) vars matematiska koordinatbeskrivningar jag tillhandahöll.

*MacTapet* har med tiden blivit ett av de mest använda av mina matematiska prototypverktyg. Det fungerar (på Macintosh) än idag, och har används av både forskare, lärare och konstnärer som ett sätt att utforska den tvådimensionella mönstervärldens dynamik. Programmet användes från början som grund för den experimentella mönsterverkstaden i *Kunskapens Trädgård* (se [(30)] och [(22)]), men är numera i detta sammanhang ersatt av *Kali* - ett program som har utvecklats av *Geometry Centre* vid University of Michigan i USA [www.geom.umn.edu].

I samband med projektet *Förstklassig Matematik*, som beskrives nedan, har jag även använt *MacTapet* på låg-och mellanstadiet i min dotters skolklass. Jag har t.ex. låtit barnen skapa egna mönster som vi sedan har analyserat och diskuterat - både samlat i hela klassen (genom att projicera upp datorskärmen på tavlan) och i mindre grupper. Tack vare *MacTapet* har barnen fått möjlighet att möta en form av matematik som inte handlar om räkning utan i stället visar upp en annan typ av struktur. I min dotters klass - som till övervägande del består av barn med s.k. invandrarbakgrund - visade sig mönstermatematiken ha en klart stimulerande verkan på flickornas intresse för matematik - speciellt de som har invandrarbakgrund - genom att vi kunde synliggöra den matematiska strukturen i det traditionella kvinnliga textilhantverket och t.ex. jämföra symmetrityper från deras respektive ursprungsländer. Att på detta sätt fokusera på matematiken i dessa traditionellt kvinnliga uttrycksformer bidrog tydligt till att höja det matematiska självförtroendet hos flickorna. Här finns en mycket intressant källa att ösa ur för att sprida intresse för matematik till dessa av tradition matematiskt undernärda grupper. Programmet *MacTapet* finns dokumenterat i projektrapporten [(11)].

## 2.4 Drawboard

### 2.4.1 De grundläggande idéerna

*Drawboard* är samlingsnamnet på ett dynamiskt geometrisystem för studiet av plana geometriska konstruktioner som jag har arbetat med i omgångar sedan mitten av åttiotalet. Målet har varit att levandegöra geometriska konstruktioner och satser genom att göra dem dynamiska och därmed möjliga för användaren att interagera med och förändra på olika sätt.

Konstruktionerna utspelar sig i det s.k. *projektiva planet*, vilket är en utvidgning av det (vanliga) *euklidiska planet* på sådant sätt att två linjer alltid skär varandra i en punkt. Skärningspunkten säges befinna sig i 'oändligheten' om linjerna är parallella. *Drawboard* använder sig helt och hållet av projektiva begrepp för sin underliggande matematiska representation av konstruktionerna.

Varje geometrisk konstruktion i *Drawboard* har sin 'historia' som kan betraktas som ett växelspel mellan slumpmässiga val (t.ex. välj två punkter *P* och *Q*) och kanoniska konstruktioner (t.ex. drag linjen *PQ*). Ett geometriskt objekt kan därför tillordnas en mängd 'barn' och en mängd 'föräldrar' på ett naturligt sätt. I exemplet ovan är t.ex. linjen *PQ* ett barn till både punkten *P* och punkten *Q* medan båda dessa punkter är föräldrar till linjen *PQ*.

För att kunna dra i en geometrisk konstruktion och förändra den interaktivt måste man kunna förändra dess konstruktionshistoria på ett konsistent sätt, dvs så att en ändring vid en viss tidpunkt i konstruktionen fortplantas framåt i tiden till samtliga efterföljande generationer av barn, barnbarn, barnbarnsbarn etc. För att kunna genomföra en sådan uppdatering krävs det tillgång till en hel hierarki av olika koordinatsystem som håller bokföring på läget av varje objekt relativt sina föräldrar. Detta effektueras av den projektiva representationstekniken som kan användas för att utrusta varje nyskapat objekt med sitt eget interna koordinatsystem som senare används för att beskriva objektets barn om sådana skapas. För en närmare beskrivning av dessa idéer, se Naeve [(26)].

På detta sätt blir det alltså möjligt att dra i punkter och linjer och flytta dem inom ramarna för de begränsningar som är inbyggda i deras respektive konstruktion. Exempelvis kan en punkt som är vald med begränsningen att den måste ligga på en viss linje enbart flyttas längs med denna sin föräldralinje, medan en fri punkt naturligtvis kan flyttas utan inskränkningar. Härigenom kan man experimentera med geometriska konstruktioner på ett interaktivt sätt. Detta har visat sig vara en utomordentligt kraftfull metod att levandegöra geometriska strukturer och satser, eftersom man direkt kan uppleva deras dynamiska struktur! Man kan se direkt vad som händer när man drar i saker, och man upplever satsens dynamiska innehåll utan att man behöver verbalisera varje enskild del av den geometriska konfigurationen.

Som ett led i arbetet med *Drawboard* utvecklade jag även på vår Interlisp maskin (Xerox-1108) under åren 1984-87 ett antal interaktiva geometriska konstruktioner som illustrerar teorem ur den klassiska projektiva geometrien. Dessa var försedda med ett användargränssnitt som gjorde det möjligt att ändra indata i den konstruktion som teoremet i fråga uttalar sig om och sedan observera effekterna av denna ändring på konstruktionen. Till de geometriska teorem som 'interaktiverades' på detta sätt hör bl.a. de berömda saterna av *Desargues*, *Pappus*, *Pascal*, *Brianchon* och *Steiner*. Detta arbete finns närmare beskrivet i [(26)].

#### 2.4.2 Den första prototypen

Den första prototypen av *Drawboard* utvecklades år 1987 i den tidigare omtalade Xerox-interlisp miljön av elever vid matematikerlinjens datalogigren på Stockholms Universitet och finns närmare beskrivet i projektrapporten [(37)]. Programmet var begränsat till att hantera geometriska relationer mellan punkter och linjer, och eftersom Xerox-1108 maskinen saknade flyttalsprocessor[1] så var uppdateringen av förändringar olidligt långsam, men det hela fungerade i princip. På grund av långsamheten i beräkningarna fick man dock som användare ingen känsla

av kontinuitet i förändringarna när man drog i någon del av en viss konstruktion. Trots att programmet fungerade i princip blev det alltså uppenbart att det behövde porteras till en snabbare beräkningsmiljö för att kunna förmedla kontinuitetsupplevelsen i förändringarna på ett effektivt sätt.

#### 2.4.3 MacDrawboard - en efterföljare

Nästa version av programmet hette *MacDrawboard* och utvecklades ett år senare (1988) för Macintosh av 8 teknologer i 3:e årskursen på datatekniklinjen på KTH. Teknologerna byggde grafik och användargränssnitt i MacApp (Objekt-Pascal) omkring en geometrisk beräkningskärna som jag hade skrivit i C. Detta representerade min första verkliga kontakt med objektorientering och förberedde mig för senare äventyr i C++ på nittotalet. Kommunikationen mellan de två språkvärldarna ställde emellertid till med allvarliga problem - framförallt med inläsning av data via C-funktionen scanf, vilken helt enkelt vägrade att fungera från MacApps horisont. Situationen räddades genom en heroisk insats av Mats Danielsson, som satte ihop ett ‘superhack’ vilket hanterade datatransporten på sitt eget sätt. Detta medförde att *MacDrawboard* blev ‘hårt’ beroende av det operativsystem som det skapades i (Finder version 6), och härigenom har programmet inte kunnat överleva på samma sätt som t.ex. *MacTapet.*

Jag har använt *MacDrawboard* vid ett flertal olika kurser och seminarier om geometri - framför allt i slutet av åttio- och början av 90-talet. Numera är programmet omöjligt att köra på Nadas Macintosh-datorer, men jag har fortfarande kvar en gammal Mac-II från 1987 där jag kör Finder-version 6, och här kan jag fortfarande köra programmet. En närmare beskrivning av hur *MacDrawboard* fungerar finns i projektrapporten [(1)].

Sedan 1991 har *MacDrawboard* även använts inom forskning och utveckling, bl.a. av fysikern och uppfinnaren Lloyd Cross i Kalifornien, som ett hjälpmedel för att modellera optiska system med höga krav på precision. Även han har kvar en gammal Mac i sitt laboratorium för att kunna köra programmet.

Flera framstående bildanalysforskare - bland vilka kan nämnas fransmannen Olivier Faugeras - introducerades till den projektiva geometriens kraftfulla möjligheter genom olika demonstrationer av dynamiska konstruktioner i *Drawboard* respektive *MacDrawboard* - eller tidigare föregångare till dessa i Interlisp-miljön. I Faugeras fall skedde det vid ett besök i det nybildade bild-och-grafik laboratoriet vid forskningsgruppen i bildanalys (CVAP) på NADA i december 1985.

#### 2.4.4 Projective Drawing Board - en modern vidareutveckling

De grundläggande idéerna bakom *Drawboard* och *MacDrawboard* har under senare år (1995-99) vidareutvecklats av Harald Winroth på CVAP - som ett doktorandprojekt under min handledning. Detta har resulterat i programmet *pdb* (*Projective Drawing Board*) samt en avhandling som Harald disputerade på i mars 1999 (se Winroth [(38)]). En närmare diskussion av *pdb* och det övergripande dynamiska geometriarbetet på CVAP återfinns i nästa kapitel.

---

1. Xerox ville vid denna tid (1987) ha ungefär 240.000 kronor för att sätta dit en sådan.

### 2.5 MacFlow - en grafisk programmeringsmiljö

*MacFlow* är namnet på ett program för grafisk programmering designat i första hand för utbildningsändamål. Det huvudsakliga syftet är att skapa en känsla för vad en algoritm egentligen är för något - för att därigenom kunna öka förståelsen för hur t.ex. en dator egentligen utför sina beräkningar. I programmet skapar användaren ett flödesdiagram med lådor som representerar antingen variabler eller operationer av aritmetisk- eller jämförelsetyp. Lådorna kopplas sedan samman genom två olika typer av flöden - dataflödet respektive kontrollflödet. Dataflödets gråa streckade linjer visar hur data transporteras genom beräkningsprocessen och kontrollflödets svarta heldragna linjer visar i vilken ordning och under vilka förhållanden som beräkningarna utförs. Detta är en kraftfull blandning som skapar en direkt visuell kontakt med motsvarande beräkningsförfarande. Algoritmerna kan sedan exekveras steg för steg varvid man ser hur data flyttas runt och adderas, subtraheras, etc, samt hur kontrollen stegvis flyttas mellan olika operatorboxar beroende på vilka villkor som är uppfyllda.

*MacFlow* hjälper användaren att skapa en mental bild ('gestalt') av en beräkningsprocess genom att visualisera den på ett överskådligt sätt, dvs i ett enda diagram. Programmet är framförallt avsett för barn och ungdomar och jag har använt det på mellanstadiet i min dotters klass inom ramen för det ovan nämnda projektet *Förstklassig Matematik*, som beskrivs mer detaljerat nedan. En närmare beskrivning av hur programmet fungerar finns i projektrapporten [(4)].

### 2.6 PrimeTime

*PrimeTime* är ett program som utvecklades av elever vid matematikerlinjens datalogigren på Stockholms Universitet och som bygger på en idé av Örjan Ekeberg och mig från mitten på åttiotalet. Programmet utnyttjar den välkända entusiasmen för dataspel i syfte att lära ut grundläggande räknefärdigheter. Spelet går ut på att skjuta ner två anfallande missiler som är behäftade med var sitt heltal. Låt oss anta att det står 7 på den ena missilen och 5 på den andra. Om spelet är inställt på addition, så träffar man missilerna genom att skjuta iväg (= mata in) talet 12. Matar man in något annat tal så blir man själv träffad av missilierna. Om spelet istället är inställt på multiplikation så vinner man naturligtvis genom att mata in talet 35. Programmets funktioner beskrivs utförligt i projektrapporten [(6)].

Trots en del brister i sin nuvarande utformning så är det klart att principerna bakom *PrimeTime* fungerar i praktiken. En vidareutveckling av gränssnittet skulle göra programmet till ett intressant hjälpmedel i undervisningen, som väsentligt borde kunna förkorta inlärningstiden för många elever vad gäller grundläggande aritmetiska färdigheter.

## 3 Geometriska IT-verktyg - utvecklingsarbetet på CVAP

### 3.1 Bakgrund

CVAP (Computational Vision and Active Perception) gruppen är en forskningsgrupp på NADA som bedriver forskning inom området datorseende (= bildanalys) och robotik. Den leds av professor Jan-Olof Eklundh, som är en internationellt ledande forskningsprofil inom detta område. Jag har arbetat på CVAP sedan 1984 - först som doktorand och sedemera som forskare. Jag dis-

puterade i Geometri 1993 på en avhandling med titeln *Focal Shape Geometry of Surfaces in Euclidean Space* [(27)].

I många olika forsknings- och utbildningssammanhang har man idag tillgång till olika typer av geometriska modelleringsverktyg med hög grad av specialisering och “egenartat beteende” av olika slag. Dessa verktyg är till stor del resultatet av ett antal ad-hoc-mässiga val och förenklingar som har givit upphov till skillnader och inkonsistenser (inkompatibiliteter) i representationerna - vilket har omöjliggjort en synergistisk samverkan mellan motsvarande datorprogram.

Under mina 15 år på CVAP har jag genomfört och handlett ett antal projekt inom området geometrisk modellering i syfte att utveckla forsknings- och utbildningsmässigt intressanta geometriska modelleringsverktyg av olika slag. Det har dels rört sig om de olika teknolog- och studentprojekt som beskrivits i kapitel (2), dels har det varit fråga om handledning av examensarbetare och doktorander.

Dessa projekt - med det övergripande arbetsnamnet *Geometric Toolbox* - syftar långsiktigt till att bygga upp en interaktiv geometrisk experimentell miljö, där geometriska egenskaper hos såväl enskilda objekt som grupper av sådana kan undersökas och deras beteende vid variation av olika parametrar simuleras och studeras på ett enhetligt sätt. I detta syfte används den underliggande projektiva representationsteknik som beskrivits ovan, eftersom den tillåter en inbäddning av samtliga typer av geometrisk information mot en gemensam bakgrund. Detta visar sig har stora fördelar - speciellt vid analys av konfigurationer som innehåller information både i form av olika egenskaper hos de enskilda objekten samt i form av relationer mellan dessa.

Bland verktygen finns även ett geometriskt objektbibliotek, dvs en samling av kompatibla och återanvändbara geometriska stukturer och algoritmiska komponenter [(26)], vilka för närvarande är implementerade med hjälp av *Mathematica*™ [(39)]. Dessa verktyg kan vara till nytta och användning inom en mängd olika områden som t.ex. datorseende, robotik och datorgrafik. Olika geometriska experiment kan t.ex. kopplas ihop och de relevanta parametrarna kan manipuleras på ett matematiskt kontrollerat interaktivt sätt. Intresset för att utföra denna typ av experiment - där man kombinerar tunga beräkningar med omedelbar visualisering av resultatet växer stadigt inom det beräkningsgeometriska forskningskollektivet - i takt med att sådana möjligheter utvecklas och ger möjligheter att testa olika typer av algoritmer utan att behöva bygga upp testmiljön från ‘scratch’. Detta bibliotek utgjorde grunden för de simuleringsexperiment som redovisas i t.ex. [(29)] och [(31)].

Denna utveckling drivs i sin tur av den enorma ökningen av tillgänglig beräkning- och presentationsskraft som manifesteras i en ständig ström av nya och mer kraftfulla hårdvarukomponenter, vilket har lett fram till att de s.k. avancerade arbetsstationerna håller på att etablera sig på var och ens skrivbord. Det har numera blivit möjligt att simulera en stor mängd av komplicerade geometriska konfigurationer och erhålla information ‘on-line’ med direkt relevans för förståelsen av det underliggande problemet. Möjligheterna att på detta sätt utvidga sin intuitiva uppfattning om ett problem kommer att ha en genomgripande effekt på forsknings- och undervisningsmetodiken i framtiden.

## 3.2 Reflections - ett system för simulering av vågfronter

*Reflections* är namnet på ett programpaket som har utvecklats för att studera 3-dimensionella ytor interaktivt - med speciell tonvikt på deras optiska reflektionsegenskaper. Via sina normaler betraktas vissa ytor som vågfronter vars reflektioner i olika speglade ytor sedan kan beräknas och presenteras på skärmen. *Reflections* utvecklades i CommonLisp på en maskin av märket *Symbolics*™ och implementerades som ett examensarbete av Johan Appelgren [(2)] under min handledning. Jag använde mig av programmet för att illustrera delar av min doktorsavhandling [(27)], som behandlade närbesläktade områden inom geometrin.

## 3.3 Dynamisk Projektiv Geometri

Som nämnts ovan har *Drawboard* under åren 1995-99 vidareutvecklats av *Harald Winroth* på CVAP under min handledning - vilket har resulterat dels i programmet *pdb* (*Projective Drawing Board*), dels i en doktorsavhandling med titeln *Dynamic Projective Geometry* [(38)] som lades fram på KTH i mars 1999. *Harald Winroth* har tagit fasta på de grundläggande strukturerna bakom *Drawboard* och ovanpå dessa skapat en datalogisk struktur som har förbättrat och byggt ut programmet på flera väsentliga sätt, varav följande kan omnämnas här:

För det första har hastigheten i den dynamiska uppdateringen mångdubblats, vilket har skapat förutsättningar för den dynamiska kontinuitet som *pdb* uppvisar. Detta har varit ett viktigt designmål för projektet.

För det andra har användargränssnittet förbättrats och byggts ut på ett flertal olika sätt. Här kan framförallt nämnas olika typer av drag-and-drop funktionalitet samt kontextkänslighet för muspekaren vad gäller olika delar av konstruktionen, vilket skapar förutsättningar för ett mera lätttfattligt och överskådligt sätt att arbeta med programmet.

För det tredje har den grafiska presentationen av en geometrisk konstruktion kompletterats med en logisk presentation av konstruktionen i form av en s.k. struktur-graf som visar beroendena mellan de olika ingående delarna. Detta var planerat - men ej implementerat - i *MacDrawboard.*

För det fjärde kan *pdb* även hantera kägelsnitt, dvs ellipser, parabler och hyperbler, medan MacDrawboard enbart kunde hantera punkter och linjer. Hanteringen av kägelsnitt sker dessutom på ett för geometriska modelleringsprogram ovanligt matematiskt sofistikerat sätt - med bl.a. en typ av representation av samspelet mellan reella och komplexa koordinater, där t.ex. punkter och linjer byter färg när de övergår från att vara reella till att bli komplexa. Denna typ av representation går tillbaka till *Felix Klein* [(19)], men har såvitt jag vet inte används tidigare i samband med datorgeometri. Representationen gör det möjligt att visa hur samspelet mellan komplexa koordinater - via s.k. komplexkonjugering - leder tillbaka till reella geometriska objekt - t.ex. vid s.k.'polarisering' i en ellips av en inre liggande punkt till densamma.

För det femte kan logiken i en pdb-baserad konstruktion förändras och olika begränsningsvillkor mellan punkter, linjer och kägelsnitt både införas och avlägsnas. Härigenom kan man utforska både nödvändighet och tillräcklighet i de villkor som måste gälla för att ett visst geometriskt samband skall vara giltigt.

### 3.4 Geometrisk Algebra

Programmet *pdb* representerar i många delar “state of the art” inom utvecklingen av plangeometriska modelleringsverktyg. Det innehåller dessutom grundstrukturerna för ett framtida 3-dimensionellt matematiskt geometriverktyg - *DrawingSpace* - som har förutsättningar att väsentligt flytta fram positionerna på detta område.

Den typ av underliggande representationsteknik som skulle krävas för att kunna hantera samspelet mellan det reella och komplexa i tre dimensioner finns att tillgå inom ett område som kallas geometrisk algebra (eller Cliffordalgebra) och som går tillbaka till 1800-talet då det skapades av *Hermann Grassmann* [(13), (14)] och *William K. Clifford* [(7), (8)]. Under de senaste 30 åren har den geometriska algebran vitaliserats och utvecklats vidare av bl.a. *David Hestenes* [(15),(16)]. För en synnerligen intressant översikt på Internet av denna forskning se [http://modelingnts.la.asu.edu/GC_R&D.html].

Sedan 1995 har jag - i samarbete med *Lars Svensson* på Matematikinstitutionen vid KTH - varit engagerad i forskning inom området geometrisk algebra. Vårt beräkningstekniska ramverk - kallat *Geo-MAP* - för gemensam behandling av de vanligast förekommande typerna av geometri - euklidisk, affin och projektiv - publicerades nyligen i [(32)] och presenterades vid den 5:e internationella konferensen om Cliffordalgebror och deras användning inom fysiken och ingenjörsvetenskapen som ägde rum i Ixtapa, Mexico i juli 1999.

## 4 Förstklassig Matematik - ett projekt i min dotters klass.

### 4.1 Bakgrund

År 1991 började min dotter Ylva i första klass på St Eriks Katolska Skola i Stockholm. Tack vare intresse och tillmötesgående från klassens låg- respektive mellanstadielärare[1] fick jag tillfälle att arbeta med barnen och “prata matematik” med dem en timme i veckan ungefär. Projektet, som gick under namnet *Förstklassig Matematik*, gav mig möjligheter att testa ett stort antal av de pedagogiska fantasier som jag utvecklat under min mångåriga verksamhet som matematiklärare på KTH (1967-90).

### 4.2 Matematiken som logiskt uttestade fantasifoster

Man brukar säga att en matematiker är en person som tänker i en dag för att slippa räkna i en timme. Ändå marknadsförs skolmatematiken nästan uteslutande i termer av räkning. Mitt förstklassiga matematikprojekt har haft som övergripande målsättning att vända denna bild av matematiken och etablera tänkandet som grundprincip inom ämnet. “Tänk först och räkna sen” skulle man kunna sammanfatta projektets syfte.

Tidiga möten med matematiska strukturer har varit ett annat genomgående tema. Jag har försökt lyfta fram matematiken som studiet av all struktur som det mänskliga tänkandet är mäktigt att uppfatta, och försökt förmedla en syn på matematiska strukturer som en sorts *logiskt uttestade fantasifoster.* Med denna utgångspunkt gäller det alltså att tidigt ge eleverna möjlighe-

1. Anette Philipsson och Kristina Lindgren.

ter att fostra sina egna matematiska fantasier, något som alltför sällan erbjuds i den vanliga skolundervisningen.

Det har varit ett stort privilegium att få arbeta med barnen i Ylvas klass och få tillfälle att presentera en matematikers motbild av matematiken i förhållande till det räknedominerade perspektiv på ämnet som skolan normalt tillhandahåller. Detta arbete har stärkt mig i min övertygelse att matematiken kan - och bör - presenteras för barn på ett fundamentalt annorlunda sätt än vad som är fallet i skolan idag, om vi vill grundlägga den sortens intresse som växer med åren och leder till högre studier i matematik. Sett ifrån detta perspektiv är det snarast ett hälsotecken att så många utbildningsplatser gapar tomma på matematiklärarutbildningarna. Vem är idag egentligen villig att ta på sig ansvaret för att lära ut den upprepningsfixerade räkneträning som matematikundervisningen har utvecklats till? Helgdagsmatematiken lyser med sin frånvaro, och vardagsmatematikens lockelser utövar inte ett tillräckligt starkt inflytande på (var)dagens ungdom!

### 4.3 Det matematiska smörgåsbordet

Det matematiska smörgåsbord som jag presenterade för Ylva och hennes klasskamrater under dessa 6 år finns närmare beskrivet i [(30)]. Här kan nämnas att detta smörgåsbord inkluderade rätter som t.ex.

- att summera de första 99 heltalen (1+2+3+ ... +97+98+99).
- att lista ut hur många olika halsband som kan tillverkas av 4 olika glaspärlor.
- att undersöka på hur många olika sätt som klassen - med 25 stycken elever -. kan ställa sig i kö
- att inse storleken av detta tal (25!) genom att köra Mathematica på tavlan (via OH-projektor och LCD-skärm).
- att bygga de 5 platonska kropparna och utforska deras dualitet via kopplingen till Eulers formel.
- att utforska strukturen hos olika typer av mönster - med hjälp av programmet MacTapet.
- att klippa upp Möbiusband i olika proportioner och försöka förstå i förväg vad som blir resultatet.
- att bestämma villkor för när en given graf kan ritas utan att lyfta pennan från papperet (s.k. Euler-grafer).
- att räkna i olika baser, och t.ex. visa hur vi skulle ha räknat om vi bara hade haft ett finger på varje hand.
- att lära sig skilja mellan ett tal och dess siffror med avseende på en viss bas.
- att beräkna hur många luftmolekyler som får plats i en tom mjölkförpackning (Avogadros tal = $6{,}02 \times 10^{23}$).
- att arbeta med lådor i stället för variabler och lösa ekvationer genom att lista ut vad som finns i en viss låda.

Flera av de ämnen som jag arbetade med finns beskrivna i en utmärkt liten bok av Kristin Dahl som heter *Matte med mening* [(9)] och som publicerades 1995. Författaren har samlat en mängd intressant utgångsmaterial för matematiska diskussioner som med fördel kan presenteras för barn i en tidig ålder (= på låg-och mellanstadiet) i syfte att stimulera deras intresse för och nyfikenhet på matematik. Det faktum att så mycket av hennes material överensstämmer med det som jag tog upp - trots att jag inte blev bekant med hennes bok förrän långt senare - vittnar om att det börjar växa fram en insikt om viken typ av s.k. “högre matematik” som kan (och bör) föras ner i åldrarna för att förankra och befästa det matematiska intresset innan det tynar bort under inflytandet av den upprepningsfixerade räkningsexercisen. Det är sorgligt att den lilla boken ska betinga ett pris som i dagens skolfinansiella läge omöjliggör en större spridning. Den är helt enkelt “för dyr att köpa in i klassupplaga” som Ylvas lärare uttryckte saken.

Trots att mitt matematiska smörgåsbord skapade ett viss utrymme för fantasi och kreativitet, så lyckades det naturligtvis inte vända den förhärskande pedagogiska principen om inlärning genom upprepad imitation: räkna samma sorts tal om och om igen - ända till du begriper! Den djupt rotade rädslan för att experimentera med nya begrepp inom den tidiga matematikutbildningen hänger samman med den "algoritmiska fundamentalism" som fortfarande tycks dominera pedagogiken och som diskuteras närmare i [(30)]

### 4.4 En variabel som en låda med ett namn och (eventuellt) ett innehåll

Problemet i den tidiga matematikutbildningen är inte eleverna utan lärarna. För att uttrycka sig rakt på sak: Eleverna har inte ännu lärt sig inse att de inte förstår matematik, men det har däremot både lärarna och föräldrarna gjort. Och denna attityd är mycket smittsam. Den konversation som återges i Fig. (1) är nog tyvärr ganska typisk i detta sammanhang.

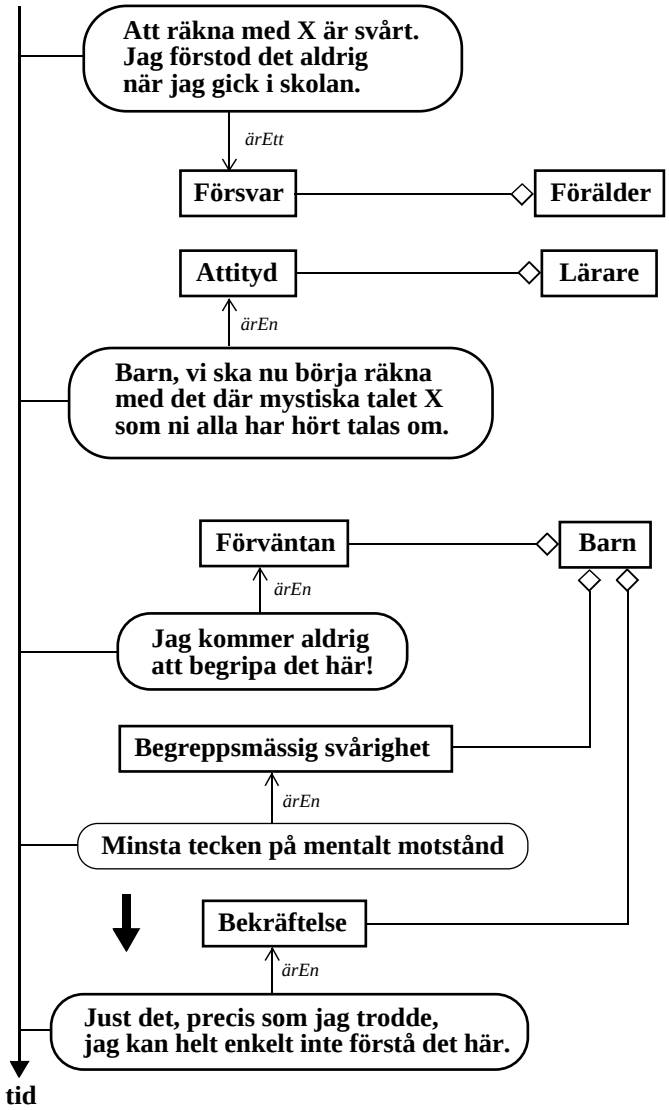


**Fig. 1. X-istentiella förväntningar.**

Att lära barnen “räkna med x” bör hanteras på ett helt annat sätt. Jag tog upp detta ämne vid ett flertal olika tillfällen under mellanstadiet, och jag använde analogin mellan en variabel och en låda som utgångspunkt. Båda har ett namn och eventuellt även ett innehåll, ifall vi har givit variabeln ett värde - dvs om vi har lagt något i lådan. Med denna metafor kan man betrakta en ekvation som ett samband mellan innehållet i ett antal olika lådor. Att lösa ekvationen blir då liktydigt med att lista ut vad som finns i (vissa av) lådorna.

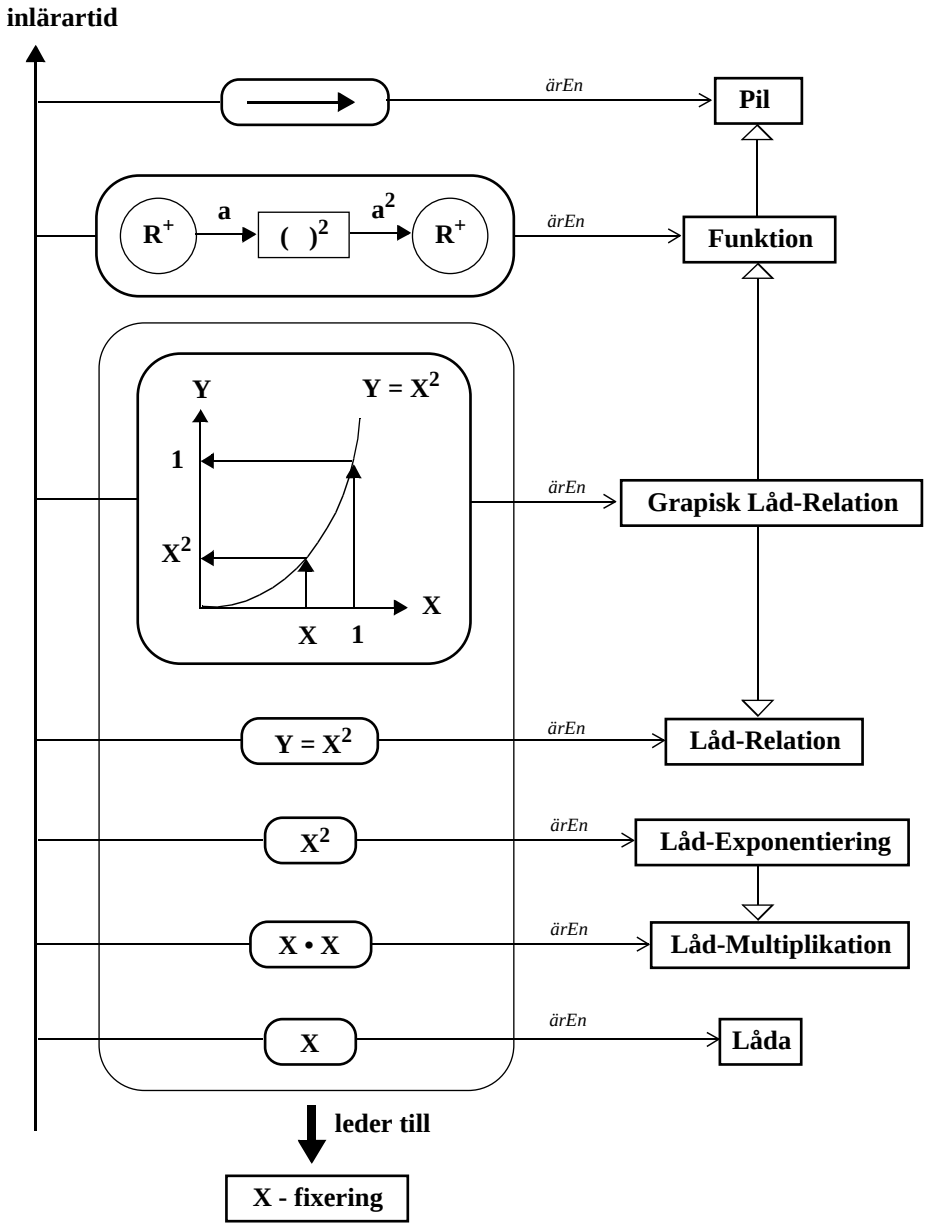


**Fig. 2. Begreppet variabel - traditionell respektive lådmässig approach.**

Detta sätt att betrakta variabler är väldigt konkret och visade sig vara lätt att förstå för barnen. Det motsvarar dessutom exakt det sätt på vilket symboliska matematikprogram, som t.ex. Mathematica™ behandlar variabler. När man matar in ett namn på en variabel till Mathematica,

svarar programmet med att ange variabelns värde (= lådans innehåll) - ifall något sådant värde har tilldelats variabeln (= lagts i lådan). I annat fall svarar Mathematica med att ange variabelns namn. Genom att demonstrera detta beteende - och förklara det i termer av att “räkna med lådor” - lyckades jag förankra idén om vad en variabel är för något hos var och en av de 25 eleverna - vid en tidpunkt då de gick i 4:e klass, dvs då de var ungefär 10 år gamla.

Figur (2) illustrerar det traditionella respektive det lådbaserade sättet att tänka på variabler och funktioner - två grundläggande begrepp som måste bemästras för att skaffa sig tillträde till matematikens högre abstraktionsnivåer. Den vänstra delen av figuren visar de olika abstraktionsnivåer som uppträder i den traditionella förklaringsmodellen, medan den högra delen visar hur motsvarande begrepp beskrivs i ‘lådmodellen’. För en närmare förklaring av den grafiska notation som används i figuren, se appendix.

# 5 Kunskapens Trädgård

## 5.1 Bakgrund - CID

CID (Centrum för användarorienterad IT Design) är ett s.k. kompetenscentrum vid NADA (institutionen för Numerisk Analys och Datalogi) på KTH. Där sammanförs en unik blandning av naturvetenskaplig, datalogisk, estetisk, beteendevetenskaplig och pedagogisk kompetens inom olika projekt som går ut på att utveckla IT baserade hjälpmedel av olika slag. På CID bedrives sedan 1996 ett projekt som kallas Kunskapens Trädgård - där jag fungerar som en sorts övergripande ‘trädgårdsarkitekt’. Projektet syftar till att främja en tvärvetenskaplig förståelse av olika fenomen och begrepp genom att beskriva deras inbördes relationer såväl som deras utveckling genom olika tider och kulturella epoker. Den prototyp som har tagits fram är inriktad på att beskriva och illustrera olika samband mellan matematik och musik, med inriktning i första hand mot nyfikna högskolestudenter inom båda dessa områden. Denna prototyp finns beskriven i [(22)] och [(30)].

De erfarenheter som vunnits genom projektet Kunskapens Trädgård har bidragit till att CID beslutat vidga arbetet till att omfatta interaktiva lärmiljöer i allmänhet. En viktig målsättning härvidlag är att bidra till utvecklingen och spridningen av analysmetoder, designprinciper och tillämpningar för skapandet av interaktiva lärmiljöer på Internet vilkas struktur och innehåll är modulariserade i enlighet med relevanta internationella standarder. På CID genomförs därför en kontinuerlig bevakning av sådana standardiseringsprojekt i syfte att bidra till utvecklingen av pedagogiskt och estetiskt genomtänkta exempel på vad vi kallar ‘kunskapskomponenter’ - vilka är designade i enlighet med relevanta internationella standarder. För närvarande arbetar vi med matematiken och litteraturen som exempelområden - med inriktning mot geometri i det förra respektive Strindbergs författarskap [(3)] i det senare fallet.

Avsikten - i det längre perspektivet - är att kunna åstadkomma en övergripande prototyp för en modulariserad läromiljö där man kan testa olika principer för beskrivning, sökning och hopkoppling av kunskapskomponenter utifrån ett användarorienterat perspektiv. För att kunna bli framgångsrik måste en sådan prototyp för utveckling av interaktiva läromiljöer kunna dra nytta av en mängd olika synergieffekter, dvs samspel mellan olika typer av kompetenser. På CID finns tillgång till ett antal olika kompetensområden som är av strategisk betydelse för en sådan utveckling:

• Estetisk kompetens, som har gjort formgivningen av Kunskapens Trädgård till ett erkänt gott exempel på design av ett interface till en länkad och modulariserad inlärningsmiljö med såväl teoretiska som experimentella inslag.

• Pedagogisk kompetens, som har gjort innehållet i Kunskapens Trädgård begripligt och stimulerande långt utanför den ursprungligen tilltänkta målgruppen.

• Datalogisk kompetens, som möjliggjort ett utnyttjande av många nya och spännande möjligheter hos den framväxande programmeringstekniken på Internet, såväl vad gäller modularisering och plattformsoberoende (Java, servlets) som hantering av informationsbeskrivning (XML).

• Beteendevetenskaplig kompetens, som borgar för att människan inte glöms bort mitt i all nödvändig maskinhantering. CIDs användarorienterade profil utgör här en viktig sammanhållande kraft.

## 5.2 Att vända ut och in på lärandet

En *läromiljö* är en miljö där man kan lära sig något. En läromiljö består av en utläromiljö (teaching) och en inläromiljö (learning) som samverkar med varandra. En interaktiv läromiljö är en läromiljö som tillåter såväl aktion som reaktion i förhållande till både ut- och in-lärandet. En IT-stödd interaktiv läromiljö arbetar med IT för att stödja och utforska olika former av interaktiva läromiljö-möjligheter.

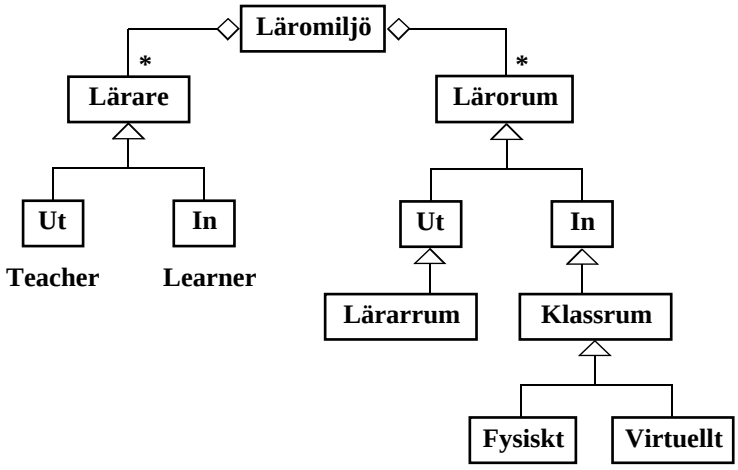


**Fig. 3. Den traditionella strukturen hos en läromiljö.**

Den traditionella strukturen hos en läromiljö är illustrerad i Figur (3). Strukturen karaktäriseras av en strikt uppdelning i utläromiljöer och inläromiljöer. Notera att även ett s.k. virtuellt klassrum kan betraktas som en digital form av fortsättning på detta traditionella tänkande.

## 5.3 Komponenter och läroupplevelser - två komplementära synsätt

En viktig del av projektet Kunskapens Trädgård är design av vad vi kallar *kunskapskomponenter*. Dessa kan sedan kan kopplas ihop och komponeras till såväl traditionella kurser som andra former av *läroupplevelser* av olika slag. Kunskapskomponenter kan bestå av såväl modulära IT-verktyg av den typ som beskrivits ovan, samt begreppspresentationer i multi-skala (= med olika upplösning) uppbyggda genom s.k. *multipelt berättande*.

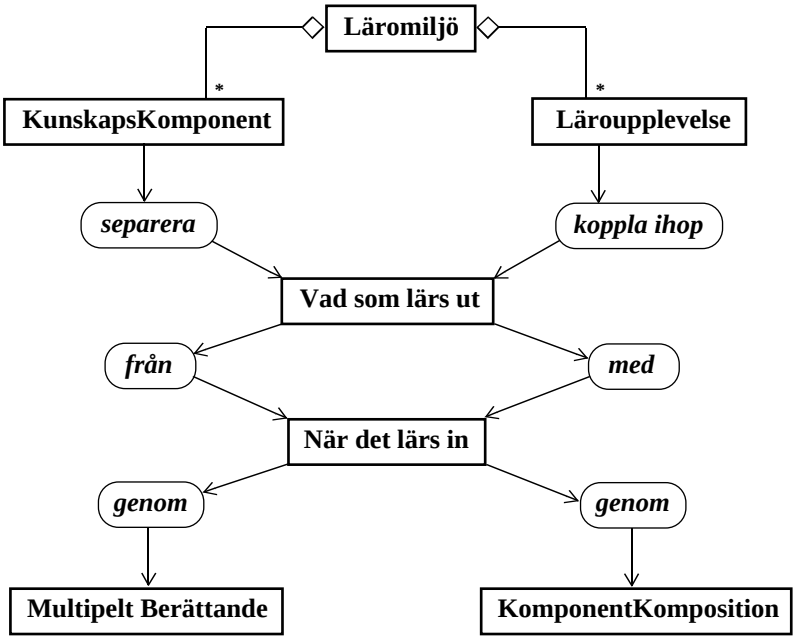


**Fig. 4. Kunskapskomponenter och läroupplevelser - två komplementära synsätt**

Man kan betrakta en läromiljö som ett sorts protokoll för separation av utlärande och inlärande, mellan ”vad som lärs ut” (innehåll) och ”när det lärs in” (sammanhang). En traditionell kurs strävar efter att *koppla ihop* och anpassa innehållet med inlärningssammanhanget. Den grundläggande idén när man designar en kunskapskomponent är i stället att *separera* mellan innehåll och inlärningssammanhang [(30), sid 89-93]. Det gäller här att beskriva begreppen på många olika nivåer på samma gång. Den moderna tekniken med hypertext gör att man sedan kan låta användaren själv styra vilken upplösning (= detaljrikedom) av berättelsen som han eller hon vill ta till sig (ungefär som ett dataspel, vilket ger möjlighet till individuellt anpassade svårighetsgrader och preferenser).

Jag brukar säga att de lärare som tror att de kan ersättas av datorer bör se till att detta går i uppfyllelse! Vårt arbete med interaktiva läromiljöer handlar inte om att “frysa levande kurser” i datoriserad form (som t.ex. totalt videobandade kurser). I stället vill vi utforska möjligheterna att designa kraftfulla och flexibla kunskapskomponenter, som ger levande lärare möjligheter att förbättra innehållet i sina egna berättelser genom att illustrera eller animera det som de har att säga på ett stimulerande och intresseväckande sätt, och som samtidigt ger eleverna möjligheter att individualisera sin egen inlärning genom att interagera med kunskapsmaterialet via sina egna personligt anpassade användarprofiler.

Den huvudsakliga poängen med en väl designad kunskapskomponent är alltså att den ska kunna komma till användning i en mängd olika inlärningssammanhang. Detta synsätt lämpar sig väl för en nätverksmiljö, där lärare och elever är fria att välja innehåll från en mångfald av tillgängliga resurskomponenter.

## 5.4 Relationer mellan begrepp

Kunskap är alltid uppbyggd och strukturerad kring ett antal begrepp som har formats och relaterats till varandra genom att dela in och strukturera våra sinnesförnimmelser i enlighet med någon form av epistemologisk modell. Den begreppsrelationsmodell som utgör den epistemologiska grunden för Kunskapens Trädgård - och som är illustrerad i Figur (5) - identifierar tre olika typer av hierarkibildande relationer mellan begrepp - nämligen *abstraktion/konkretion, sammanhang/del* och *typ/instans*. Dessa relationer genererar var och en för sig en 'begreppshierarki', dvs en riktad, icke-cyklisk graf (= DAG). Dessutom finns det en 'begreppsanarki' av allmänna associationer, som kopplar olika begrepp till varandra.

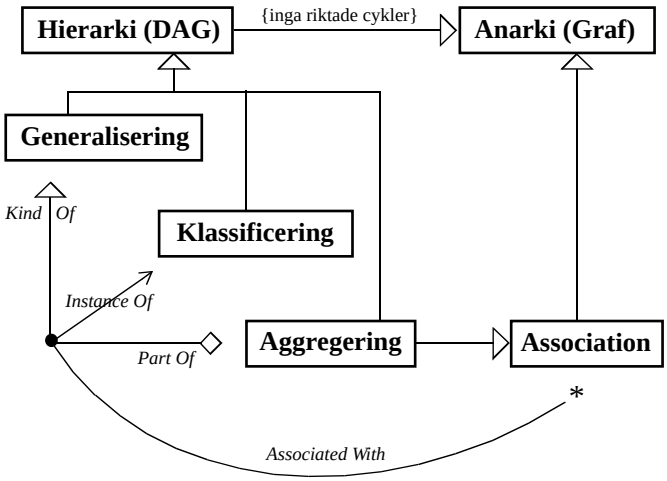


**Fig. 5. De fyra grundläggande typerna av begreppsrelationer**

## 5.5 Lingvistiskt baserad begreppsbildning

I sitt berömda 'Erlangerprogram' från 1872 framlade den store geometrikern Felix Klein idén att betrakta en geometri som en samling påståenden om objekt som är invarianta (= förblir oförändrade) under en grupp av transformationer. Detta markerar början på det moderna betraktelsesätt som betraktar varje geometri som en sorts språk, med sin egen samling av verb (= transformationer) och substantiv (= invarianter).

Vårt språk är intimt relaterat till våra metoder för begreppsbildning. Verben beskriver operationerna (förändringarna) som vi kan observera – eller föreställa oss – medan substantiven beskriver invarianterna, dvs de ”substanser” som överlever verbens operationer (transformationer). Adjektiven beskriver värdet av aspekter (egenskaper) hos substantiven – som t.ex. i satsen: ”den röda billen stannade”, där adjektivet ”röd” representerar värdet av aspekten ”färg” hos just den omtalade instansen av substantivet ”bil”.

I det moderna matematiska synsättet uppför sig alltså ett geometriskt system som ett språk. Det har sina verb, som uttrycker de tillåtna transformationerna (= rörelserna), och sina substantiv, som uttrycker dess invarianter, dvs de begrepp som överlever verbens rörelse-transformationer. Så utgör t.ex. begreppet ”kvadrat” ett substantiv i den vanliga (euklidiska) geometrien, eftersom denna geometris transformationer består av de vanliga (stela) rörelserna - samt likformig förstoring och förminskning – och dessa förändringar förvandlar en kvadrat till en ny kvadrat – ”lämnar kvadrat-egenskapen invariant” som en matematiker skulle uttrycka saken. Begreppet kvadrat är ett euklidiskt begrepp, eftersom det överlever att transformeras av de euklidiska verben. I de geometriska skuggornas värld däremot – s.k. projektiv geometri – är inte begreppet kvadrat väldefinierat. I projektiv geometri är alla skuggbilder av en plan figur ekvivalenta, eftersom de överförs i varandra via projektion från en punkt mot ett plan – en tillåten transformation i projektiv geometri. Eftersom en kvadrat kan projiceras till en ”sned” fyrhörning, så överlever inte kvadraten de projektiva rörelserna, och är därför - i Kleins mening – inget projektivt substantiv. Detta illustreras i Figur (6) som beskriver begrepsrelationer mellan några olika typer av geometri.

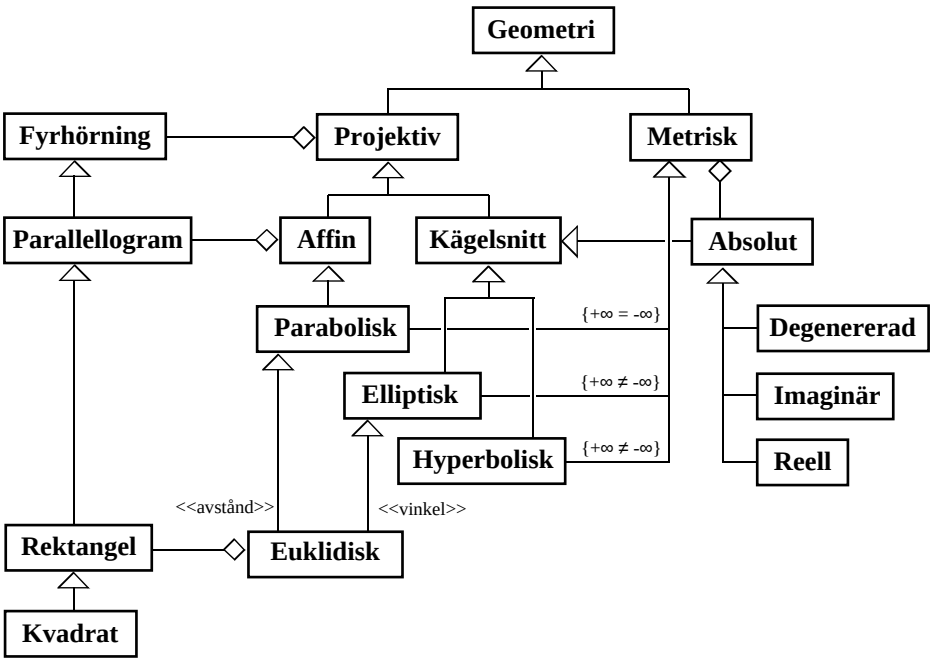


**Fig. 6. Begreppsrelationer mellan några olika typer av geometri**

## 5.6 Begreppet Kunskapstäppa

Kunskapens trädgård kan betraktas som en sort koloniträdgård bestående av ett antal sammanlänkade kunskapstäppor. Varje sådan kunskapstäppa representerar en subjektiv framställning av ett visst kunskapsmaterial, och underhålls av en kunskapsträdgårdsmästare som fungerar som ansvarig utgivare av informationen som presenteras på kunskapstäppan. Den grundläggande begreppsmässiga strukturen hos en kunskapstäppa framgår av Figur (7).

Täppan består av ett antal fakta, som kan vara av typ *begrepp* eller *verktyg*. Verktygens uppgift är att underlätta samspelet mellan begrepp (= concepts) och upplevelser (= percepts). Notera att begreppsrelationerna i Figur (5) uppträder i Figur (7) som associationer mellan begrepp.

Begreppen är i sin tur uppdelade i *mentala* respektive *mediala* begrepp. De förra är strikt personliga (= inre) representationer av olika fenomen, medan de senare är kommunicerbara representationer av fenomenen. Med detta synsätt består själva kommunikationsprocessen av att översätt mentala begrepp till mediala - samt att utbyta de senare med sin kognitiva omgivning.

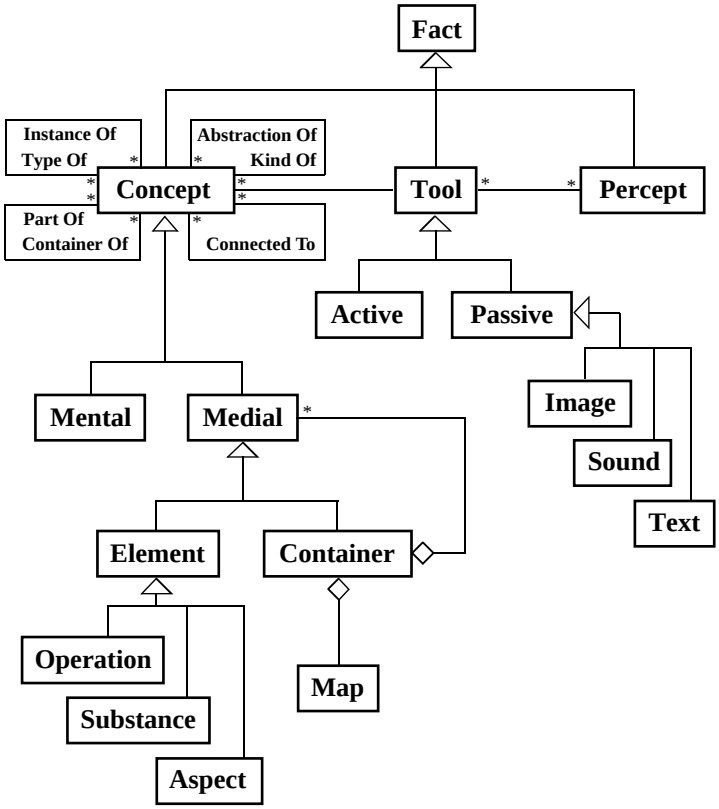


**Fig. 7. Den begreppsmässiga strukturen hos en kunskapstäppa**

De mediala begreppen är i sin tur uppdelade i *elementära* (= element) respektive *behållare* (= container). Ett container-begrepp består i sin tur av ett antal begrepp, som kan vara både element eller containers, medan ett element-begrepp kan vara av typ *operations*-begrepp, *substans*-begrepp eller *aspekt*-begrepp. Dessa tre subtyper av element-begrepp motsvaras i språket av de respektive ordklasserna *verb*, *substantiv* och *adjektiv* - i enlighet med den lingvistiskt baserade begreppsbildingsmodell som har beskrivits ovan.

Ett operations-begrepp motsvaras alltså av ett verb och beskriver någon form av förändring (transformation). Ett substans-begrepp motsvaras av ett substantiv och beskriver något som förblir oförändrat (invariant) under verbens transformationer. Instansvärdet av ett aspekt-begrepp är ett adjektiv, och det beskriver någon form av invariant egenskap hos ett substans-begrepp.

## 5.7 Dynamisk omstrukturering av kunskap

Varje begreppsklassifikationsmodell är till sin natur dynamisk och därför utsatt för olika typer av evolutionära förändringar - både i form av förfining och paradigmskiften. Vetenskapshistorien tillhandahåller många belysande exempel på sådana former av kunskapsutveckling. Inom fysiken användes t.ex. begreppet ‘atom’ för hundra år sedan för att beteckna en storhet som var odelbar, dvs inte var möjlig att dela upp ytterligare. I vår begreppsmodell kan vi alltså säga att begreppet atom för hundra år sedan var ett typiskt exempel på ett element-begrepp.

I takt med att atomfysiken utvecklades, förändrades emellertid atom-begreppet till ett container-begrepp - bestående av element-begreppen ‘elektron’ och ‘atomkärna’, och i takt med utvecklingen av kärnfysiken förändrades begreppet atomkärna till ett container-begrepp som till en början bestod av element-begreppen ‘proton’ och ‘neutron’. Idag har även dessa begrepp transformerats till container-begrepp, bestående av en mängd olika s.k. ‘elementarpartiklar’.

## 5.8 Kunskapens Koloniträdgård - en mångfald av länkade täppor

Som beskrivits ovan kan Kunskapens Trädgård uppfattas som ett antal kunskapstäppor (= subjektivt sammanställda informations-collage) med var sin trädgårdsmästare (= ansvarig utgivare). Varje kunskapstäppa består av ett antal olika kunskapskomponenter, och de olika kunskapstäpporna kan sedan länkas samman till en sorts *Kunskapens Koloniträdgård*.

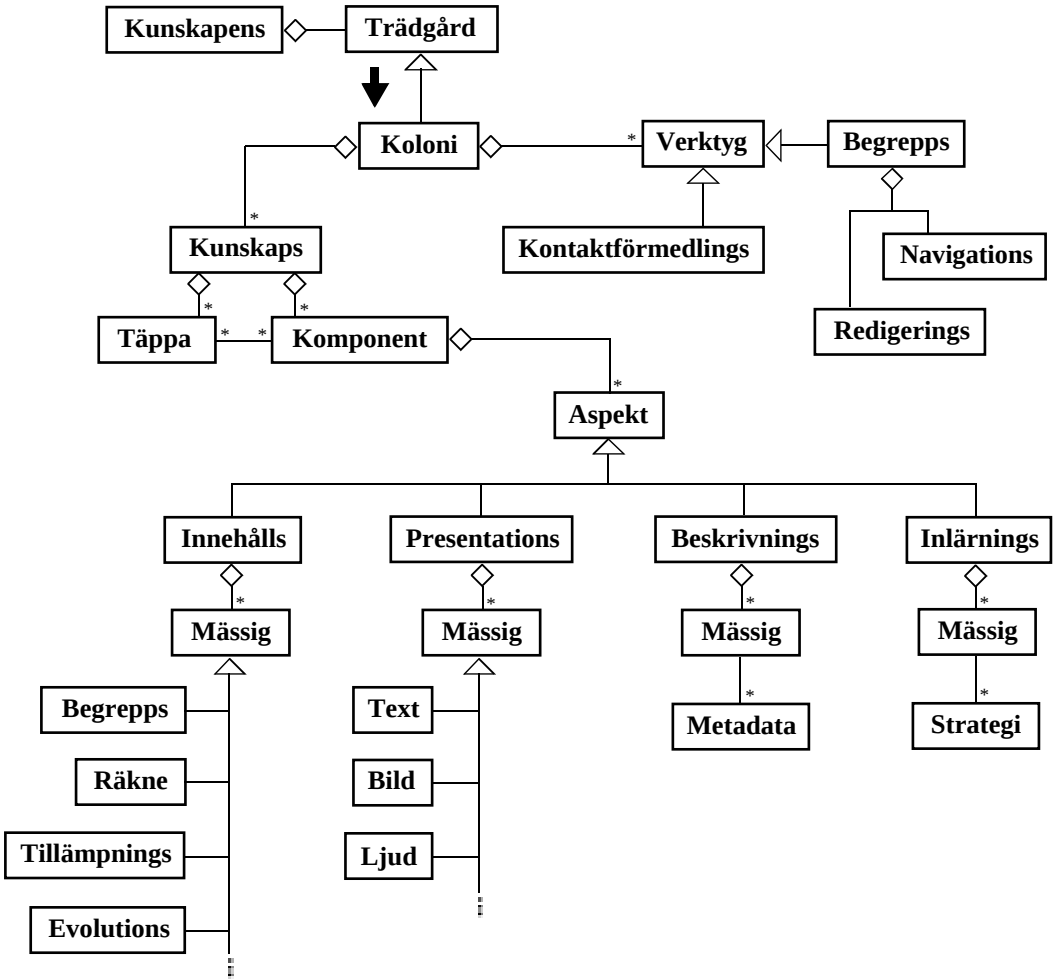


**Fig. 8. Den grundläggande begreppsstrukturen hos kunskapens koloniträdgård**

En sådan koloniträdgård ger möjligheter att separera och modularisera både utlärande (teaching) och inlärande (learning) på olika sätt. Två av hörnstenarna i detta sammanhang är dels den ovan beskrivna modulariseringen av det begreppsmässiga innehållet i kunskapskomponenter, dels en beskrivning och etikettering av detta innehåll med hjälp av s.k. *metadata*. Design av mekanismer för såväl komponentsamverkan som innehållsbeskrivning sker i enlighet med de framväxande internationella standarder för nätverksbaserade läromiljöer som framför allt utvecklas och samordnas av IMS (Instructional Management Systems) projektet. För vidare information om IMS, se appendix - samt [http://www.imsproject.org]. I likhet med IMS-projektet bygger vårt arbete med kunskapens koloniträdgård på det objektorienterade synsätt som visat sig så framgångsrikt vid utvecklingen av distribuerade system i allmänhet.

## 5.9 Att strukturera kunskapskomponenter

För den framväxande nätverksbaserade kunskapsindustrin är ett distribuerat och komponentifierat synsätt på kunskap av avgörande betydelse. Med tanke på de stora kostnader som är förknippade med all form av mjukvaruutveckling blir det viktigt att utforma kunskapskomponenter på ett sådant sätt att de kan användas i en mängd olika inlärningssam-

manhang. Att åstadkomma “många ingångar” samt förmåga till samverkan mellan de olika komponenterna är strategiskt viktiga designmål för att möjliggöra de elevdrivna och individ-orienterade läromiljöer som alla eftersträvar.

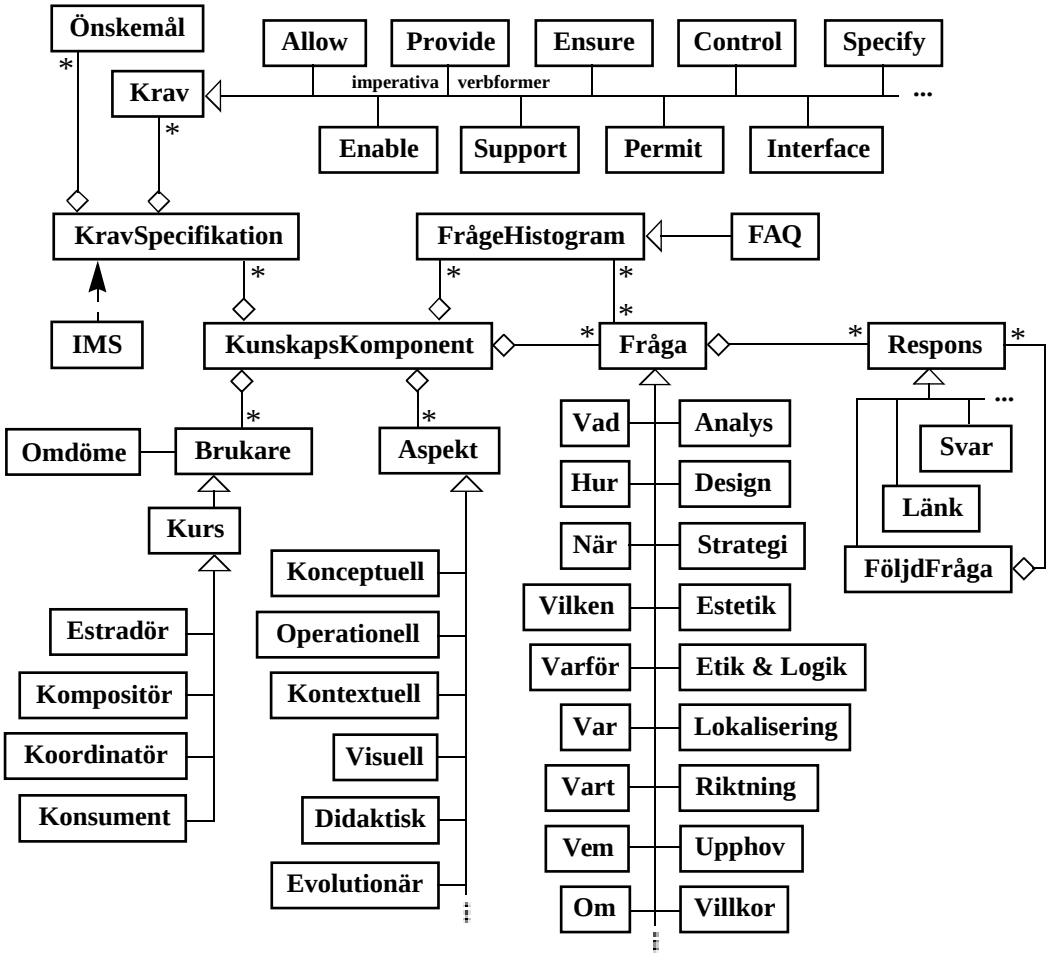


**Fig. 9. Översikt över begrepp som är kopplade till kunskapskomponenter**

Den huvudsakliga poängen med en väl designad kunskapskomponent är alltså att den ska kunna komma till användning i en mängd olika inlärningssammanhang. Detta synsätt lämpar sig väl för en nätverksmiljö, där lärare och elever är fria att välja innehåll från en mångfald av tillgängliga resurskomponenter, som är beskrivna med hjälp av relevanta metadata.

Kunskapskomponenter kan t.ex. ta formen av filer, vilka kan laddas ner i en kunskapstäppa från olika digitala hyperarkiv, uppbyggda med hjälp av flyttbara informationsformat (som t.ex. HTML, SGML, XML, VRML, MIDI, Quicktime eller Java). Vi vill utforska möjligheterna att utnyttja sådana portabla format i syfte att separera ”VAD som lärs ut” (= content) från ”NAR det lär in” (= context). En sådan separation mellan design av ”innehåll” och ”miljö” kan ge insikter i hur man skapar elektroniska läromedel med högre potential för ett individualiserat inlärande.

En väl designad kunskapskomponent kan liknas vid ett skidområde, med många olika vägar att ta sig ned från toppen. Precis som en bra nedfart är markerad med en speciell färg - som talar

om vilken svårighetsgrad som nedfarten erbjuder - så bör en väl designad kunskapskomponent ha olika ‘genomgångsvägar’ med någon form av märkning (= metadata) som svarar mot dokumenterade skillnader i förkunskaper.

Att separera mellan kunskapskomponenter och läroupplevelser kompliceras av det faktum att varje givet svar måste utgå från någon ‘förutfattad mening’ om det sammanhang (= kontext) i vilket motsvarande fråga formulerades. Detta innebär en ofrånkomlig typ av beroende mellan vad som presenteras och när (= under vilka förhållanden) det presenteras. En väl designad kunskapskomponent minimerar denna typ av beroenden genom att undvika onödiga antaganden om det inlärningssammanhang i vilket den kommer att användas.

## 5.10 Övergripande designprinciper för kunskapskomponenter[1]

I dag fungerar de flesta IT-stödda läromiljöer som isolerade öar. Såväl systemuppbyggnad som innehåll är ofta knutet till en specifik plattform och en specifik typ av tillämpning. Såväl producenter (förlag, läromedelsförfattare m.fl.) som lärare och elever har behov av mer öppna system som tillfredsställer följande krav på innehåll, metadata och struktur.

### 5.10.1 Innehållet skall:

• vara återanvändbart (värdesäkrat), dvs. kunna utnyttjas i nya och ofta oförutsedda tillämpningar och sammanhang

• vara flyttbart, dvs kunna överföras mellan skilda plattformar och tillämpningar.

• kunna presenteras på skilda media (skärm, papper, webplatser, CD-ROM etc).

• vara modulärt uppbyggt, dvs. nedbrytbart till komponenter som kan sammanfogas på skilda sätt för varierande behov.

• så långt som möjligt vara tillgängligt i format som följer framväxande internationella märkningsstandarder eller de facto-standarder, såsom SGML, XML, HTML, VRML, MIDI etc.

### 5.10.2 Metadata, dvs. information om innehållet, skall:

• i likhet med innehållet vara återanvändbara, flyttbara, modulärt uppbyggda samt anpassade till pågående internationellt standardiseringsarbete.

• på ändamålsenligt sätt stödja sökning, indexering samt återvinning av innehållet.

• kunna utformas för användning i lärandesammanhang, bl.a. genom att underlätta för lärare och elever att finna resurser (digitala arkiv, innehållskomponenter, länkkataloger, personer att diskutera med etc) samt för producenter att varudeklarera sina produkter och fokusera sina målgrupper.

1. Detta avsnitt är författat i samarbete med Donald Broady

• tillåta flexibel exponering (filtrering) av innehållet i enlighet med skilda behov, så att t.ex. företagsintern information kan döljas för omvärlden eller skilda utsnitt ur ett och samma innehåll presenteras för elevgrupper med olika förkunskapsnivåer.

• ge information om såväl själva innehållet och dess tänkta användningsområden (t.ex. "avsett för årskurs 6", "instuderingsfrågor om miljövård") som innehållets format (tiff-bild, XML, HTML version 4.0 etc).

• göra tjänst som kopplingdon mellan innehållskomponenter, vilka därmed kan byggas samman i nya kombinationer utan onödigt manuellt arbete med konvertering o likn.

#### 5.10.3 Strukturen, dvs. systemet av relationer mellan komponenter, skall:

• vara dynamisk, dvs kunna modifieras och utvidgas för att passa ett brett spektrum av behov och situationer.

• tillåta att komponenter och metadata är flyttbara, dvs kan överföras från exempelvis en webplats till en annan

• tillåta att innehåll hämtat från t.ex. skilda noder på Internet integreras i lärares eller elevers egna system.

• anpassas till det internationella standardiseringsarbetet, exempelvis tillämpliga riktlinjer utarbetade inom projektet IMS (Instruction Management system, se http://www.imsproject.org) som syftar till en de facto-standard för system för distribuerad undervisning.

### 5.11 Helpdesk - människa till människa via maskin

Det finns ingen form av automatiserat inlärningssystem som kan mäta sig med fysisk kontakt med engagerade och kunniga lärare - ifall man som inlärare är intresserad av deras ämne. Problemet är att det inte finns tillräckligt med sådana lärare för att kunna möta dagens och morgondagens enorma utbildningsbehov. Ytterligare ett problem är att dagens kunskapskällor är dåligt lokalitetsmatchade till inlärarnas intresse - dvs kunskapen och intresset har för få naturliga mötesplatser i det fysikaliska rummet.

En fundamental del av arbetet med kunskapens koloniträdgård är därför utvecklingen av ett helpdesk-system som gör det möjligt för en inlärare att få direkt kontakt med resurspersoner som har kompetens att diskutera de djupare frågor som dyker upp under inläroprocessens gång. Sådana kunskapskällor ska kunna erbjuda sina tjänster i ett katalogsystem, samt lokaliseras och kontaktas on-line. Det handlar här om att kunna upprätta en levande diskussion kring frågor som går utanför materialets ramar på olika sätt. Sådan frågor representerar oundvikliga problematiseringar - naturliga tecken på växande insikt och förståelse - vilket gör dem extra viktiga att fånga upp, om det gryende intresset hos inläraren ska kunna vidmakthållas.

### 5.12 FrågeBaserad Inlärning (FBI)

Problembaserad inlärning är idag ett väl etablerat begrepp inom pedagogiken. Problemet med PBI är att det oftast är lärarna som formulerar problemen. Om man i stället utgår från inlärarnas frågor så får man ett system som skulle kunna kallas FrågeBaserad Inlärning (FBI). Inom FBI

betraktar man elevernas frågor som en sorts officiella brev till myndigheterna - och registrerar dem och övervakar deras hantering på motsvarande sätt.

Frågebaserad inlärning hänger intimt samman med det pågående paradigmskiftet bort från pliktbaserad utbildning i riktning mot rättighetsbaserad utbildning som för närvarande pågår i många utbildningssammanhang. Kunskapssamhället och det livslånga lärandet är två välkända uttryck för denna förändrade inställning till utbildning som diskuteras dagligen i den allmänna debatten. Den begreppsmässiga strukturen i de båda utbildningsparadigmen framgår av Figur (10) respektive Figur (11). En mera ingående diskussion av dessa frågor återfinns i [(30), kap. 6.2].

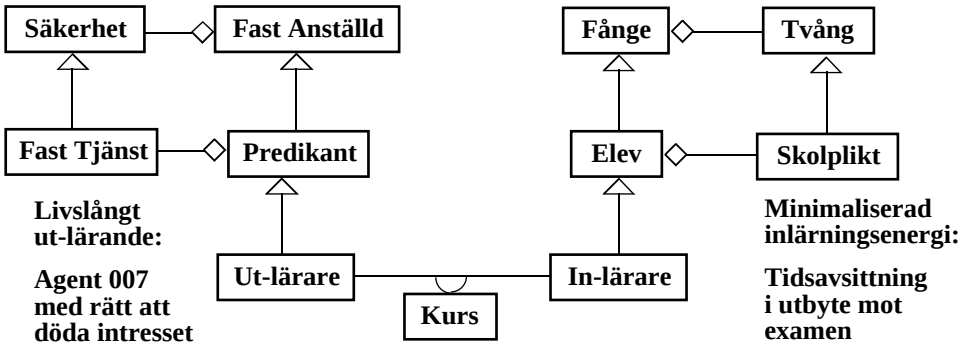


**Fig. 10. Pliktbaserad utbildning baserad på livslångt ut-lärande**

Till den pliktbaserade utbildningsstrukturen hör de traditionella, kollektivt orienterade, klassbaserade läroplanerna. Den rättighetsbaserade utbildningen däremot är kopplad till individuellt orienterade läroplaner och inlärningsstrategier som baserar sig på individens unika förmågor och intressen. Detta är en förutsättning för att uppnå det klasslösa utbildningssystemet, eller - som jag föredrar att kalla det - ett förstklassigt utbildningssystem för alla! När allt kommer omkring så är ju skolan den enda plats där vi betraktar en förflyttning från första klass till andra klass som en form av förbättring. Man kan ju föreställa sig hur man skulle reagera om man blev behandlad på motsvarande sätt av t.ex. ett flygbolag!

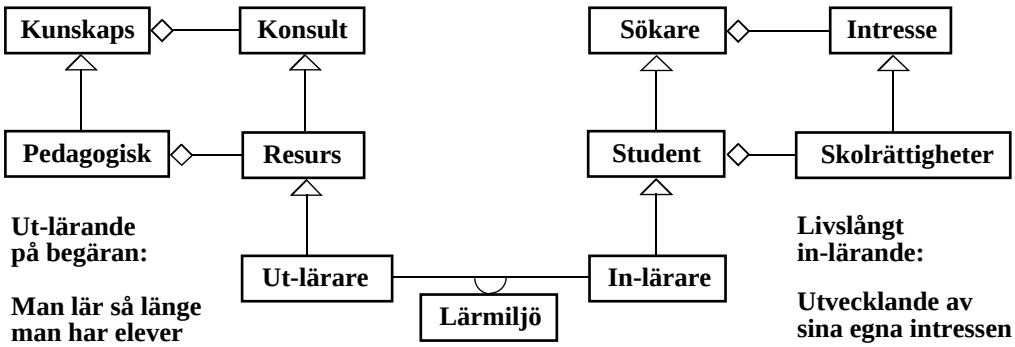


**Fig. 11. Rättighetsbaserad utbildning baserad på livslångt in-lärande**

En stor fördel med FBI är att den bidrar till att ge inlärarna kontroll över sina egna inlärningsstrategier. Genom att synliggöra de frågor som varje enskild inlärare ställer - och behandla dem med den respekt de förtjänar - skapas ett individualiserat förhållningssätt till kunskap som motverkar den alltför vanliga IGI-pedagogiken (Inlärning Genom Imitation) - speciellt inom området matematik.

Detta innebär ingalunda att inlärarna 'lämnas i sticket' (= åt sig själva) under själva frågeformuleringsprocessen- Tvärtom stimuleras de hela tiden av utlärarnas beskrivningar och presentationer - vilka fungerar som en sorts kunskapens smörgårdsbord. Som inlärare har man möjligheter att lyssna på och ta del av en mängd olika saker, men det är hela tiden det egna intresset som styr, och de egna frågeställningarna som filtreras fram.

Frågebaserad (= intressebaserad) inlärning ställer delvis nya krav på lärarrollen i förhållande till den traditionella pliktbaserade inlärningen. Som elev i ett intressebaserat inlärningssystem ska man inte behöva fyllas av känslan att "jag måste lära mig det som läraren redan kan" (läraren som kunskapsfilter). I stället ska man uppmuntras att känna att läraren är en viktig resurs som kan hjälpa mej att ta reda på det som jag vill veta.

Det är detta som är den övergripande avsikten med de tre olika dimensionerna hos lärarrollen som beskrives i Figur (12): läraren som *predikant, trädgårdsmästare* respektive *rörmokare.* Läraren-predikanten predikar som förut - dock utan den nuvarande 'prediko-garantin' (= tenure), utan snarare en form av 'predikan-på-begäran' ("Man lär så länge man har elever"). Läraren-trädgårdsmästaren hjälper varje elev att designa sin egen personliga läroplan och inlärningsstrategi samt fångar upp elevernas frågor. När trädgårdsmästaren inte kan svara själv kopplar han eller hon in sitt eget nätverk av predikanter. Om frågan ändå inte leder till en tillfredsställande 'svarskontakt', så kopplar trädgårdsmästaren in läraren-rörmokaren som fortsätter att arbeta för att etablera en samtalsyta för vidare diskussion av frågan - antingen i den elektroniska (virtuella) eller den fysiska (reella) miljön.

Trädgårdsmästaren arbetar alltså för att locka fram elevernas frågor, rörmokaren ser till att så få frågor som möjligt går förlorade, och predikanten ser till att så många frågor som möjligt blir besvarade på ett sätt som stimulerar till ett vidare studium av ämnet.

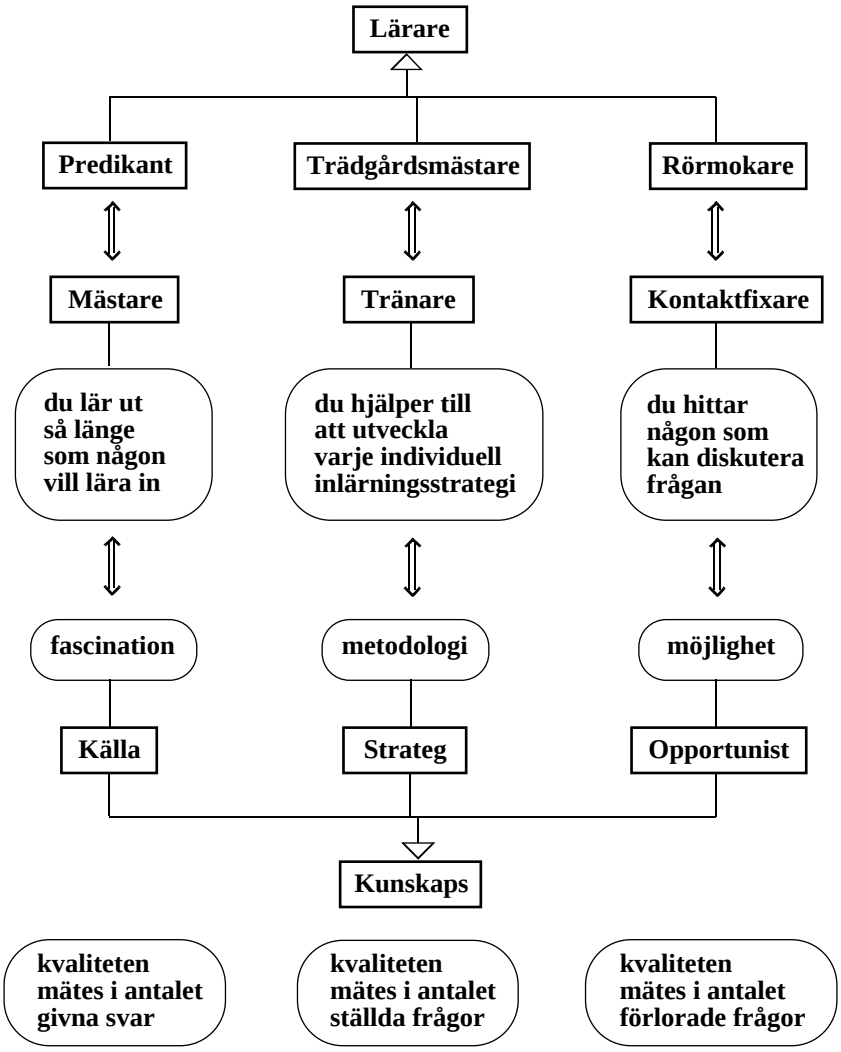


**Fig. 12. Tre olika dimensioner av lararrollen i ett FBI system**

Flera av dessa roller kan samexistera inom samma fysiska person, men med tanke på dagens explosionsartat ökande krav på utbildning - samt de låga löner som fortfarande betalas för (offentligt anställda) utlärare - så kommer det helt enkelt inte att finnas tillräckligt med sådana 'super-lärare' i elevernas fysiska miljö. Genom de virtuella lärmiljöernas framväxt öppnar sig möjligheten att distribuera dessa lärar-roller över det fysiska rummet - och utnyttja cyber-rymden som en elektronisk switch-board för att förbinda inlärarnas nyfikenhet och intresse med utlärarnas kunskap och kommunikationsförmåga. På detta sätt skulle det kunna skapas reella möjligheter för utlärarna att växa ur sina nuvarande roller som kunskapsfilter för sina respektive inlärare.

## 5.13 Att designa sin egen lärostrategi

Att designa sin egen lärostrategi innebär bl.a. att dokumentera sitt förhållningssätt till följande tre frågor: "Vad är jag intresserad av?", "Vad finns det att lära sig om detta?", samt "Vad är jag villig att investera min tid på att lära mig?". Dessa frågor är naturligtvis oerhört komplexa, och

kräver kontinuerlig bearbetning och uppdatering i en iterativ process för att leda till meningsfulla resultat. Att vårda dessa frågor i en pågående dialog med varje enskild elev är en viktig uppgift för läraren-trädgårdsmästaren.

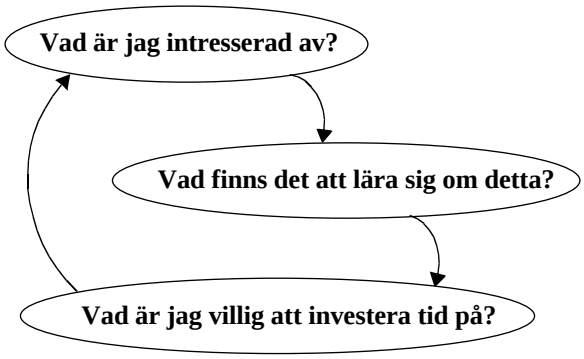


**Fig. 13.Tre grundläggande frågor vid iterativ design av en inlärningsstrategi.**

### 5.14 Kunskapsvaruhuset - ett arkiv av kunskapskomponenter

En av framtidsvisionerna för projektet kunskapens trädgård är en sorts IKEA-liknande 'kunskapsvaruhus' på nätet. Där finns kunskapskomponenter med metadata-baserade katalogbeskrivningar, samt olika stödverktyg för t.ex. 'begrepps-browsing' och täppkonstruktion. Här ska utlärarna kunna plocka ihop sina egna täppor av kunskapskomponenter och skräddarsy sina egna läromiljöer (= kombinationer av inlärningsmöjligheter). Som inlärare ska man sedan kunna 'browsa' (= botanisera) bland täpporna, och med hjälp av olika diagnoser (tester + rådgivning) formulera sin egen inläroplan. Sedan är man redo att följa sina egna inlärovägar genom materialet - för att så småningom dokumentera de vägar man vandrat - och de lärouppleverser som detta har givit upphov till - i sin egen takt.

## 6 Begreppsorganisation och innehållspresentation

### 6.1 Bakgrund

För att kunna överblicka relationerna mellan olika kunskapskomponenter och presentera deras innehåll på ett överskådligt sätt behöver man tillgång till någon form av överblicksverktyg. Jag har föreslagit namnet *begrepps-browser* för ett sådant verktyg - i analogi med s.k. web-browsers - som Netscape Navigator™ och Microsoft Explorer™ - vilka används för att presentera information på Internet och som blivit välkända under senare år.

Den grundläggande epistemologiska strukturen som ligger till grund för konstruktion av denna typ av överblicksverktyg diskuteras ingående i en kommande CID-rapport[1]. Av denna framgår bl.a. att för att kunna uppfylla ovan nämnda krav och fungera på ett tillfredsställande sätt bör en begrepps-browser konstrueras i enlighet med ett antal grundläggande designprinciper som kan sammanfattas enligt följande.

1. Naeve, A., *Conceptual Navigation and Multiple Scale Narration in a Knowledge Manifold* [(33)]

## 6.2 Grundläggande designprinciper för begrepps-browsers

(i) Separera *kontext* (= sammanhang) från *innehåll*.

(ii) Beskriv varje kontext med en *begreppskarta* uttryckt i UML (Unified Modeling Language).

(iii) Tilldela ett lämpligt antal kunskapskomponenter som *innehåll* till varje begrepp respektive relation på begreppskartan.

(iv) *Visa* innehållet i ett visst begrepp (eller relation) *filterat* genom olika *aspekter.*

(v) *Etikettera* komponenterna med ett standardiserat databeskrivningssystem (*metadata*).

(vi) Transformera en innehållskomponent som själv är en begreppskarta till en kontext genom att *kontextualisera* den.

## 6.3 Fördelar med en begreppskarta

En begreppskarta har flera fördelar i jämförelse med en verbal presentation av motsvarande begreppsrelationer. Två av dessa förtjänar att särskilt framhållas i detta sammanhang:

För det första: En begreppskarta bryter upp ordningen i den verbala framställningen av begreppsrelationerna. Den visar alla relationer på samma gång, till skillnad från en verbal framställning, som är tvungen att välja att beskriva dem i en viss ordning. En verbal framställning av en relation mellan två begrepp kan betraktas som en resa på kartan - en navigerad väg från det ena begreppet till det andra. En bakomliggande orsak till fördelen med en begreppskarta i förhållande till en verbal presentation är det faktum att vår förmåga att visuellt överblicka en begreppsrelation från olika håll är betydligt större än vår förmåga att verbalt byta riktning på relationen ifråga. Vi kan alltså lättare “visuellt integrera” informationen och med kartans hjälp skapa oss en “helhetsbild” av relationerna mellan begreppen. Det är ju därför som vi kallar det “helhetsbild” och inte t.ex. “helhetsord”.

För det andra: En begreppskarta skapar ett övergripande logiskt sammanhang mellan begreppen. Både begreppen (= noderna) och länkarna (= bågarna) på kartan kan sedan förses med olika typer av innehåll, varefter kartan kan navigeras och innehållet presenteras ur ett antal olika (konfigurerbara) aspekter (= filter). Genom att begreppsinnehållet presenteras separat från begreppsrelationerna blir det möjligt att ta del av innehållet utan att förlora överblicken över relationerna, dvs sammanhanget.

## 6.4 Conzilla - en första prototyp till begrepps-browser

Under det senaste året har jag handlett ett examensarbete som syftar till att ta fram en prototyp till begrepps-browser som är implementerad i enlighet med ovanstående designprinciper. Arbetet har utförts av två studenter vid institutionen för Teknisk Datavetenskap vid Uppsala Universitet, Michael Nilsson & Mattias Palmér, och har resulterat i en första prototyp - kallad Conzilla - som uppfyller alla ovan listade designprinciper så är som på (iv), vilken kommer att uppfyllas i en senare version.

Examensarbetet har koncentrerats på att utveckla stöd för navigering i begreppsrymden med hjälp av begreppsrelationer uttryckta i UML (Unified Modeling Language) samt implementera maskinläsbara beskrivningar och filtreringar av sådana begreppsrelationer med hjälp av XML och Java. Arbetet består av två samverkande delar som går ut på att utveckla en *begrepps-editor* och en *begrepps-browser* enligt följande:

#### 6.4.1 Begrepps-editorn

ska möjliggöra för en konstruktör av en kunskapskomponent att maskinellt beskriva hur de ingående begreppen är tänkta att relatera till andra begrepp. Detta är en del av den meta-modell som hör till vår komponentifierade kunskapsarkitektur, och som är nödvändig för att möjliggöra en effektiv sammankoppling av de olika komponenterna vid konstruktion av en viss lärromiljö (= kunskapstäppa). En central del i uppgiften består i att beskriva de grundläggande konstruktionerna i UML med hjälp av tag-språket XML, vilket ger kopplingen till den Java-miljö i vilken Kunskapens Koloniträdgård är tänkt att fungera (JavaWebserver, servlets).

#### 6.4.2 Begrepps-browsern

ska stödja diagram-baserad begrepps-browsing i enlighet med ovanstående designprinciper. Genom att utnyttja begrepps-editorns förbearbetning av materialet ska man med hjälp av begrepps-browsern maskinellt kunna söka i och traversera olika typer av begreppsmodeller (= begreppskartor) som är uttryckta i UML, samt presentera olika varianter av aspektfiltrerade vyer av materialet. Till exempel ska man kunna röra sig mellan olika kartor, samt kunna få en överblick av vilka vägar som finns inlagda mellan två godtyckliga begrepp A och B (dvs få en uppfattning om på vilka sätt som A och B är relaterade i modellen). Man ska även kunna variera presentationen beroende på olika typer av bivillkor, som t.ex. förkunskaper och mediala begränsningar (typ handikapp). Sålunda ska ett begrepp kunna undertryckas när dess närvaro inte är meningsfull.

Begrepps-browsning blir på ett naturligt sätt uppdelad i två olika moder - kallade *surfning* respektive *visning* - vilka motsvarar uppdelningen av begreppen i *relationer* respektive *innehåll*. Man surfar (= rör sig mellan) begreppskartor och betraktar (= visar) de olika kartsymbolernas innehåll. I en tredje s.k. *metadata-mode* kan man dessutom studera ‘varudeklarationen’ (= etiketteringen) av de olika komponenterna - t.ex. vem som är författare, vilka målgrupper som avses, vilka restriktioner som gäller för användning, etc. Denna metadatamärkning sker i enlighet med den framväxande IMS-standarden (se appendix).

Som nämnts ovan kommer Conzilla i framtiden även att kunna filtrera innehållet med avseende på ett antal konfigurerbara aspekter. Den första versionen är av naturliga skäl både ofullständig och behäftad med ett antal brister, framförallt vad gäller användarstöd för konstruktion och editering av de ingående komponenterna. Koden bakom Conzilla är dock logiskt välstrukturerad och skriven i ren Java (= 15000 rader), vilket gör den till utmärkt grund att bygga vidare på. Det borde inte vara alltför svårt att kunna intressera begåvade studenter för att anta de programmeringsmässiga utmaningar som kan utveckla Conzillas kapacitet och användargränssnitt vidare.

Vi planerar att göra Conzilla tillgängligt på Internet så snart som det grafiska användarstödet för konstruktion och editering av komponenter har förbättrats till den punkt där dessa aktiviteter blir mindre omständiga att utföra än vad som är fallet idag. För närvarande finns ett visst mått

av grafiskt stöd för editering av kartor, medan begrepp och metadata måste editeras som text i motsvarande bakomliggande XML-filer - något som upplevs som ganska plågsamt och tidsödande.

# 7 Begrepps-browsning av komponentarkiv i multipel skala

## 7.1 Introduktion

För att illustrera ovanstående idéer och principer för hur vår begrepps-browser kommer att fungera presenterar jag här ett exempel på begrepps-browsning av ett arkiv för matematiska kunskapskomponenter. Jag avser att konstruera ett sådant arkiv med hjälp av Conzilla, och jag kommer att presentera exemplet med hjälp av den tänkta framtida version av programmet som uppfyller de sex designkrav som angetts ovan.

## 7.2 Surfa kontexten

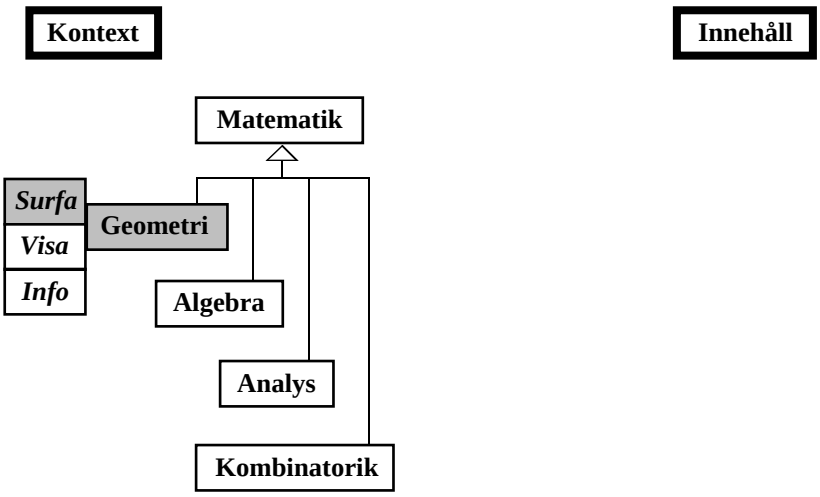


**Fig. 14. Surfa kontexten - matematiken indelad i fyra olika ämnestyper.**

Figur (14) vissar ämnet *matematik* uppdelat i fyra olika ämnesområden: *geometri*, *algebra*, *analys* och *kombinatorik*. Motsvarande begreppskarta har laddats in i kontext-delen, som visas i den vänstra delen av display-fönstret. Om man pekar (med musen) på geometri-lådan så markeras densamma, vilket visas med skuggning i figuren. Detta medför dessutom att det dyker upp en pop-up meny som ger möjlighet att välja *surfa*, *visa* respektive *info*. Om man sedan klickar på *surfa* så ändras kontexten och man hamnar i den begreppskarta som visas i Figur (15).

## 7.3 Visa innehållet under 1-dimensionell aspektfiltrering

Den nya kontexten framställer begreppet geometri uppdelat i undertyperna *algebraisk geometri*, *differentialgeometri* respektive *projektiv geometri*. Om man pekar på ‘projektiv’ så markeras motsvarande begreppslåda och samma pop-up meny som förut dyker upp. Om man nu

väljer *visa* (i stället för *surfa*), så dyker det upp en ny pop-up (under)meny till höger om den första pop-up menyn. Denna under-meny representerar en mängd aspekter genom vilka innehållet i begreppet projektiv geometri kan betraktas (= filtreras).

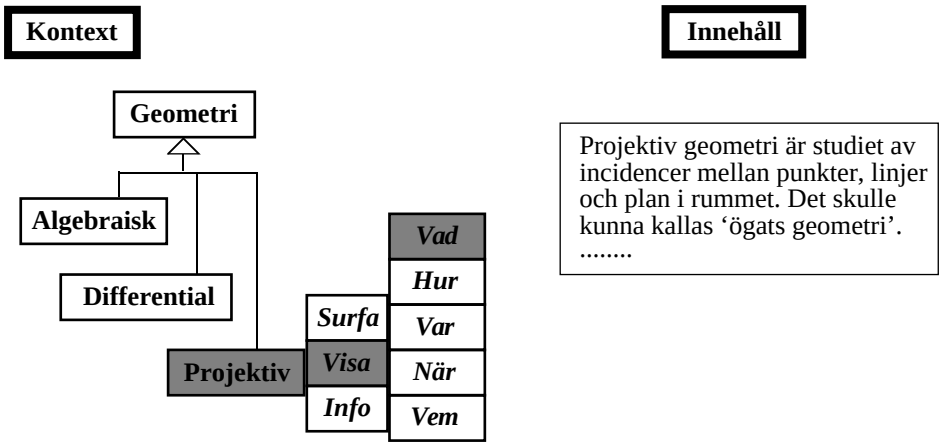


**Fig. 15. Visa innehållet under aspekten *vad* (1-dimensionell aspekt filtrering). Vad är projektiv geometri? (= begreppsmässig aspekt).**

Figur (15) visar den begreppsmässiga (= *vad*) aspekten av projektiv geometri som en ruta i den högra (innehålls)delen av display-fönstret.

## 7.4 Multipelt berättande

Figur (15) visar en beskrivning - i en viss upplösning (= detaljnivå) - av vad projektiv geometri är för något. Genom att dubbelklicka på innehålls-lådan till höger kan man nu ändra upplösning på berättelsen och t.ex. öka upplösningen för att visa upp en mer detaljerad beskrivning. Detta är ett exempel på vad man kan kalla *multipelt berättande* - baserat på en 1-dimensionell upplösningsskala. Två-dimensionellt multipelt berättande baserat på upplösningsdimensionerna *förklaring* och *fördjupning* diskuteras nedan i samband med Figur (20) och Figur (21).

## 7.5 Två-dimensionell aspektfiltrering

Figur (16) illustrerar en annan typ av multipelt berättande som kan kallas 2-dimensionell (eller matris-baserad) aspektfiltrering. *Grund-*, *gymnasie-*, och *högskolenivåaspekten* på *projektiv geometri* kan kombineras med alla andra aspekter på ämnet (*vad*, *hur*, *var*, ...). För att t.ex. ta fram innehållet som motsvarar *gymnasienivåaspekten* av *hur* (= den räknemässiga) aspekten på ämnet projektiv geometri så klickar man helt enkelt i motsvarande position i aspekt-matrisen (som visas skuggad i figuren).

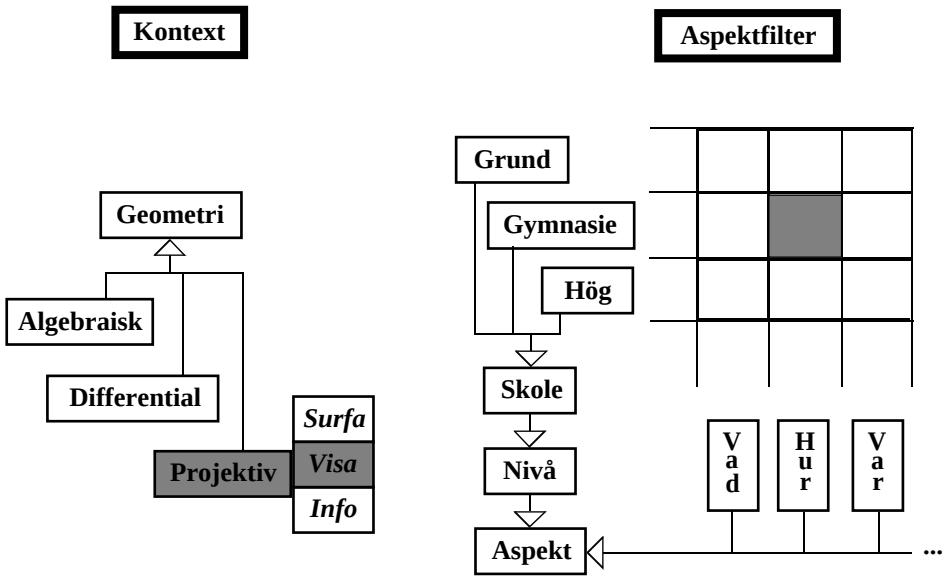


**Fig. 16. Visa innehållet under 2-dimensionell (matris-baserad) aspektfiltrering. Alla andra aspekter kan kombineras med skolnivåaspekterna.**

Det är intressant att notera att många dataspel erbjuder liknande typer av innehållsfiltrering, där nivåaspekter (nybörjare, medelduktig, avancerad, expert) påverkar interaktionen med spelet på olika sätt.

## 7.6 Multipelt berättande baserat på förklaring och fördjupning

I Figur (17) betraktas begreppet matematik som isolerat, dvs i kontext av sig själv. En ny display-struktur har lags till jämfört med tidigare, nämligen scroll-fönstret i den övre högra delen av figuren. Genom att scrolla i detta fönster kan man ändra upplösning på beskrivningen av innehållet. Genom att scrolla mot höger så ökar man *förklaringen*, och genom att scrolla uppåt så ökar man *fördjupningen* av presentationen. Grundtanken är att om man inte förstår en viss beskrivning så ökar man förklaringen, vilket ger nya exempel med samma begrepp inblandade. När man så småningom har förstått förklaringen kan man sedan scrolla vertikalt och öka fördjupningen, vilket blandar in nya begrepp och expanderar de enklare (= mindre djupa) beskrivningarna av innehållet. Ett exempel på detta förfarande visas nedan i Figur (20) och Figur (21).

## 7.7 Ett längre sammanhängande exempel

Figur (17) visar innehållet i begreppet matematik under aspekten *vad*. En ändring av aspekt från *vad* till *hur* leder till beskrivningen i Figur (18), vilken visar hur matematik bedrives (= *algoritmisk* aspekt).

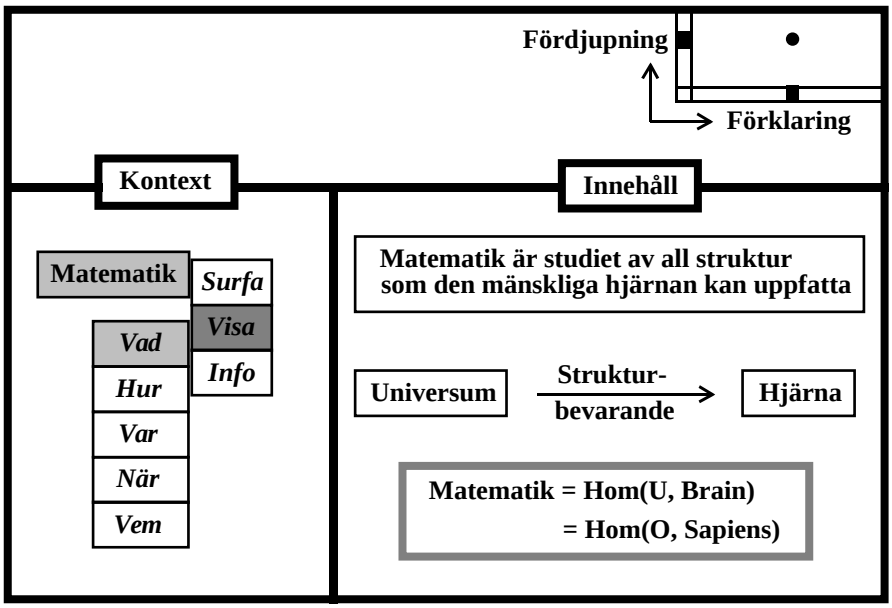


**Fig. 17. Visa innehåll under aspekten *vad*: Vad är matematik?**

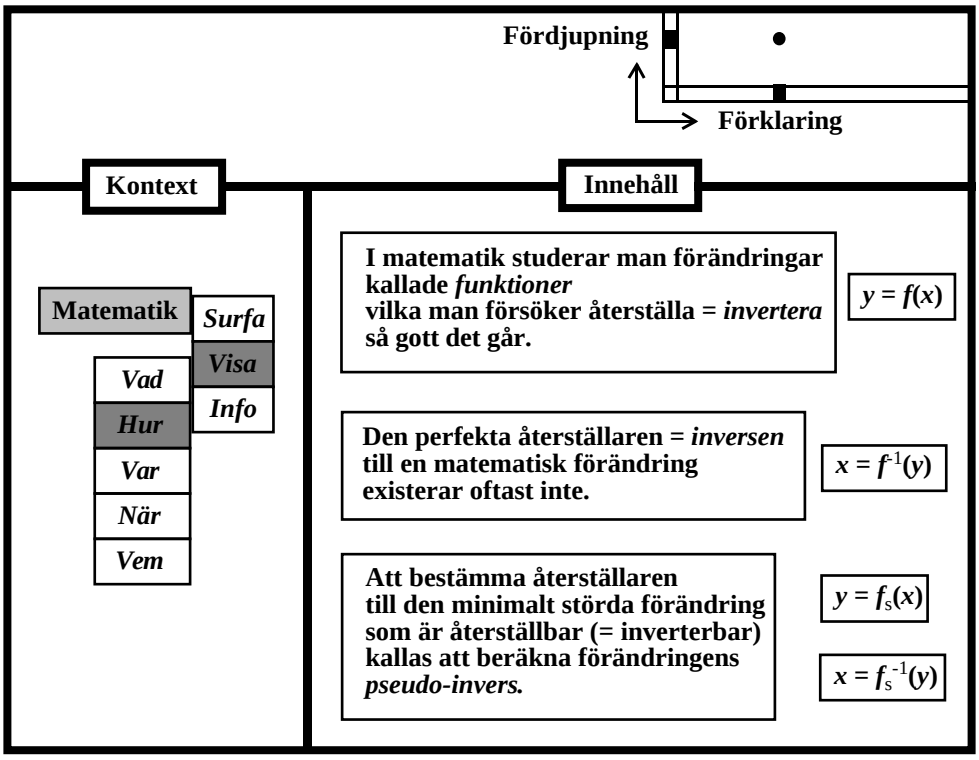


**Fig. 18. Visa innehåll under aspekten *hur*: Hur bedriver man matematik?**

Ännu en ändring av aspekt på matematiken - från *hur* till *var* - leder från Figur (18) till Figur (19). Eftersom innehållet i detta fall visar sig vara en begreppskarta, dyker det upp en ruta med texten *kontextualisera* i den övre delen av fönstret - i enlighet med designprincip (vi) ovan.

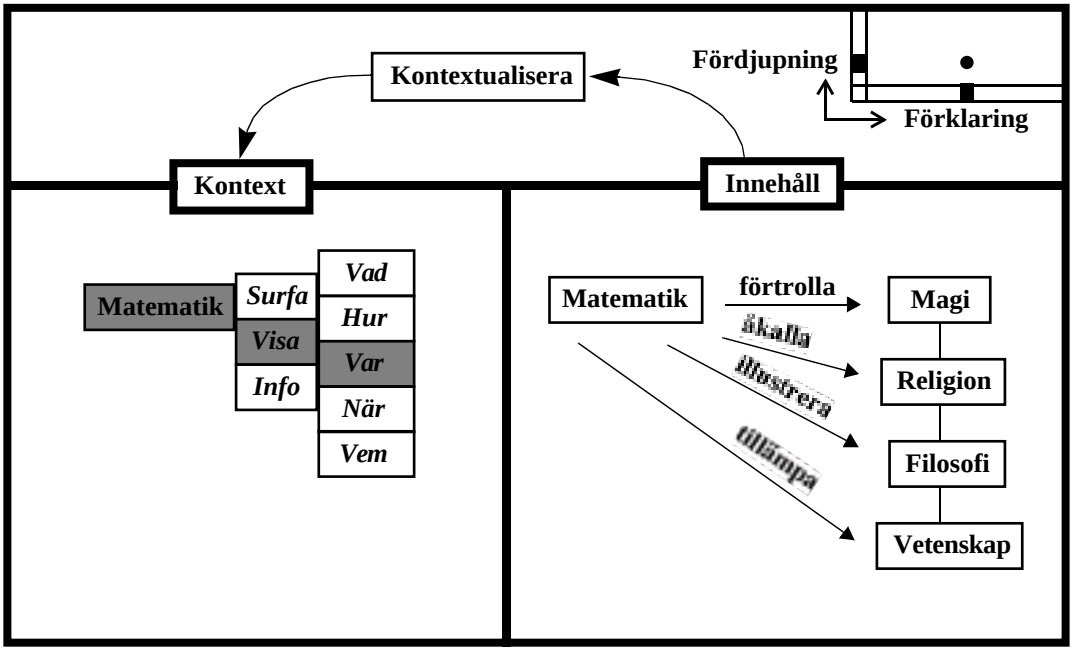


**Fig. 19. Visa innehåll: matematik under aspekten *var*:**
**Var använder man matematik?**

Om man klickar på rutan *kontextualisera* i Figur (19), så flyttas begreppskartan i innehållsfönstret till höger till kontextfönstret till vänster, vilket gör den tillgänglig för kontext-surfning respektive innehålls-visning på samma sätt som tidigare. Följer man upp detta med att peka på bågen (= begreppsrelationen) *tillämpa* - och sedan väljer att visa innehållet i denna relation under aspekten *hur*, så leds man till presentationen i Figur (20), vilken visar *hur matematiken tillämpas inom vetenskapen*.

Figur (20) presenterar en förklaring av denna fråga på en minimal fördjupningsnivå. Matemmatiken beskrivs som en mängd villkorliga (= hypotetiska) påståenden. Vetenskapen använder matematikens logiska (= deduktiva) resonemangsförmåga ($A \Rightarrow B$) för att transformera ett *antagande* ($A$ är sant) till en *logisk slutsats* (då måste $B$ vara sant).

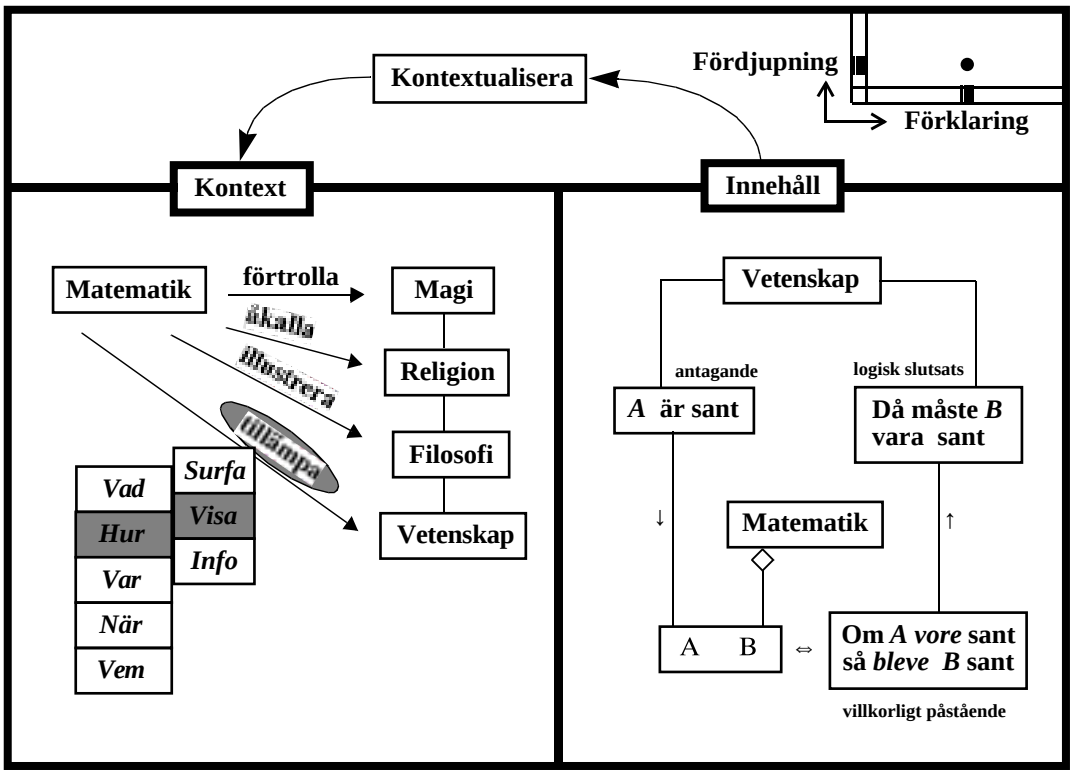


**Fig. 20. Visa innehåll: *tillämpa* under aspekten *hur*:**
**Hur tillämpar man matematik på vetenskap? - minimal fördjupningsnivå**

Observera att på denna (minimala) fördjupningsnivå får man ingen förklaring till *varför* vetenskapen vill omvandla antaganden till logiska slutsatser. Om man nu ökar förklaringsdjupet - genom att scrolla uppåt längs motsvarande fördjupningsdimension - så dyker det upp en fördjupad förklaring som visas i Figur (21). Här kan man utläsa att orsaken till detta är att vetenskapen vill omvandla *teoretiska antaganden* till *experimentella fakta*, vilket den gör genom att utsätta de *logiska slutsatser* som följer ur antagandena för *falsifikationsförsök* genom *experiment*. Poängen är att de logiska slutsatserna (om de ska vara användbara) är mycket lättare att testa experimentellt än de ursprungliga antaganden från vilka de har härletts med hjälp av logiska (= matematiska) metoder. På detta sätt sluter vetenskapen cirkeln och de antaganden, vars logiska slutsatser överlever de experimentella falsifikationsförsöken transformeras (gradvis) till vetenskapliga fakta.

Om man i stället hade scrollat horisontellt - och därigenom ökat förklaringen i stället för fördjupningen - så hade man fått fram en ny beskrivning av innehållet i Figur (20) - kanske i form av ett konkret exempel på transformationen från antagande till logisk slutsats. Poängen är att det hela tiden är inläraren själv som kontrollerar upplösningsfönstret och därför kan skjuta upp kontakten med de djupare nivåerna och röra sig (horisontellt) i förklaringsriktningen ända tills han eller hon har förstått beskrivningen på denna nivå och därmed är redo för fördjupade beskrivningar.

I själva verket kan man tänka på punkten i upplösningfönstret som en prick på en karta som beskriver det begreppsmässiga beroendet mellan de olika innehållskomponenterna. När man scrollar så rör man sig längs olika håll på kartan, men det är inte säkert att man omedelbart kan öka djupet utan att först ha passerat en viss horisontell position (= förklaring) som behövs för att fördjupa motsvarande beskrivning.

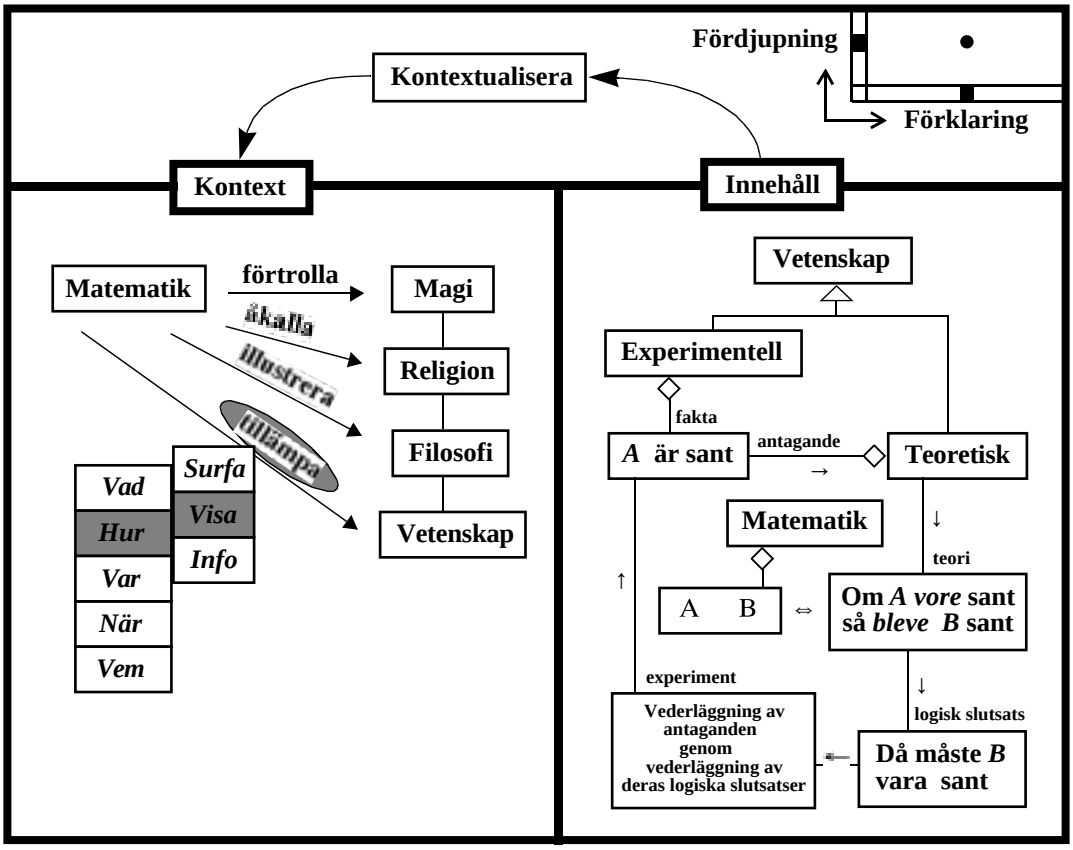


**Fig. 21. Visa innehåll: *tillämpa* under aspekten *hur*:**
**Hur tillämpar man matematik på vetenskap? - ökad fördjupningsnivå**

## 7.8 Att manipulera kontexten för ett begrepp

Hittills har vi förutsatt att vi kunnat röra oss (= surfa) mellan olika kontexter utan att beskriva hur detta egentligen har gått till. Figur (22) illustrerar en möjlig mekanism för att åstadkomma sådana *kontextförändringar.* Genom att markera t.ex. matematik och välja *ändra-kontext* får man fram en display över alla inmatade kontexter (= begreppskartor) som innehåller begreppet matematik. I figuren är kartorna “matematiken och tänkandet” respektive “matematiska ämnesområden” markerade, vilket medför att dessa begreppskartor visas upp i kontextfönstret.

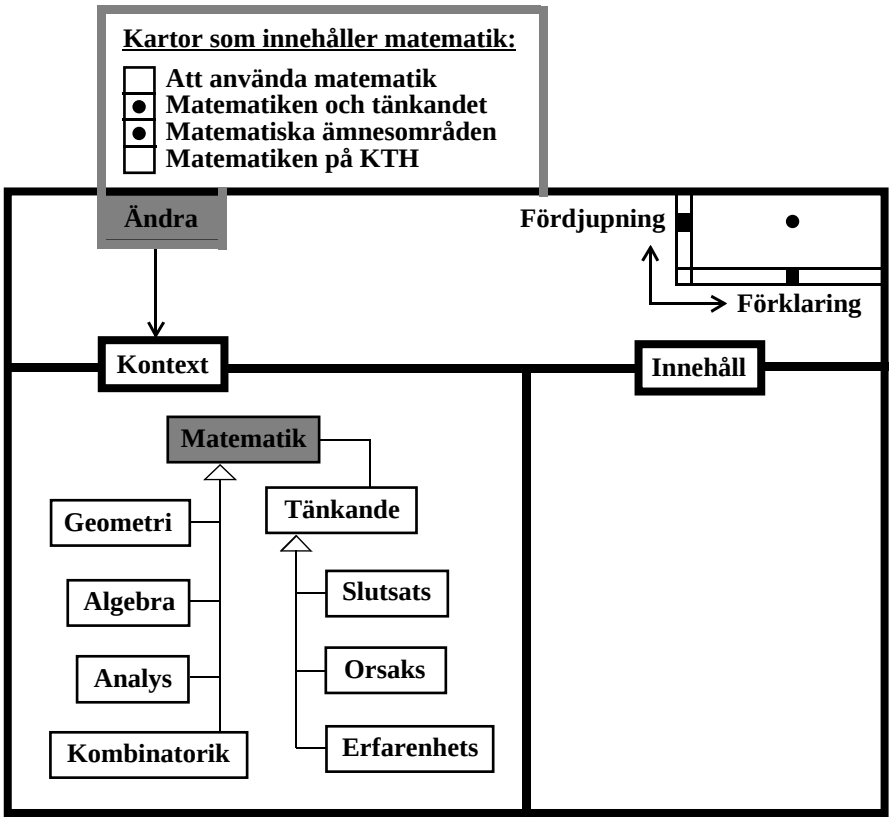


**Fig. 22. Kontext-surfning - Ändra kontext**

## 7.9 Att tilldela innehåll till en kombination av olika begrepp

Figur (22) exemplifierar ett sätt att bryta ner olika kontexter i enklare delar, vilka kan presenteras i kombination med varandra. Detta gör det möjligt att tilldela innehåll inte bara till enskilda begrepp - som vi hittills har gjort - utan till godtyckliga kombinationer av begrepp (och begreppsrelationer, vilket innebär en väsentlig ökning av användbarheten hos en begreppsbrowser. Ett exempel på presentation av innehållet i en begreppskombination visas i Figur (23).

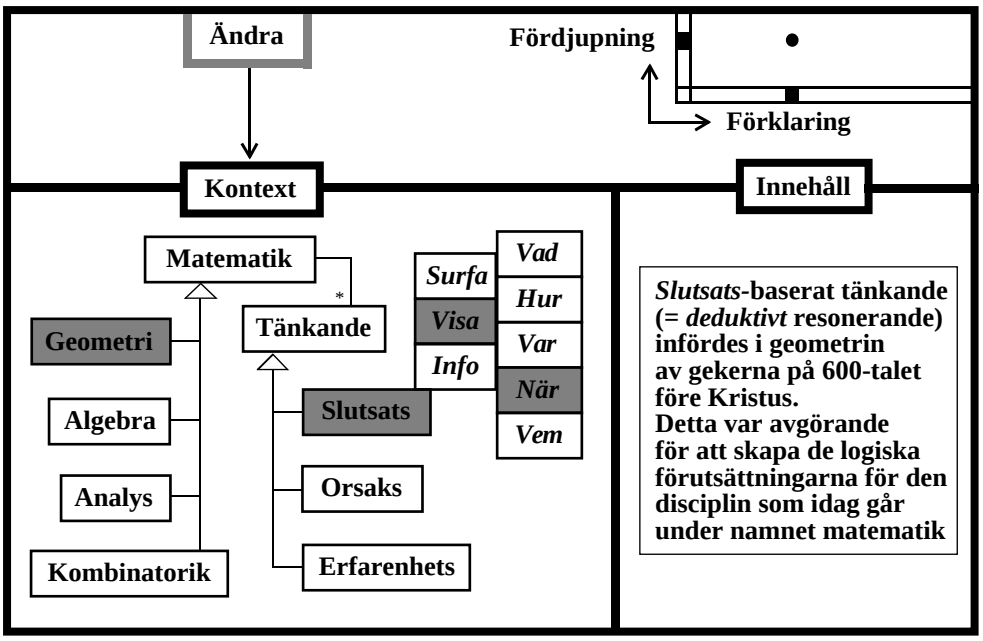


**Fig. 23. Visa innehåll: *När* introducerades *slutsats*-baserat tänkande i *geometri*?**

## 7.10 Översikt av matematiska begreppsrelationer på universitetsnivå

Av pedagogiska skäl har de matematiska begreppskartor som hittills diskuterats varit ganska enkla. Figurerna (24), (25) och (26) visar tre exempel på hur mera komplicerade matematiska begreppskartor kan se ut. Förutom att de illustrerar några av de grundläggande begreppsrelationerna i en beskrivning av matematiken på universitetsnivå, så visar de även på behovet av att kunna manipulera kontexter genom selektiv undertryckning av olika grenar av kartorna för att inte den kontextuella komplexiteten ska bli för stor.

Trots sin komplexitet förmedlar dessa kartor en kontextuell överblick som vore svår att skapa inom motsvarande utrymme med någon form av icke-diagrammatisk presentationsteknik. Dessa diagram (presenterade i sin helhet) är därför av värde som ett sätt för studenter att 'integrera' relationerna mellan olika matematiska begrepp och internalisera förståelsen för hur de hänger ihop med varandra.

De kan även fungera som grund för en kompetensdeklaration av matematiska kunskapskällor (= predikanter), som i sin tur kan bilda en utgångspunkt för uppbyggnaden av ett matematiskt helpdesk system i enlighet med de principer som diskuterats ovan i kapitel (5.11).

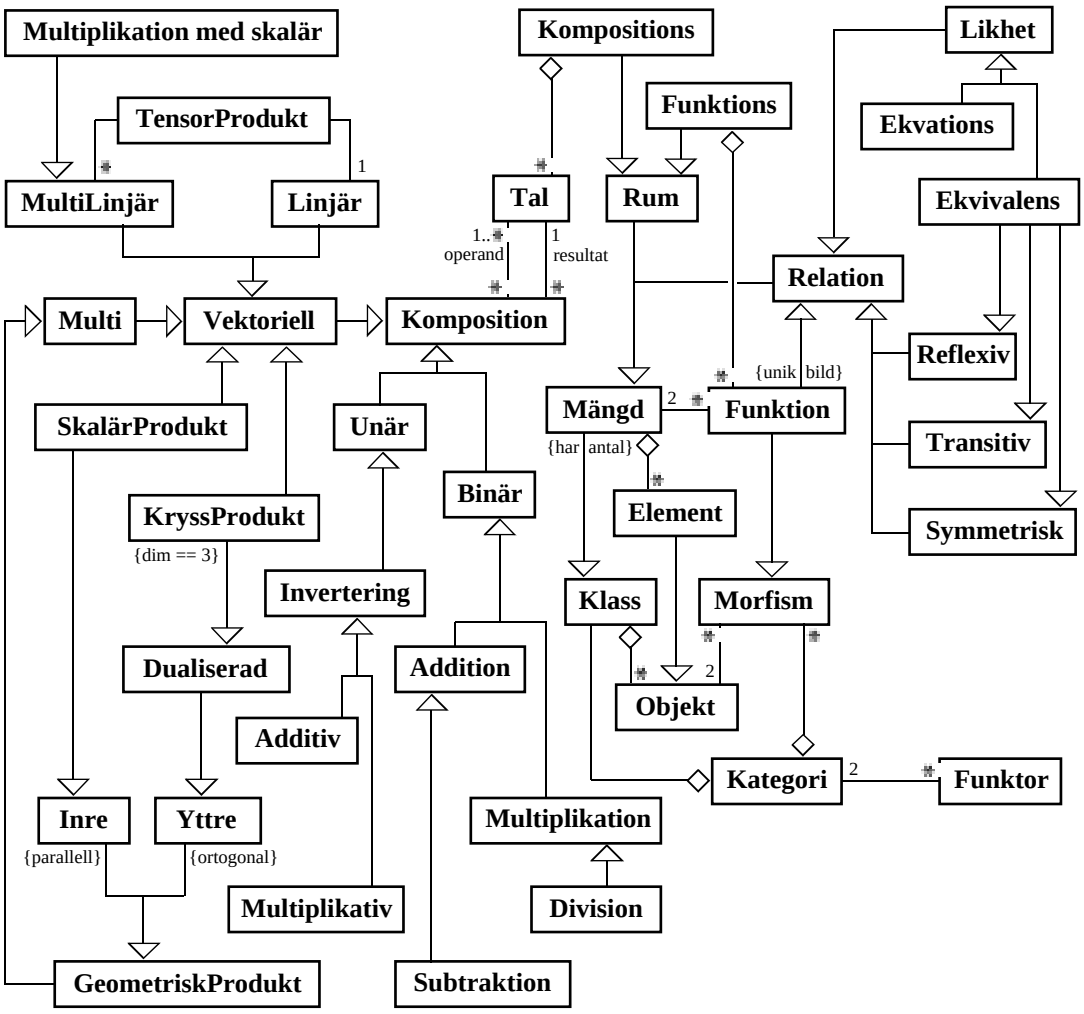


**Fig. 24.En relationsöverblick av några viktiga matematiska begrepp**

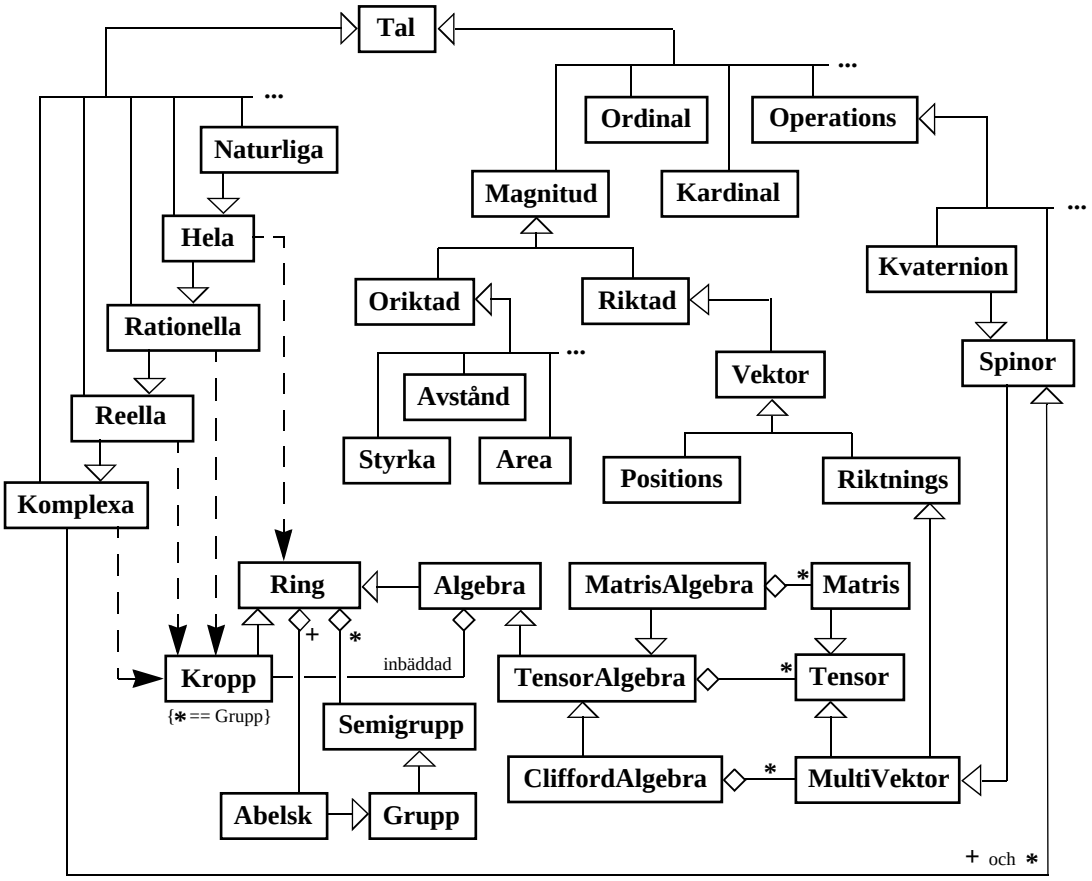


**Fig. 25. En relationsöverblick över begreppet Tal**

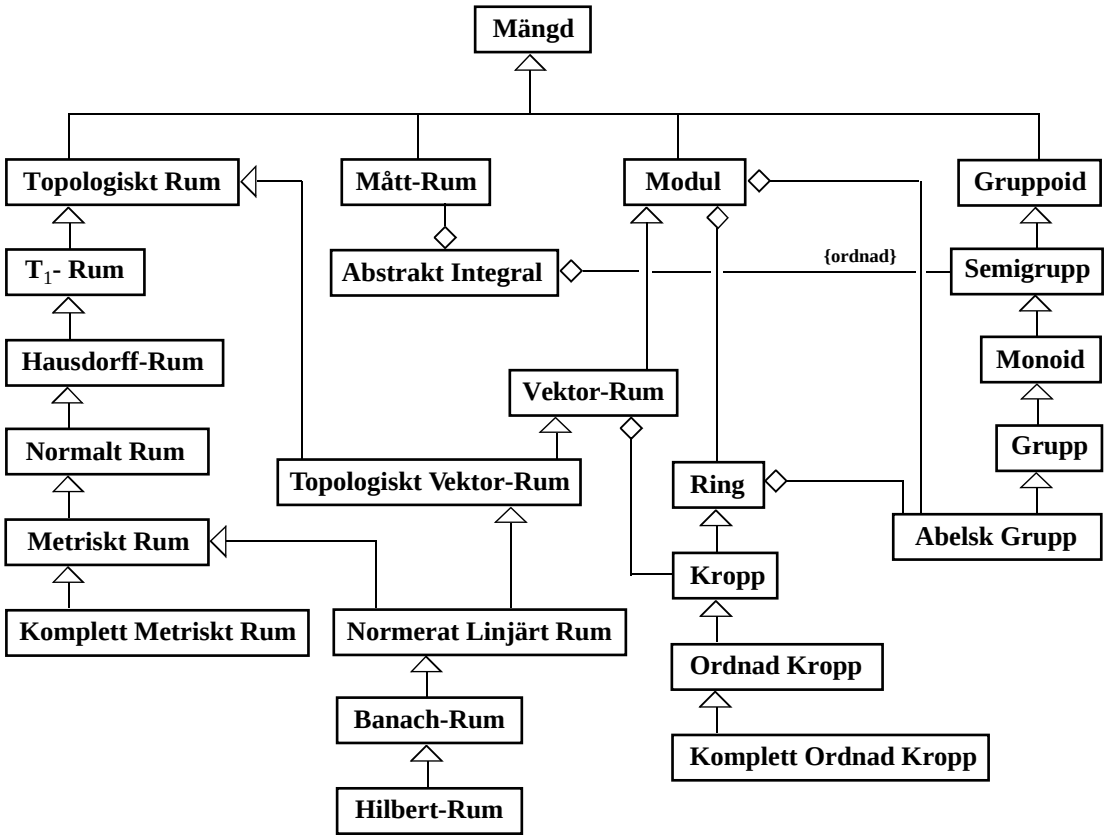


**Fig. 26. Relationer mellan några moderna matematiska begrepp med rötter i kontinuitet och aritmetik.**

# 8 Referenser


[1] Andersson, N. & Berg, M. & Danielson, M. & Gyllensporre, D. & Johnson. C. & Järvklo, Å. & Levitte, T. & Paltzer, D., *MacDrawboard - en applikation för tillämpad projektiv geometri*, projektuppgift för D-teknologer i PMI-kursen DB330 under ledning av Ambjörn Naeve på NADA & Matematik, KTH, 1988.

[2] Appelgren, J., *Reflections - a program for the interactive study of surface shape*, MsC thesis, NADA, 1994.

[3] Broady, D., *Digitala utgåvor av August Strindbergs verk - inledande överväganden*, Version 3, 1996-08-27, Centre for user-oriented IT-Design (CID), NADA, KTH, 1996.

[4] Bång, A. & Hulthén, M. & Lindberger, P. & Nedlich, K. & Persson, J. & Sundberg, J. & Svensson, C., *MacFlow - a graphical programming environment*, PMI-project assignment in course DB330. Idea and Supervision: Johan Appelgren & Ambjörn Naeve, NADA / Matematik, KTH, 1989.

[5] Carlsson, C. & Gustafsson, J. & Heimdal, M. & Lantz, J. & Lundeborg, M. & Mattsson S., *MapCon , ett System för Laborativ Matematik* - ett projekt utfört förPMI-kursen av D-teknologer i årskurs 3 under ledning av Ambjörn Naeve på NADA & Matematik, KTH, 1986.

[6] Carlsson, P & Hartikainen, M. & Jörgensen, J., *PrimeTime - en användarhandledning för Macintosh*, projektarbete inom kursen Interaktiva System för Matematikerlinjens datalogigren, Stockholms Universitet. Projektledning och matematisk grundidé: Ambjörn Naeve, NADA & Matematik, KTH, 1992.

[7] Clifford, W. K., *Applications of Grassmann's extensive algebra*, American Journal of Mathematics, Vol.

I, pp. 350-358, 1878.

[8] Clifford, W. K., *Mathematical Papers*, (ed. R. Tucker), Macmillan, London, 1882.

[9] Dahl, K. & Nordqvist, S., *Matte med mening - tänka tal och söka mönster*, Alfabeta bokförlag, 1995.

[10] Ekman, T. & Kuna, R. & Ångqvist, T. & Scocco, W. & Sandbom, J & Vivas, J. L., *Mapanalyze* - ett projekt utfört för IPM-kursen av matematikerlinjens årskurs 3 under ledning av Ambjörn Naeve på NADA & Matematik, KTH, 1989.

[11] Engström, A. & Koistinen, J. & Avatari, A. & Persson, K. & Grape, P. & Ottosson, A. & Andersson, Å. & Andersson, A., *MacTapet - användarhandledning*, projektarbete inom kursen Interaktiva System för Matematikerlinjens datalogigren, Stockholms Universitet. Projektledning och matematiskt grundarbete: Ambjörn Naeve, NADA & Matematik, KTH, 1988.

[12] Gamma, E. & Helm, R. & Johnson, R. & Vlissides, J., *Design Patterns - Elements of Reusable Object-Oriented Software*, Addison-Wesley Publ. Co., Reading MA, 1995.

[13] Grassmann, H., *Die Lineale Ausdehnungslehre - ein neuer Zweig der Mathematik*, Leipzig, 1844. (Translated to English by Lloyd C. Kannenberg, *A New Branch of Mathematics*, Open Court, 1995).

[14] Grassmann, H., *Der Ort der Hamilton'schen Quaternionen in der Ausdehnungslehre*, Mathematische Annalen, Vol. 12, p. 375, 1877.

[15] Hestenes, D., *New Foundations For Classical Mechanics*, Kluwer Academ. Publ., Dordrecht, The Netherlands, 1993 (1986).

[16] Hestenes, D. & Sobczyk, G., *Clifford Algebra to Geometric Calculus: A Unified Language for Mathematics and Physics*, D. Reidel Publishing Co., Dordrecht, The Netherlands, 1984 (1992).

[17] Högskoleverket, *Datorstödd eller datorstörd matematikundervisning*, Rapport, 1999.

[18] Kjelldahl, L. & Sundblad, Y., *Experience from 10 years of students projects oriented towards graphic interaction*, IP-lab-107, Interaction and Presentation Laboratory, NADA, KTH. Published in: *Computers and Graphics*, Vol 20, No. 3, pp. 463-471, Pergamon, 1996.

[19] Klein, F., *Vorlesungen über nicht-Euklidische Geometrie*, Springer, Berlin, 1926.

[20] Lea, D., *Concurrent Programming in Java - Design Principles and Patterns*, Addison-Wesley Publ. Co., Reading, MA, 1997.

[21] Lenman, S. & See, H. & Century, M. & Pennycook, B., *Merz: Creating Personal and Shared Spaces on the World Wide Web*, WebNet 96 - World Conference of the Web Society, Proceedings, pp. 292-297, S.F., 1996.

[22] Linde, R. & Naeve, A. & Olausson, K. & Skantz, K. & Westerlund, B. & Åsvärn K., *Kunskapens Trädgård*, Centre for User Oriented Information technology Design, CID-18, TRITA-NA-D9709, KTH, 1997.

[23] Naeve, A., *Projective Line Geometry of the Visual Operator*, Computational Vision and Active Perception Laboratory (CVAP-29), TRITA-NA-8606, KTH, 1986.

[24] Naeve, A., *On the use of Exterior Algebra in Image Analysis*, Computational Vision and Active Perception Laboratory (CVAP-30), TRITA-NA-P8709, KTH, 1987.

[25] Naeve, A. & Eklundh, J. O., *On Projective Geometry and the Recovery of 3D-structure*, Proceedings of the first International Conference on Computer Vision (ICCV), pp. 128-135, London, 1987.

[26] Naeve, A., *Geometric Modeling - A Projective Approach*, Computational Vision and Active Perception Laboratory (CVAP-63), TRITA-NA-P8918, KTH, Stockholm, Sweden, 1989.

[27] Naeve, A., *Focal Shape Geometry of Surfaces in Euclidean Space*, Dissertation, Computational Vision and Active Perception Laboratory (CVAP-130), TRITA-NA-P9319, KTH, Stockholm, Sweden, 1993.

[28] Naeve, A., *The Mathemagic of Wallpaper Patterns* (*Tapetmönstrens Matemagi*), lecture given at the Swedish Mathematical Society, Västerås, 18 Mars 1995, at the 9:th Mathematics Biennal, Sundsvall, 26 January 1996, at the Mathematics Biennette 97, Stockholm, 25 January 1997. Proceedings of the 9:th Mathematics Biennal , pp. 307-311, Mitthögskolan, Sundsvall, 1996.

[29] Naeve, A. & Eklundh, J. O., *Representing Generalized Cylinders*, Proceedings of the Europe-China Workshop on Geometrical Modeling & Invariants for Computer Vision, pp. 63-70, Xi'an, April 27-29, 1995. Published by Xidan University Press, Xi'an, China, 1995.

[30] Naeve, A., *The Garden of Knowledge as a Knowledge Manifold - a conceptual framework for computer supported subjective education*, Centre for user oriented Information technology Design (CID-17),

TRITA-NA-D9708, KTH, Stockholm, Sweden, 1997.

[31] Naeve, A., *Structure From Translational Observer Motion*, Proceedings of the International Workshop on Algebraic Frames for the Perception-Action Cycle - Trends in the conceptualization, design and implementation of artificial autonomous systems, Kiel, Germany, September 8-9, 1997. Published in Springer, Lecture Notes In Computer Science, Vol. 1315, pp. 235-248.

[32] Naeve, A. & Svensson, L., *Geo-MAP Unification*, to be published as chapter 6 in: Sommer (ed), *Geometric Computing using Clifford Algebra*, Springer, 1999.

[33] Naeve, A., *Conceptual Navigation and Multiple scale Narration in a Knowledge Manifold*, Centre for user oriented Information technology Design, (work in progress).

[34] Odell & Martin, *Object-Oriented Methods - a Foundation*, Prentice Hall, 1998.

[35] Rumbaugh, J. & Blaha, M. & Premerlani, W. & Eddy, F. & Lorensen, W., *Object-Oriented Modeling and Design*, Prentice Hall, New Jersey, 1991.

[36] Rumbaugh, J. & Jacobson, I. & Booch, G., *The Unified Modeling Language Reference Manual*, Addison Wesley Longman Inc., 1999.

[37] Tollmar, K. m. fl., *Drawboard - Projektiv Geometri i Praktiken*, projektuppgift av elever vid Matematikerlinjens datalogigren, Stockholms Universitet under ledning av Ambjörn Naeve på NADA & Matematik, KTH, 1987.

[38] Winroth, H., *Dynamic Projective Geometry*, Dissertation, March 1999, Computational Vision and Active Perception Laboratory (CVAP), TRITA-NA-99/01, NADA, KTH, Stockholm, Sweden.

[39] Wolfram, S., *Mathematica - a System for Doing Mathematics by Computer*, Addison-Wesley, 1996 (1988).

# APPENDIX

## 9 Notation för begreppsmodellering

I denna text förekommer ett antal diagram - de flesta uttryckta i begreppsmodelleringsspråket UML (Unified Modeling Language). Diagrammen är avsedda att beskriva viktiga begrepp och relationer på ett grafiskt sätt, och därigenom bryta den totalordning som framtvingas av varje verbal beskrivning av motsvarande begreppsrelationer. Detta bidrar till att skapa större överblick och förståelse.

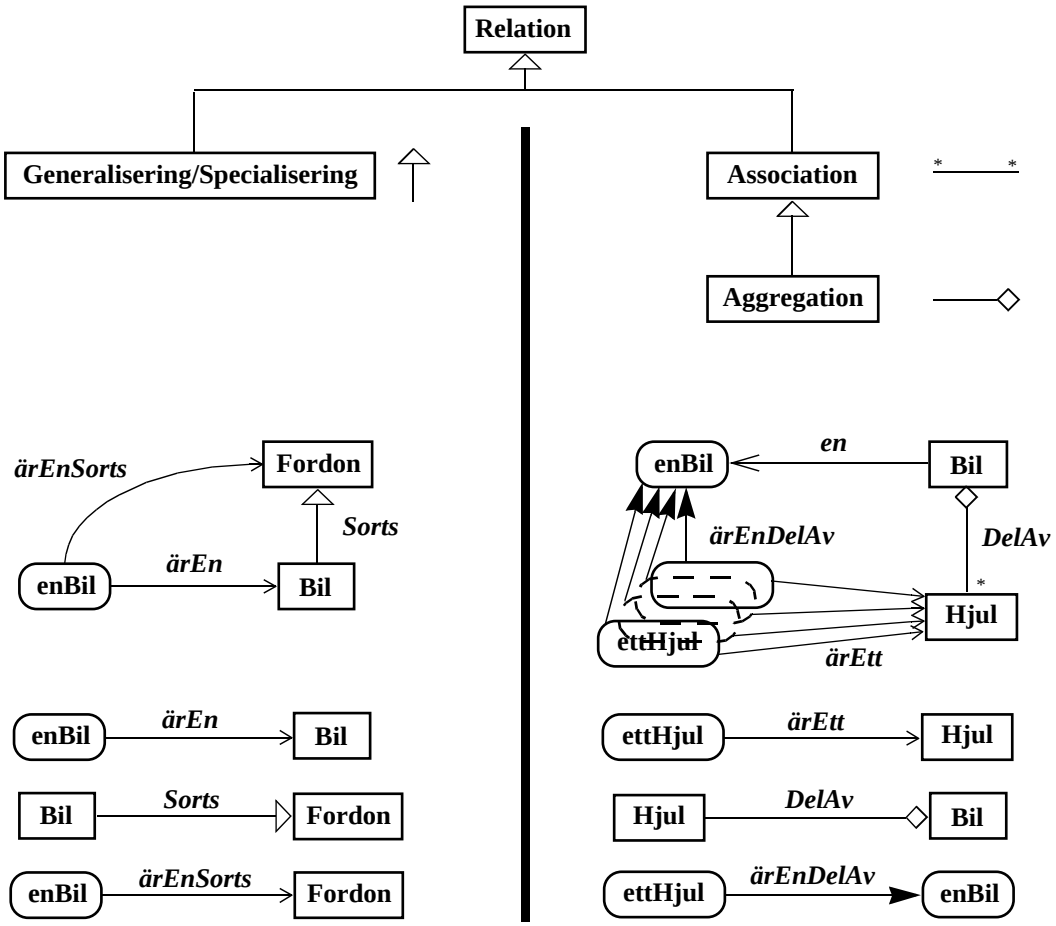


**Fig. 27.** UML notation med tillägg av relationerna *ärEn*, *ärEnSorts*, *en* och *ärEnDelAv*.

De grundläggande relationerna mellan begreppen - och deras respektive uttrycksformer i UML är beskrivna i Figur (27). För en fördjupande beskrivning av UML se t.ex. [(36)].

# 10 IMS-projektet

## 10.1 Bakgrund

Att betrakta läromiljöer som växelverkande distribuerade system leder till en modellbeskrivning grundad på interagerande delar. Idén om att delar av inlärningsprocessen och läromaterialet kan modulariseras ger upphov till en förskjutning mot en mer diversifierad och elevdriven läromiljö. Ett exempel på detta är den omvittnade populariteten hos olika skräddarsydda kurspaket med speciellt utvalda läsavsnitt och övningar i förhållande till de gamla statiska och standardiserade läroböckerna. Ett annat exempel på individualisering av inlärningsprocessen är möjligheten att inleda en kurs med en test som antingen godkänner eleven direkt eller skapar ett individualiserat inlärningsmaterial som syftar till att uppnå godkänt på kursen.

Denna syn på inlärning som ett distribuerat system av interagerande delar avspeglas i en motsvarande förskjutning av modellerna för teknologi och affärsverksamhet i riktning mot en nätverksstruktur som kommer att understödja framväxten av en distribuerad inlärningsindustri. På teknologisidan är både hårdvaru- och mjukvaruplattformar designade på ett sätt som stödjer modulära och distribuerade system. Den enskilda arbetsstationen utgör inte längre den primära plattformen för datorstödd inlärning, utan det är i stället det sammankopplade nätverket av ”inlärningsstationer” som representerar den grundläggande inlärningsmiljön. I detta sammanhang erbjuder kunskaperna inom området objektorientering ett antal erfarenhetsbaserade principer och verktyg för analys och design av modulära och distribuerade inlärningsmiljöer.

Parallellt med detta växer det på affärssidan fram en beskrivning av inlärningsmiljöer som dynamiska system med diversifierade behov – i motsats till den gamla ”enhetsmodellen” med fixerade kurser och läroböcker. Möjligheten att enkelt skräddarsy material och miljöer stöds av ett system av sammanlänkade delar. Olika affärsidéer och strukturer tar fasta på att utforska hur man kan stödja framväxten av flexibla mekanismer för sammanfogning av innehållskomponenter – i motsats till statiska ”läroboksmodeller” med en ”allt-eller-inget” typ av försäljningsstruktur. Möjligheterna att modularisera innehållet och sälja olika delar i olika sammanhang utgör här ett affärsmässigt lockande perspektiv.

## 10.2 Intressenter, aktörer och grundläggande syfte

Den explosionsartade utvecklingen av Internet under senare år har skapat oerhörda möjligheter för interaktiva och individcentrerade former av kommunikation. Bland operatörerna på Internet råder emellertid en allmän insikt om att utvecklingen av verktyg för intelligent informationshantering - för t.ex. interaktiva läromiljöer och elektronisk handel - allvarligt begränsas av bristen på allmänt accepterade specifikationer och standardiserade beskrivningar av det tillgängliga informationsinnehållet. Denna brist på en övergripande ”metadata-struktur” motverkar den nödvändiga resursmodularisering som möjliggör effektiv samverkan mellan olika aktörer i form av effektiv lokalisering och återanvändning av resurskomponenter i olika miljöer och på olika plattformar.

Det pågår idag ett antal projekt för att stödja såväl framväxten av sammankopplingsbara resurskomponenter som standardiserade beskrivningar av sådana komponenter med hjälp av metadata. Det mest övergripande av dessa projekt är IMS (Instructional Management Systems) projektet (se http://www.imsproject.org). Projektet bedrivs i form av ett partnerskap mellan

olika akademiska, kommersiella och statliga organisationer, med stöd av EDUCOM och under överinseende av såväl NLII (the National Learning Infrastructure Initiative) som W3C (the World Wide Web Consortium).

Bland de tyngsta intressenterna i IMS projektet (IMS Investment Members) kan nämnas:
Buena Vista University, California State University (CSU), COLLEGIS Research Institute, Educational Testing Service, George Mason University, University of California, University of North Carolina at Chapel Hill, University of Michigan, Committee on Institutional Cooperation (CIC), Miami Dade Community College, Apple Computer, Empower Corporation, Farance, Inc., IBM Education, Sun Microsystems, International Thomson Publishing, Joint Information Systems Committee, KPMG Peat Marwick, Microsoft, National Institute for Standards and Technology (NIST), Oracle, UNISYS, US Department of Defense.

Några exempel på organisationer och projekt som samarbetar med IMS-projektet är:
WWW Consortium (W3C), Object Management Group (OMG), The Open Group, IEEE P1484 Personal Learning Systems (PLS) Initiative, The Learning Objects Metadata Group (LOMG), Advanced Technology Program (ATP), The Educational Object Economy, Software Publisher's Association, International Federation of Library Associations and Institutions (IFLA.net), Digital Libraray Resources and Projects, University of Michigan's Digital Library Project, Internet2, Ca*net2, Coalition for Networked Information, Association of Research Libraries, Cause, Western Interstate Commission for Higher Education, Western Governor's University, The Asynchronous Learning Network.

IMS-projektet identifierar fyra grundläggande typer av aktörer (intressenter) i det man kallar det expanderande lärolandskapet (the expanding learning landscape):

• elever (learners),
• lärare (teachers),
• läromedelsutveckare (providers) och
• administratörer (coordinators).

Deras roller påverkas och förändras av utvecklingen, vilken man beskriver genom ett antal förändringstendenser i lärmiljön:

| FRÅN | I RIKTNING MOT |
|---|---|
| ämnesspecialisering | ämnesövergripande |
| examensinriktad | livslångt lärande |
| envägsinriktad (lärare --> elev) | interaktiv |
| klassrumsbaserad | gruppdynamisk (communities) |
| kollektivorienterad (klass) | individorienterad |
| punktmässigt utvärderad | iterativt utvärderad |
| isolerade problemområden | inbäddade simuleringar |
| tids- och rumsberoende | tids- och rumsoberoende |

IMS-projektet syftar bl.a. till att:

• fokusera inlärningen på elevens intresse och behov.
• möjliggöra inlärning var som helst och när som helst.

• tillåta större grad av flexibilitet och individualiserad anpassning av inlärningsmiljön.
• understödja en ökad grad av interaktion mellan lärare och elever.
• skydda och kompensera det intellektuella kapital som är nedlagt i resurskomponenterna.
• öka den innehållsmässiga kvaliteten hos resurskomponenterna.
• förbättra sökmöjligheter genom att stödja innehållsdeklaration av resurskomponenterna.

IMS-projektet håller på att ta fram ett antal specifikationer av en lärmiljö som strävar att uppfylla ovanstående mål. Genom att dra nytta av erfarenheter från framgångsrika pilotprojekt ämnar man inrikta specifikationerna på att föreskriva på vilka sätt olika resurskomponenter ska kunna samverka med varandra - utan att låsa sig vid hur denna samverkan skall implementeras. ”Don’t build what already exists” och ”Leverage what already works” är härvidlag två grundläggande principer.

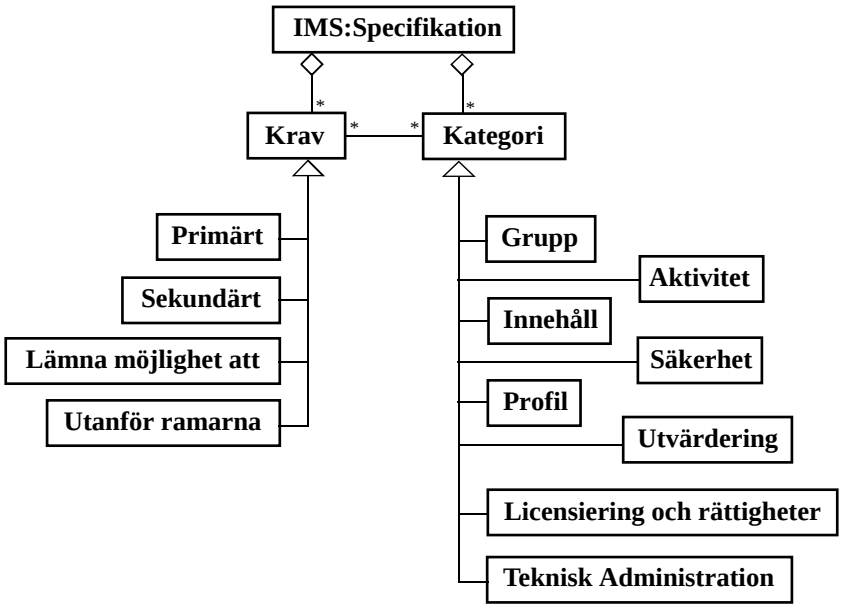


**Fig. 28. Olika typer av krav och kategorier som IMS projektet arbetar med**

I enlighet med denna filosofi skall IMS specifikationerna tillhandahålla gränssnitt för att (utan att föreskriva hur man ska):

• fastställa kostnader för resurskomponenter.
• beskriva informationsinnehållet i resurskomponenter med hjälp av metadata.
• utveckla eller välja pedagogik och utvärderingsmetodik.
• lagra och återfinna information.
• anta och examinera elever.
• konstruera användargränssnitt för hantering av resurskomponenter.

En första version av IMS-projektets specifikation för metadata släpptes den 23/3-98. Den bygger på det s.k. Dublin Core konceptet, som består av 15 stycken beskrivningselement för information. De är framtagna genom en treårig samarbetsprocess där ledande experter inom

bibliotek, nätverk och digital biblioteksforskning har samverkat för att uppnå en interdisciplinär och internationell konsensus kring grunderna för hantering av informationsbeskrivning.

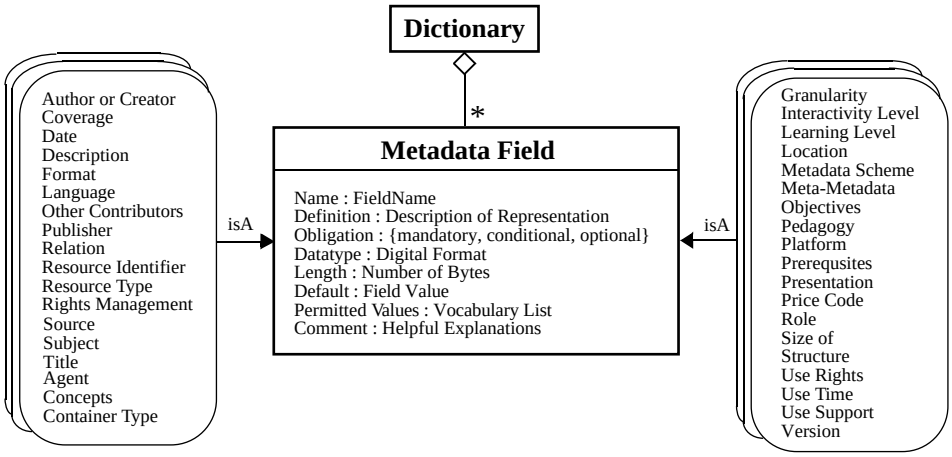


**Fig. 29. IMS projektets metadatastruktur**

## 10.3 Marknadskrafternas uppgift

IMS-projektet förlitar sig på marknadskrafterna för att hantera bland annat:

• Resurskomponenternas kvalitet:

IMS-specifikationen föreskriver inga kvalitativa restriktioner av innehållet. Marknaden kommer att tillhandahålla relevanta mekanismer för att möjliggöra för lärare och elever att identifiera resurskomponenter av hög kvalitet.

• Resurskomponenternas storlek:

IMS-specifikationen lägger inga begränsningar på storleken hos resurskomponenterna, men man förutser att eftersom dessa kommer att enkelt kunna sammanfogas, kommer marknaden att uppmuntra producenterna att bryta ner det traditionella inlärningsmaterialet i mindre, återanvändbara moduler.

• Utvärderings- och examinationsstrukturer:

IMS-specifikationen kommer att avspegla nuvarande utvärderingsformer för elevprestationer, men den kommer även att tillåta utveckling av de nya utvärderingsmetoder för utbildning som kommer att uppstå för att möta marknadens krav.

Sett från ett konsumentperspektiv ligger de drivande marknadskrafterna bakom IMS projektet i dess potential för att öka utbildningens kvalitet, förbättra dess tillgänglighet och reducera dess kostnader. Kommersiella organisationer kommer att kunna uppnå marknadsfördelar genom:

• Visuella gränssnitt:

utvecklingen av sådana utgör ett område där företag och andra organisationer kommer att tillföra värde, genom utveckling och tävling med varandra (på samma sätt som det idag finns många olika Internet gränssnitt för hantering av e-post som arbetar mot samma underliggande standardiserade e-post protokoll).

• Skräddarsydd funktionalitet:

IMS specifikationerna är avsiktligt tillåtande i motsats till restriktiva, vilket erbjuder många tillfällen till lokala tillägg till den grundläggande funktionaliteten. Sådana tillägg kommer att användas av IMS-kompatibla aktörer för att utveckla marknadsfördelar för sina produkter.

• Marknadstillväxt:

Den nuvarande tillgången till online-resurser för inlärning är otillräcklig. Utvecklingen av IMS-kompatibla system kommer att stimulera utvecklingen av sådana produkter och därigenom skapa en större marknad med ökad avkastning för framgångsrika aktörer.

## 10.4 IMS-projektets framtida roll

Det initiala uppdraget för IMS-projektet består i att utveckla och sprida ovan nämnda specifikationer. På längre sikt finns behov av en nedbantad organisation med en eller flera av följande potentiella roller:

• Fortsätta utvecklingen av specifikationer samt underlätta utvecklingen av fritt tillgänglig programkod.

• Medverka till konvergens hos olika kommersiella implementationer av IMS, vilka har divergerat från den existerande standarden. Denna funktion är analog med den nuvarande rollen hos W3C vad gäller anpassning av marknadens olika utvidgningar av HTML.

• Upprätthålla ett register av auktoriserade organisationer med rätt att utfärda IMS-identiteter för att säkerställa unicitet hos resurskomponenternas IMS ID-nummer.

• Bidra till att etablera metoder för certifiering av IMS-kompatibilitet hos resurskomponenter och lärmiljöer.