

DATABASSTRUKTURER FÖR MODELLERING AV
BYGGPROJEKT, EN JÄMFÖRELSE MELLAN GDS,
MEDUSA OCH INTERGRAPH.

Ett examensarbete utfört av Per Waernborg
vid Lunds Tekniska Högskola Sektionen för
Väg och Vattenbyggnad.

Lund VT 1985

FÖRORD

Detta examensarbete har utförts vid Lunds Tekniska Högskola, sektionen för Väg och Vattenbyggnad.

Syftet med examensarbetet har varit att göra en jämförelse av modellering i tre olika CAD-system. Inriktningen har varit mot CAD-systemens användning som design-system mer än som rit-system. Arbetet har till största del genomförts genom litteraturstudier och intervjuer. Där inte fakta helt räckt till har jag fått komplettera med egna antaganden och erfarenheter från systemkunniga för att knyta ihop det hela. Ämnesområdet är mycket stort varför jag har varit tvungen att göra kraftiga begränsningar och lämna många områden lämpliga för mera ingående studium till framtiden. Detta arbete skall ses som ett inledande studium av problemområdet. Pågående forskningsarbete bedrivs parallellt med detta arbete av Per Christiansson och Alberto Herrera på avdelningen för bärande konstruktioner vid Lunds Tekniska Högskola.

Min handledare, Per Christiansson, har gett många goda råd och hjälpt mig med arbetet.

Dessutom vill jag gärna tacka följande personer som villigt ställt upp med goda svar på våra frågor.

Sten Forsler	VBB
Alberto Herrera	LTH
Anders Olausson	DAPAB

Lund 850419

Per Waernborg

INNEHÅLLSFÖRTECKNING

1. INLEDNING

- 1.1 Bakgrund
- 1.2 Syfte
- 1.3 Begränsningar
- 1.4 Arbetsmetod
- 1.5 Definitioner
- 1.6 Integrering
- 1.7 CAD-systemet i byggprocessen
- 1.8 Tidsrelatering

2. BEDÖMMNINGSGRUNDER

- 2.1 Söktiden
- 2.2 Minnesutnyttjande och minneskapacitet
- 2.3 Nya data
- 2.4 Radering
- 2.5 Uppdatering
- 2.6 Säkerhet

3. DE TRE OLIKA DATASTRUKTURERNA

- 3.1 Hierarkisk databas
- 3.2 Relationsdatabas
- 3.3 Nätverksdatabas

4. LAGERINDELNING OCH OBJEKTSINDELNING

- 4.1 Sammanfattning
- 4.2 Exempel

5. ALLMÄNT OM RITSYSTEMEN

- 5.1 GDS
- 5.2 Medusa
- 5.3 Intergraph

6. HJÄLPDATABASERNA

- 6.1 Allmänt
- 6.2 En katalog (Exempel från Intergraph)

7. PROJEKTDATABASERNA

- 7.1 GDS
- 7.2 Medusa
- 7.3 Intergraph
- 7.4 Sammanfattning

8. MODELLERING I GDS

- 8.1 Allmänt
- 8.2 Hierarkien
- 8.3 Grupper
- 8.4 Objekt
- 8.5 Grafiska block
- 8.6 Text block

9. MODELLERING I MEDUSA

- 9.1 Allmänt
- 9.2 Hierarkien
- 9.3 De lägre elementklasserna
 - 9.3.1 Linjer
 - 9.3.2 Texter
 - 9.3.3 Primmar
- 9.4 De högre elementklasserna
 - 9.4.1 Klumpar
 - 9.4.2 Ritningsblanketter
 - 9.4.3 Symboler
- 9.5 Avsökning av datastrukturen

10. MODELLERING I INTERGRAPH

- 10.1 Allmänt
- 10.2 Egenskapsdatabasen
- 10.3 Grafikdatabasen
- 10.4 Arbetsmetodik
- 10.5 Sammanfattning

11. SKISSNING OCH ÄNDRINGAR I DETALJERINGSGRAD

- 11.1 Varför skissa i CAD-system?
- 11.2 Skissning i terrängmodell
- 11.3 Ändringar i detaljeringsgrad
 - 11.3.1 GDS
 - 11.3.2 Medusa
 - 11.3.3 Intergraph
- 11.4 Säkerhet
- 11.5 Sammanfattning
- 11.6 Kommentar

12. KONVERTERINGSFORMAT

- 12.1 Allmänt
- 12.2 Lagerindelning
- 12.3 Kopiering
- 12.4 Sammanfattning

13. ARMERINGSRITNING (Exempel)

- 13.1 Inledning
- 13.2 GDS
- 13.3 Medusa och Intergraph

14. BYTE AV DÖRRAR (Exempel)

- 14.1 Inledning
- 14.2 GDS
- 14.3 Medusa
- 14.4 Intergraph

15. SLUTSATSER

- 15.1 Allmänt
- 15.2 GDS
- 15.3 Medusa
- 15.4 Intergraph
- 15.5 Slutord

16. LITTERATURFÖRTECKNING

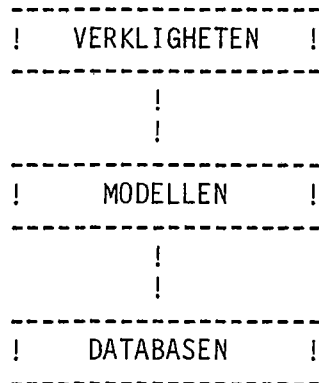
1. INLEDNING

1.1 Bakgrund

Dataresurser för stöd i byggnadstekniskt projekteringsarbete har utvecklats snabbt under de senaste åren vilket inneburit att så kallad CAD (Computer Aided Design) system allt mer användes i praktiskt projekteringsarbete. Hittills har huvudsakligen större konsulter (mer än 20 anställda) haft råd att investera i avancerade CAD-system. Efterhand som mjuk- och hårdvara utvecklats och blivit billigare börjar nu även mindre firmor investera i datorstödda projekteringsystem.

Dagens CAD-system förknippas ofta med framtagandet av detaljerade ritningar i ett sent skede av en projektering (Computer Aided Draughting). Emellertid skapas en modell av det verkliga projektet i datorsystemet som i vissa fall mycket riktigt ligger nära dess redovisning i form av ritningar.

Projekteringsprocesser påverkar givetvis utformningen av de datorstödda projekteringshjälpmedlen liksom påverkan även sker i motsatt riktning. Mellan de verkliga problemen och dess implicering i dataresurser befinner sig ett mellanområde där verkligheten modelleras med de verktyg som systemen erbjuder. Examensarbetet omfattar endast den delen av struktureringsarbetet som omfattar beskrivning av projektet, och inte den resterande delen av processen såsom t.ex. administrativa rutiner och rutiner för styrning av byggprocessen.



1.2 Syfte

Att undersöka hur modellering av projektinformation kan ske i olika CAD-system (GDS, Medusa, Intergraph). Passar modellerna speciellt bra för vissa delmodeller. Examensarbetet utgör ett inledande studium och en introduktion till problemområdet.

1.3 Begränsningar

Här tas endast upp tre CAD-system även om marknaden erbjuder många fler. En annan viktig begränsning är att systemen endast studeras ur synvinkeln byggande och konstruerande. Det som diskuteras här är till stor del tillämpligt även inom andra användningsområden men skillnaderna kan på sina ställen vara avsevärda. Man måste även tänka på att utvecklingen är snabb och att systemen uppdateras hela tiden, vilket innebär att det som jag funnit då jag gjort mina studier kanske inte alls stämmer med fakta i ett uppdaterat system.

Databaserna och dess möjligheter till modellering är det enda som tas upp, detta innebär inte att ett system med en bra databas är ett bra system, det finns även många andra aspekter som måste beaktas. Databaserna är dessutom bara teoretiskt undersökta de kan därför till synes vara bra men i praktiken p.g.a. programmeringstekniska orsaker vara dåliga.

Större delen av litteraturen på vilken jag baserar mitt examensarbete är på engelska. Detta medför problem vid ordvalet då man ska översätta engelska facktermer till svenska. De största problemen uppstår då flera engelska ord är lämpliga att översätta till ett och samma svenska ord. Mitt mål har varit att de svenska orden skall vara direkta översättningar av de engelska och även väl beskriva vad som menas.

Det finns två olika metoder att göra tredimensionella bilder på, antingen bygger man direkt upp en tredimensionell modell och tar sedan framvyer från dessa, eller så gör man endast tvådimensionella ritningar och då man sedan önskar framställa en tredimensionell bild sätter man ihop de tvådimensionella ritningarna till en tredimensionell modell. Denna skillnad är viktig men berörs inte vidare.

1.4 Arbetsmetod

Arbetet har till större del bestått av litteraturstudier och studier av manualer. Dessutom har intervjuer gjorts med systemansvariga på byggföretag.

1.5 Definitioner

Elementidentifikation:

I alla de tre systemen ges olika former av elementidentifikation. Elementidentifikation innebär att man på något sätt markerar de element i ritningarna. Denna märkning ser olika ut i de olika systemen.

Exempel:

- På vissa väsentliga element sätter man ett nummer. Överkantsarmering märkes med en etta. Underkantsarmering märks med en tvåa.
- Användning av olika linjetyper. Överkantsarmering ritas med "streck-punkt-streck-punkt". Underkantsarmering ritas med "streck-punkt-punkt-streck-punkt-punkt".

Systemet med elementidentifikation är ett komplement till struktureringen i databasen, men har egentligen inte med själva datastrukturen att göra.

Fönster:

En ritareal innehåller normalt sett så mycket information att man inte samtidigt kan visa allt på en dataskärm. Därför använder man dataskärmen som ett fönster genom vilket man kan se en liten del av vad som finns i ritarealen. Detta fönster kan flyttas runt på ritarealen och det kan ges olika storlek d.v.s. olika skala. Man kan även definiera flera olika fönster som man kan hoppa mellan. Fönstren skiljer sig något mellan de tre olika systemen men detta har ringa betydelse för studiet av databaserna varför det inte vidare kommer att beröras.

Grupper:

I GDS använder man en indelning i "drawings" direkt översatt till svenska skulle detta betyda ritningar, men detta överensstämmer inte med vad vi vanligtvis menar med en ritning. Valet av namnet grupper förstås senare väl av sammanhanget.

Katalog:

Katalog är ett uttryck som kan ha flera betydelser här kommer det endast att användas i en. Nämligen som beteckning på en databas där man lagrar objekt och relationer mellan objekt, fungerande ungefär som ett uppslagsverk eller register.

Lager:

I Medusa använder man en indelningsform som kallas för lager. I Intergraph finns det en indelningsform som kallas för nivåer. Detta är i praktiken bara två olika namn för en och samma sak. Fortsättningsvis kommer endast benämningen lager att användas. Med lager menas en indelning som går tvärs igen databasen utan hänsyn till datastrukturen. Denna indelning får inte förväxlas med indelningen i nivåer.

Nivåer:

En indelning i nivåer är det samma som att ett element kan innehålla ett eller flera andra element i en hierarkisk struktur. Nivåindelningen får ej förväxlas med indelningen i lager.

Pekare:

Istället för att i en minnescell lagra ett tal ett tecken eller dylikt så kan man lagra en referens till en annan minnescell, denna referens kallas för en pekare. Pekare brukar ofta representeras med en pil vilken även bäst beskriver vad en pekare är för något.

Ritareal:

Detta är ett område i vilket man kan "rita". Ytan av en ritareal är stor, att kalla ritarealen för ritning är fel. Ur ritarealen kan man ta ut små delar och mata ut dessa som ritningar. I ritarealen ritar man utan skala.

Ritningsblankett:

En ritareal och en ritningsblankett är i stort sett samma sak, den väsentliga skillnaden är att man på ritningsblanketten ritar i skala.

Svart box:

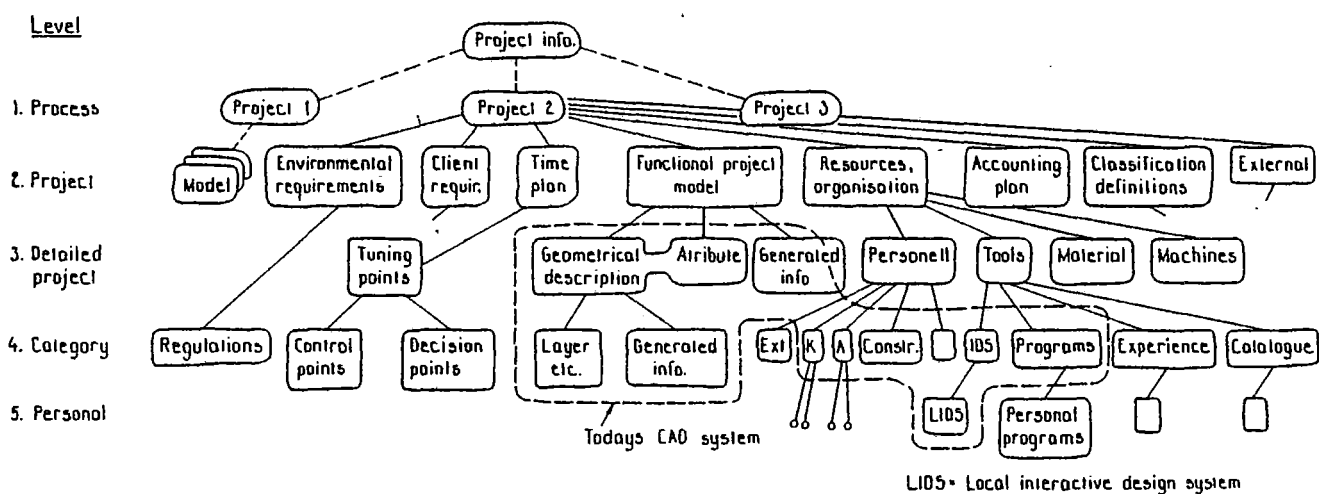
En Svart box får man när ett program, en datastruktur eller något liknande inte längre går att ändra och är svåra att genomskåda. Anledningarna till att den inte går att förändra kan vara av olika slag men de flesta bottnar i programmets komplexitet.

1.6 Integrering

Integrering av olika processdeltagare är mycket väsentlig, d.v.s. samröret mellan konstruktören, arkitekten, elektrikern, vvs-konstruktören o.s.v. Den möjlighet till integrering som de här studerade CAD-systemen erbjuder är lagerindelning. Detta utnyttjar man genom att tilldela varje processdeltagare ett antal lager.

1.7 CAD-systemet i byggprocessen

Då man tittar på CAD-systemet får man hela tiden ha i åtanke vilken plats det har i den totala byggprocessen. CAD-systemet är ju en liten del i den stora helheten. I framtiden kan man tänka sig att datorer och CAD-system kommer att ha en betydligt större omfattning och beröra större delar i byggandet än vad de gör i dagens läge. (Se bild.)



CAD-systemets plats i byggprocessen. (/CHR83a/)

1.8 Tidsrelatering

Det är ofta önskvärt att få ut tidsrelaterade data ur ett CAD-system, så att man t.ex. vet hur stora mängder av olika materiel som behövs i olika tidsskeden av ett byggprojekt. Det finns två olika möjligheter att tidsrelatera data, antingen lagrar man den tidsrelaterade datan i kronologisk ordning i databasen, eller så lagrar man datan utan någon tidsordning i databasen och använder en särskild procedur till att sortera datan tidsmässigt då man vill ha ut dem tidsrelaterade. Väljer man att sortera datan i tidsrelaterad ordning direkt i databasen medför det att man komplicerar det normala användandet av CAD-systemet i projekteringsarbetet. Detta är normalt sett inte acceptabelt.

2. BEDÖMNINGSGRUNDER

2.1 Söktiden

Söktiden är en viktig bedömningsgrund. Idag är det minneskapaciteten och söktiden som är gränssättande för vad ett CAD-system klarar av. Med tanke på den utveckling som har varit får man anta att söktiden även i framtiden kommer att vara av stort intresse. Detta just p.g.a. att söktiderna hela tiden minskar och därmed medför att man kan göra saker som man tidigare inte haft möjligheter till.

Söktiden är något som klart påverkar alla större datastrukturer och inte minst de som användes i dagens CAD-system. Men en hård satsning på korta söktider medför lätt problem med införande av nya data, radering, uppdatering o.d. Dessa problem i sin tur kan medföra att söktiderna försämras. Vill man göra en extremt söksnabb konstruktion är man tvungen att skraddarsy den för såväl användare som användningsområde.

2.2 Minnesutnyttjande och minneskapacitet

Hur väl man använder minnet är något som klart påverkar söktider och kapacitet, eftersom det tar lång tid att söka i ett minne som har mycket inaktuella data eller där en och samma information lagras på flera olika ställen.

Du kan i dagens läge få en liten dator med god minneskapacitet till ett "lågt" pris. Vill man däremot arbeta med datorer i lite större skala är minneskapaciteten ofta ett problem, eftersom en ökad minneskapacitet ger ökade möjligheter till komplicerade program. De komplicerade programmen i sin tur ställer sedan högre krav på minnet. Frågan är om minnet och söktiderna någonsin kommer att bli så utvecklade att vidare utveckling är ointressant. Detta speciellt i ljuset av de kapacitetskrävande kunskapsbaserade systemen som börjar se dagens ljus.

Ett väl utnyttjat minne kan i vissa fall vara till nackdel eftersom det gärna medför en komplex struktur som kan göra databasen till en svart box med medföljande problem.

2.3 Nya data

Att sätta in nya data i en databas bör alltid vara lätt eftersom nya data är något som kan förutsättas tillkomma hela tiden. Om man koncentrerar sig på att det ska vara lätt att införa nya uppgifter så kommer söktiderna att påverkas negativt. Att söktiderna påverkas negativt är oacceptabelt, sökningar förekommer ju mycket frekvent i databaser.

Kontentan blir alltså att man får göra en avvägning mellan svårigheterna att införa nya data och möjligheterna att söka snabbt i strukturen. Denna avvägning är beroende av vad databasen skall användas till. I framtiden kan man vänta sig att söktiderna minskar och därmed ställer mindre krav på att nya data skall lagras in "snyggt".

2.4 Radering

För radering gäller i princip detsamma sak som för inlagring av nya data, men det tillkommer ett extra problem man kan nämligen radera på två olika sätt. För det första genom att verkligen ta bort elementet, för det andra genom att bara göra det oåtkomligt genom att ta bort hänvisningen till elementet. Den andra metoden är den som ställer till problem med dåligt minnesutnyttjande. Skulle man vid dessa olyckliga typer av data strukturer försöka att komma ifrån denna dåliga form av radering kan man bli tvungen att kontrollera en stor del av de lagrade elementen.

2.5 Uppdatering

Likheten med sökning och radering är uppenbar men ibland kan det bli problem eftersom ett element kan vara lagrat på flera ställen och det därför kan bli nödvändigt att gå igenom nästan hela databasen för att finna ett önskat element. Uppdatering kan vid vissa användningar vara mycket frekvent. När en uppdatering sker är det ofta mycket stora datamängder som skall behandlas.

2.6 Säkerhet

Hur stor är chansen att fel smyger sig in i databasen. Det är mycket svårt att avgöra men det är klart att valet av datastruktur är av betydelse. En datastruktur där ett och samma element lagras på flera ställen ökar risken för att något fel kommer in eftersom en stor mängd data ska registreras. En struktur där ett element bara lagras på en plats har däremot nackdelen att den är mer komplicerad och på grund härav ökar riskerna för fel.

En helt annan form av säkerhet är frågan om utomstående kan stjäla information ur databasen. Detta kan man förhindra t.ex. genom att göra vissa delar av databasen tillgänglig endast för den som har kännedom om ett visst lösenord. Fortsättningsvis kommer endast säkerheten att beröras då den kan påverkas genom olika struktureringar i databaserna.

Säkerhet kan även vara en fråga om backup, d.v.s. att spara en kopia av det arbete man har utfört, dels under det att man arbetar i systemet och dels långtidsförvaring. Detta är inget problem och är väl löst i alla systemen.

3. DE TRE OLIKA DATASTRUKTURERNA

3.1 Hierarkisk databas

Den hierarkiska databasen är uppbyggd som ett träd som succesivt förgrenar sig mer och mer. Detta är en fix databas som kan bli oändligt stor och inte innehåller några tomma utrymmen. (Naturligtvis finns det programmeringstekniska undantag från detta.) Så länge man finner rätt väg genom hierarkien så får man i denna struktur mycket korta söktider. Men när man blir tvungen att söka på ett annat sätt än vad datastrukturen är avsedd för får man problem och söktiderna ökar avsevärt. När denna databas används blir man ofta tvungen att lagra ett och samma element på många olika ställen, vilket ju medför ett dåligt minnesutnyttjande. Att lägga till nya data eller ta bort gamla är enkelt, eftersom man bara kan nå ett data från en punkt. Uppdateringar är en svaghet hos denna struktur eftersom ett element ofta kan lagras på flera ställen.

3.2 Relationsdatabas

Denna struktur beskrivs enklast med hjälp av en tabell. Denna tabell har ett bestämt antal kolonner och ett obestämt antal rader. Varje rad är ett element och kolonnerna är elementets attribut. Sökningar i denna struktur går fort om radantalet är litet. Är radantalet stort blir söktiderna långa, det bör man vara medveten om då man använder denna struktur, det behöver dock inte innebära att man skulle fått kortare söktider med en annan strukturering. Minnesutnyttjandet är bra eftersom det inte finns några tomma utrymmen i tabellerna. Att sätta in nya data och radera gamla är tillsammans med att hantera komplexa sökningar denna strukturs stora styrka, eftersom det bara är att ta bort eller sätta dit nya rader utan någon hänsyn till att de ska vara sorterade. Uppdateringar är enkla att utföra eftersom strukturen är simpel och ett element bara lagras på ett ställe.

Detta är en struktur som i dagens läge inte användes så mycket eftersom den ställer stora krav på söktider, men framtiden ser annorlunda ut. Ty när söktiderna minskar p.g.a. bättre hård och mjukvara är detta en struktur som har framtiden för sig eftersom den är väldigt användarvänlig, man kan ju stoppa in och plocka ut element nästan hur som helst.

3.3 Nätverksdatabas

Detta är en konstruktion som lätt blir mycket komplicerad. Nätverksdatabasen bygger på ringformade listor som sätts ihop av pekare. Nätverksstrukturen fås genom att flera pekare pekar på samma element.

Minnesutnyttjandet är bra eftersom det inte sker någon dubbel bokföring. Dock kan vissa problem tillståta vid radering, se nedan, som kan göra att utnyttjandet i vissa extrema fall blir helt oacceptabelt. Nya data är lätta att sätta in eftersom pekare är speciellt lämpade för sådant arbete.

Radering kan ställa till problem. Ett godtyckligt element i databasen har en pekare som pekar på ett annat element och har ett okänt antal element som pekar på sig självt, just i det sistnämnda ligger problemen. När man vill ta bort detta element så vet man ju inte vilka element som pekar på det element man vill avlägsna. Man känner alltså inte till om de övriga elementen "godkänner" att man tar bort det element som man i ett annat sammanhang tycker är skräp. För att man då ska kunna ta bort ett element får man då undersöka samtliga andra element ifall något skulle ha en pekare mot det aktuella elementet, detta är mycket tidsödande. Det finns ett alternativ till detta förfarande, nämligen att bara ta bort pekaren till det element man vill avlägsna, men det medför att man fyller databasen med information som är både inaktuell och oåtkomlig.

Sökning i denna struktur kan jämföras med sökning i en vanlig hierarkisk databas, d.v.s. sökningen är mycket snabb om man kan söka så att man följer inlagringsstrukturen och om data som hör ihop ligger lagrade på ett och samma ställe i datorns minne. Att detta är fallet är inte alls helt självklart. Uppdatering brukar gå bra eftersom varje element bara lagras på ett ställe. (Se /PAG83/)

4. LAGERINDELNING OCH OBJEKTSINDELNING

4.1 Sammanfattning

GDS använder objektsindelning. Någon möjlighet att använda lagerindelning i GDS finns inte. Att det inte finns någon möjlighet att använda lagerindelning i GDS beror på vissa små skillnader vid modelleringen och förklaras längre fram. Det finns dock ytterligare en indelningsform i GDS som kallas för grupper. Grupperna kan betraktas som en mycket förenklad form av lager.

Medusa använder lagerindelning. Dessutom har man en indelningsform med klumpar. Indelningen i klumpar har stora likheter med GDS's objektsindelning.

Intergraph använder också lagerindelning. Lagren i Intergraph användes på samma sätt som i Medusa och enheterna i Intergraph är mycket lika klumparna i Medusa.

4.2 Exempel

För att belysa skillnaderna mellan att arbeta med lager eller objekt kommer här några exempel:

- Du skall rita ett hus. I GDS definierar du objekt och gör varje objekt till ett rum dessa objekt sätter du sedan samman till en grupp som representerar en våning. I Medusa och Intergraph tar du ett antal lager och låter dessa representera en våning. Då du sedan ritat på denna våning så placerar du olika element i skilda lager.
- Ritar du i lager är det betydligt enklare att massberäkna, eftersom det är naturligt att sortera materialet i lager. Vill du däremot veta vad som skall finnas i vart rum torde objekten vara mer lätthanterliga.
- När man arbetar på ett stort projekt får man med lagerindelningen på ett enkelt sätt ritandet systematiserat. Med objekt kan den oförsiktige betydligt lättare göra det hela ganska rörigt. Exempelvis om man har ett stort hus, kanske 5000 rum numrerade 1-5000 och man vill hitta rummet öster om den sydvästra hissen på 17 vån.

5. ALLMÄNT OM RITSYSTEMEN

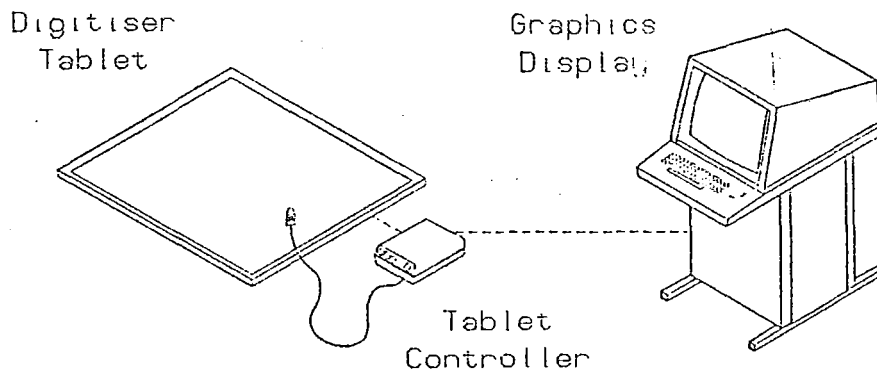
5.1 GDS

GDS, General Drafting System, är ett generellt ritsystem utvecklat i Cambridge, England av ARC (Applied Research of Cambridge). Det här studerade CAD-systemet lanserades 1980, före det hade man ett system kallat Building Design System, BDS. GDS står för General Drafting System och är utvecklat främst för byggnads och markentreprenörer. Dock är systemet generellt vilket innebär att det utnyttjas inom följande områden: mekanik, vvs, el, rör, arkitektur, väg och vatten och kartografi.

GDS är ett interaktivt system vilket innebär att man sitter framför en skärm och ser hur ritningen växer fram allt efter som indata ges. Kommandona kan ges antingen genom en alpha-numerisk skärm eller via menyn framför den grafiska skärmen. Högst en vy kan ses åt gången på skärmen.

I GDS bygger man upp ritningen med hjälp av ett antal objekt. Först definierar man ett objekt, sedan ritar man på det tills det är färdigt, därefter definierar man ett nytt objekt. Det gäller att systematiskt dela upp ritningen i många objekt och lika systematiskt döpa dessa. När man i efterhand ska ändra i ritningen så plockar man fram det objekt som ska redigeras med hjälp av dess "namn". Vid en utskrift har man möjlighet att bestämma vilka objekt som skall vara med på ritningen. Objekten i sin tur delas sedan in i grupper. Det finns ingen möjlighet att relatera grupper till varandra annat än genom dess placering i ritarealen.

I GDS finns det inte någon möjlighet att bygga upp en tredimensionell modell på det sättet som man kan i de andra systemen. Här består den tredje dimensionen helt enkelt av en z-koordinat, med vars hjälp man kan göra tredimensionella bilder, någon riktig tredimensionell volymmodell byggs inte upp i systemet. Det finns möjlighet att arbeta i färg. (Se/ARC82c/)



Arbetsstation för GDS.(/ARC82b/)

5.2 Medusa

Medusa är ett generellt ritsystem och 3D modelleringsystem utvecklat i Cambridge, England av CIS. CIS står för Cambridge Interactiv Systems. CAD-systemet lanserades 1980. Systemet är inte utvecklat för någon speciell branch, det användes inom elektronik, mekanik, varvsindustri, vvs, el, rör, arkitektur, väg och vatten och kartografi.

Indata och kommandon tilldelas på samma sätt som i GDS. Dock har man i Medusa möjlighet att visa valfritt antal vyer samtidigt på skärmen.

I Medusa är ritblanketten indelad i 1024 lager, några av lagren är reserverade för systemet men bland de övriga kan man själv välja i vilket lager man vill lägga vilken information. Man kan under ritandet bestämma vilka lager man vill se genom att "tända" eller "släcka" dessa. Samma möjligheter ges vid utskrift där man kan bestämma vilka lager man vill ha med på en speciell ritning.

I Medusa kan man bygga upp en riktig tredimensionell volymsmodell och naturligtvis arbeta i färg. Användaren kan själv göra fortranprogram för att snabbare klara av rutinarbeten. (Se /CIS82/)

5.3 Intergraph

Intergraph är ett interaktivt system som marknadsföres av Intergraph Corporation. Systemet är Amerikanskt och kommer från Huntsville, Alabama. Intergraph Corporation började marknadsföra sitt CAD-system 1973.

Intergraph används inom följande områden: elektronik, mekanik, varvsindustri, vvs, el, rör, arkitektur, väg och vatten och kartografi.

Även i Intergraph kan man bygga upp en tredimensionell ytmodell och arbeta i färg. Man har två stycken grafiska bildskärmar till skillnad från de andra systemen där man bara har en. På varje skärm kan man samtidigt se upp till 4 vyer d.v.s. man kan samtidigt ha upp till 8 bilder framför sig.

När man bygger upp en ritning i Intergraph görs det på samma sätt som i Medusa, men här finns det 2048 lager. (Se /INT82a/)

6. HJÄLPDATABASERNA

6.1 Allmänt

GDS, Medusa och Intergraph har alla i grunden ett liknande system med tre olika typer av databaser. Ritning och modellering utföres först och främst i projektdatabasen, men under arbetet med projektdatabasen gör man hänvisningar till hjälpdatabaserna. Hjälpdatabaserna är ganska lika varandra i de tre systemen. De skillnader som förekommer kommer att beröras på följande sidor.

Det finns även programmoduler och databaser som innehåller exempelvis kommandoprocedurer, interface till beräkningsrutiner, formatsatser för utskrifter o.s.v. Ett exempel är menyerna. Denna del av datastrukturen berörs inte vidare.

Symbolbibliotek och egenskapsbeskrivningar i de tre systemen lagras på olika vis och kan sedan användas vid uppbyggnad av projektdatabaser.

6.2 En katalog (Exempel ur Intergraph)

Uppbyggnaden består i grunden av många små objekt, som förses med attribut. (Ett attribut är det samma som en beskrivning eller bestämning av objektet.) Katalogerna i allmänhet är tillräckligt stora för alla upptänkliga användningsområden. En katalog kan innehålla över miljonen objekt, då detta är otillräckligt kan man räkna med att det inte är katalogen som sätter en gräns utan det är andra delar i CAD-systemet som är begränsande.

ETT OBJEKT OCH DESS ATTRIBUT

```
SKAP
-----
! TYP      ! AR      ! MÄRKE  ! MODELL ! X-REF  !
-----
                ↑
                Attribut
```

ETT OBJEKT OCH DESS ATTRIBUTS VÄRDEN

SKÅP

!	HÖG	!	1982	!	MARBOD.	!	H2300	!	X	!
---	-----	---	------	---	---------	---	-------	---	---	---

↑
Attributets
värde

När man sedan sätter ihop många objekt så måste de struktureras så att de kan hittas med korta söktider. Därför indelas de i huvudgrupper t. ex. SKÅP, DÖRRAR, HYLLOR, INREDNING o.s.v. Detta kan också betraktas som att man delar in katalogen i flera små kataloger. Sedan måste sökning i varje huvudgrupp göras effektivt det görs genom att man tilldelar varje objekt en identifierare. Denna identifierare ges ett valfritt alfanumeriskt värde, enda kravet är att värdet är unikt.

ETT OBJEKT MED IDENTIFIERARE

!	IDENTIFIERARE	!	TYP	!	ÅR	!	MÄRKE	!	MODELL	etc.
!		!		!		!		!		

Det behövs ytterligare en variabel, nämligen en möjlighet att koppla ihop flera objekt som på något sätt hör ihop. Detta behövs för att förenkla användandet och förkorta söktiderna. Ett bra sätt att gruppera ihop element är med pekare.

OBJEKT MED PEKARE

! IDENTIFIERARE	! PEKARE	! TYP	! AR	! MÄRKE	etc.
!	!	!	!	!	

Detta är alltså ett exempel på ett fullständigt objektet.
Det består av:

Identifierare
 Pekare
 Typ
 År
 Märke
 Modell
 X-ref.

ETT LITET UTDRAG UR KATALOGEN

SKAP

! IDENT.	! PEKARE	! TYP	! AR	! MÄRKE	! MODELL
----------	----------	-------	------	---------	----------

!
 !
 DÖRRAR !

! IDENT.	! PEKARE	! TYP	! MTRL.	! FÄRG	! GANGJ.
----------	----------	-------	---------	--------	----------

!
 !
 HYLLOR !

! IDENT.	! PEKARE	! MTRL.	! FÄRG	! DJUP	! FÄSTE
----------	----------	---------	--------	--------	---------

!
 !
 INREDNING !

! IDENT	! PEKARE	! TYP	! MTRL.	! LÅS
---------	----------	-------	---------	-------

Det är väsentligt att man vid lagring av data i katalogen gör lagringen så att element som hör ihop även kommer nära varandra i den "fysiska" databasen. Annars ökar söktiderna.

7. PROJEKTDATABASERNA

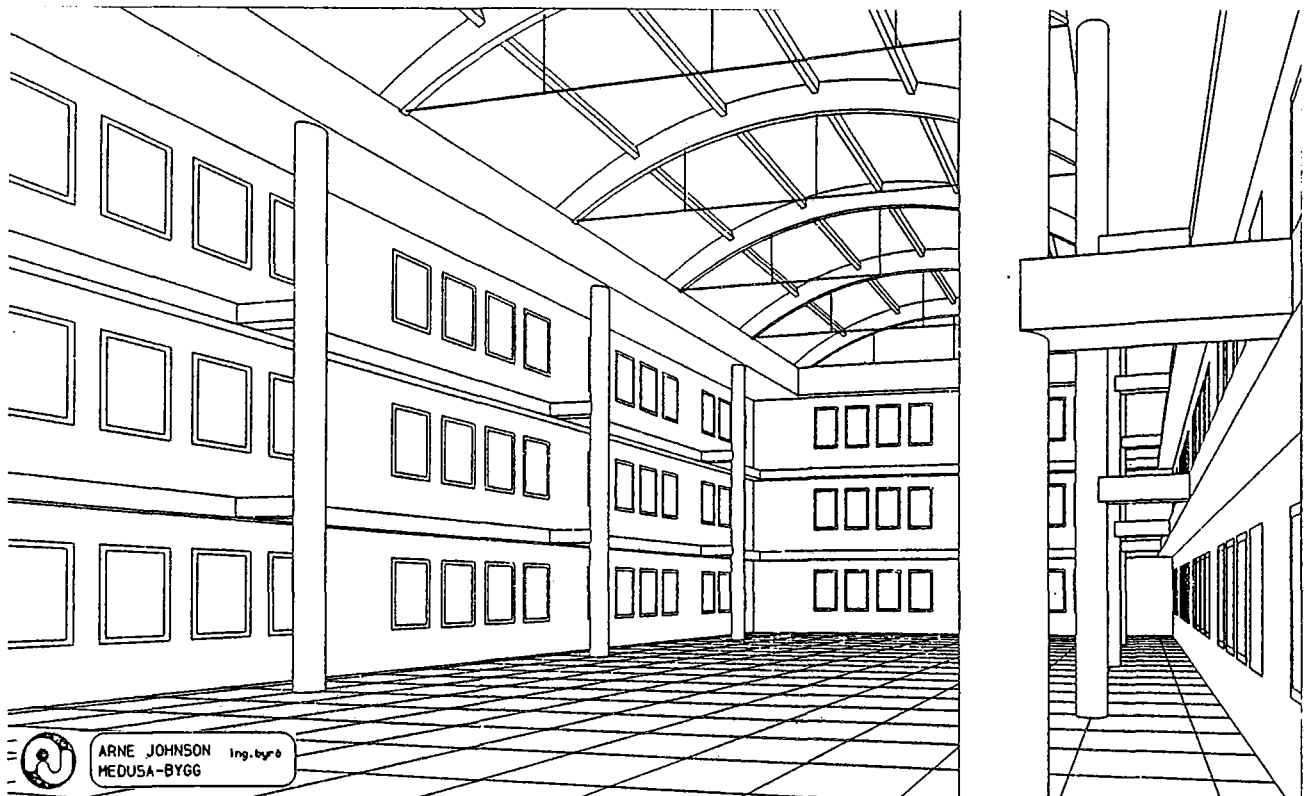
7.1 GDS

Det finns en samlingsfil för varje större oberoende arbetsobjekt. Samlingsfilen innehåller alla konstruktionsdokument, som berör objektet.

Exempel: -Sakregister
 -Listor över konstruktionsdokument
 -Ritningar och grafiska konstruktioner

Konstruktionsdokumenten förs som en nätverksdatabas med en prioritet genom hierarkiska strukturer. Man bygger upp hierarkiska strukturer och nätverket uppstår eftersom man kan göra pekare till katalogen och andra konstruktionsdokument. För strukturering i databasen använder man objektsindelning och grupper.

7.2 Medusa



Exempel på tredimensionell ritning gjord i Medusa.(/JON84/)

I Medusa har man en grafisk databas. Det kan bara finnas en ritningsmodell åt gången i basen. Det finns möjlighet att göra en hierarkisk modelluppbyggnad med ett antal grundläggande elementtyper.

För att göra arbetet i databasen enklare är den indelad i 1024 lager.

7.3 Intergraph

Även detta är en hierarkisk databas som har hänvisningsmöjligheter precis som de andra. Det som skiljer Intergraph från de andra systemen är att grafiken ligger i en separat databas. Denna grafikdatabas är indelad i lager på samma sätt som i Medusa men här är lagerantalet 2048. I projektdatabasen (den icke grafiska) har man en indelning i enheter.

7.4 Sammanfattning

Databaserna i de tre systemen har stora likheter med varandra. Objekten i GDS, klumparna i Medusa och enheterna i Intergraph är mycket lika varandra. Skillnaderna mellan objekten, klumparna och enheterna uppstår mest vid själva modelleringen. En väsentlig skillnad finns dock, i Medusa har man möjlighet att låta en klump innehålla en annan klump, i Intergraph har man möjlighet att låta en enhet innehålla en annan enhet, men i GDS finns det ingen möjlighet att låta ett objekt innehålla ett annat objekt.

Har man inte möjligheten att låta ett element innehålla ett annat så har man inte heller någon möjlighet att strukturera elementen hierarkiskt, detta medför även att man inte kan göra någon indelning i nivåer.

8. MODELLERING I GDS

8.1 Allmänt

GDS bygger på objektsindelning, d.v.s. man definierar ett objekt och ger det ett namn. Därefter "ritar" man på detta objekt, när man är färdig så definierar man ett nytt objekt och ritar på det tills man är färdig o.s.v.

Under ritandet kan man hänvisa till objektkatalogen och andra konstruktionsdokument. Dessa hänvisningar är enbart hänvisningar, d.v.s. det sker ingen kopiering så att en uppgift lagras på flera ställen.

Antalet objekt som kan definieras kan i praktiken betraktas som obegränsat.

8.2 Hierarkien

GDS ritareal är detsamma som en fil. Denna fil innehåller grupper och objekt:

8.3 Grupper

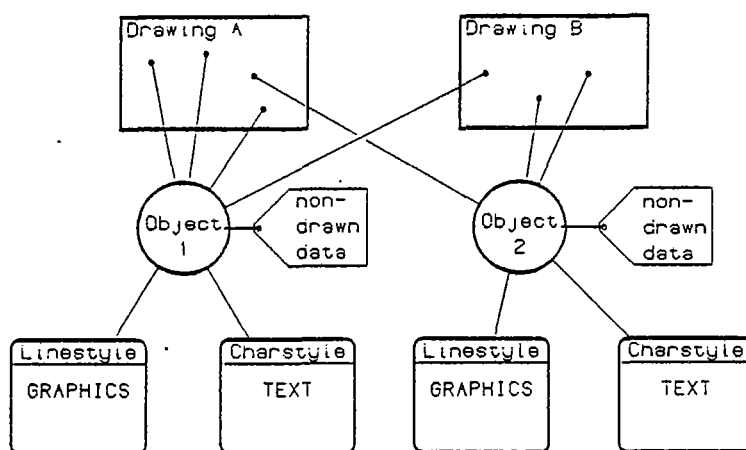
En fil kan innehålla många grupper som var och en har ett namn. Varje grupp kan betraktas som ett stort papper, på vilket man placerar objekt i skala med hjälp av referenser. En referens består av namn, position, skala och rotation av ett objekt.

Tillsammans med varje grupp förs ett register över vilka utskrifter som gjorts från densamma.

8.4 Objekt

Linjer och tecken i grupperna är sammansatta till enheter som kallas objekt. Objekten kan manipuleras som enheter, och kan kopieras många gånger i olika skalor och positioner. Ett objekt kan användas i flera grupper, och kopieras från en ritareal till en annan.

Varje objekt har ett unikt namn som definieras av användaren. Till detta namn lägger systemet ett versionsnummer. Varje objekt består av grafiska block som definierar punkter, linjer, kurvor, cirklar samt text block som innehåller text och kommentarer. Det rekommenderas att man i ett objekt har mellan 5 och 100 grafik eller textblock. Text och grafikblock är det enda man kan ha i ett objekt, förutom sådana attribut som utgör hänvisningar till katalogen. Det finns inte någon möjlighet att låta ett objekt tillhöra ett annat objekt.



(/ARC82b/)

När du definierar ett nytt objekt så definieras även dess fysiska placering i ritarealen, därefter gör du en referens i den grupp där du vill ha objektet, på så sätt placerar du objekt på den plats du önskar. På detta sätt får du naturligtvis möjlighet att göra många referenser till objektet. Gör du därefter en ändring av objektets attribut får du en ny variant av objektet, denna nya variant får även ett nytt namn. Tar du bort referenserna till ett objekt så att det blir helt utan referenser så försvinner det automatiskt. Det finns möjlighet att göra objekt som har preferens över alla andra varianter av samma objekt.

Objekt som användes ofta lägger man i en extern objekt fil, som ett bibliotek. Sedan kan man när man vill använda detta objekt kopiera in det från objektbiblioteket. Detta ger också en möjlighet att enkelt tillverka snarlika objekt, genom att kopiera in ett objekt och därefter ändra i det.

8.5 Grafiska block.

Ett grafiskt block definierar det grafiska innehållet i ett objekt, i termer av punkter, linjer, kurvor och cirklar. Varje block representerar en sträng av punkter, normalt sett sammanfogade av raka linjer eller kurvor.

Varje grafiskt block har en definition av hur linjerna skall se ut, tjocklek, form och utseende.

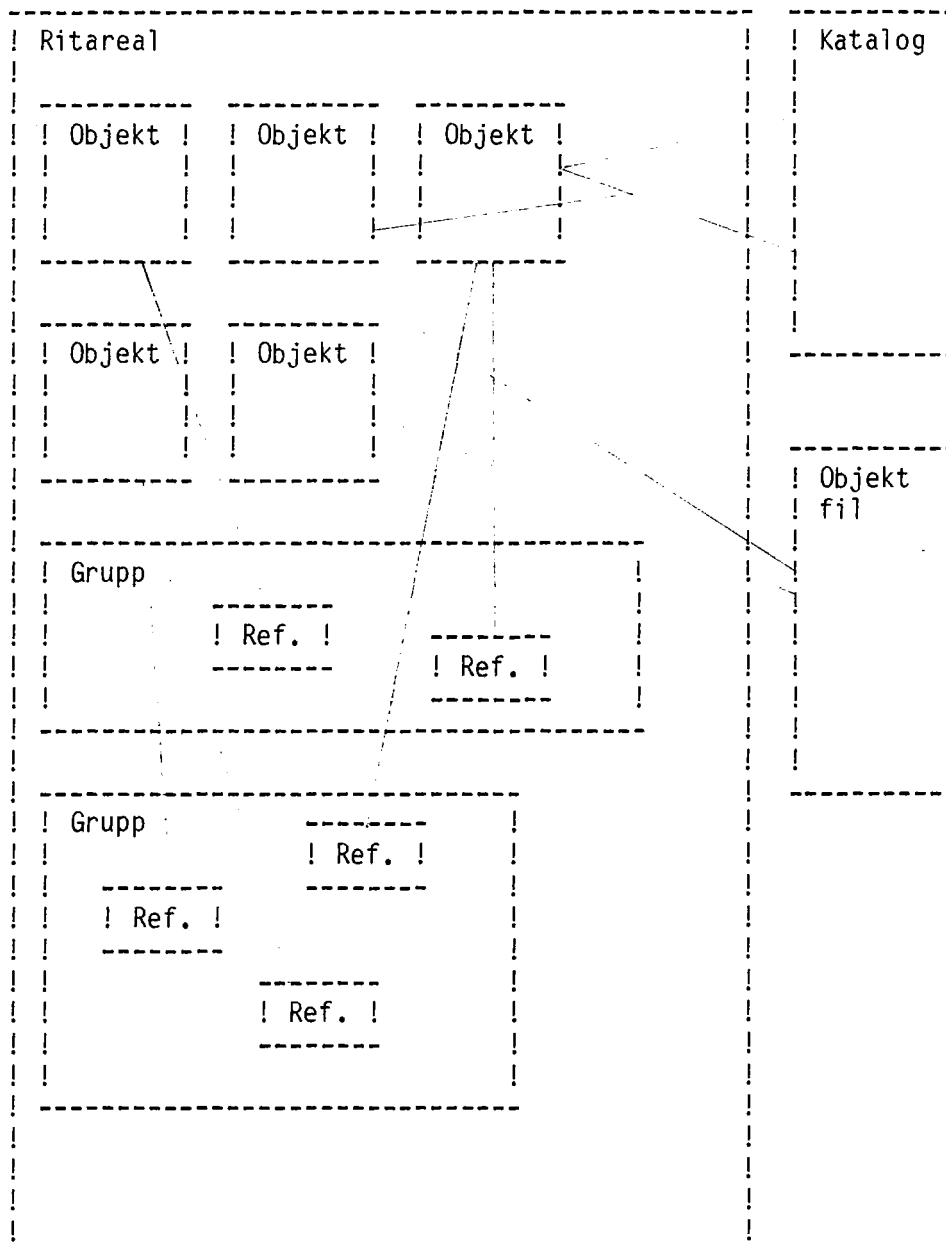
Grafiska block har inga namn, de identifieras endast genom sin placering.

8.6 Text block

Text blocken innehåller de texter som är associerade till objektet. Varje block representerar en eller flera rader med ett eller flera tecken. Dessutom innehåller blocket textens storlek och rotation.

Alla textblock är associerade med en teckenstil, som definierar tjocklek och teckentyp på alla i ritarealen ingående texter. Teckenstilen kan naturligtvis senare ändras i varje enskilt textblock om så önskas.

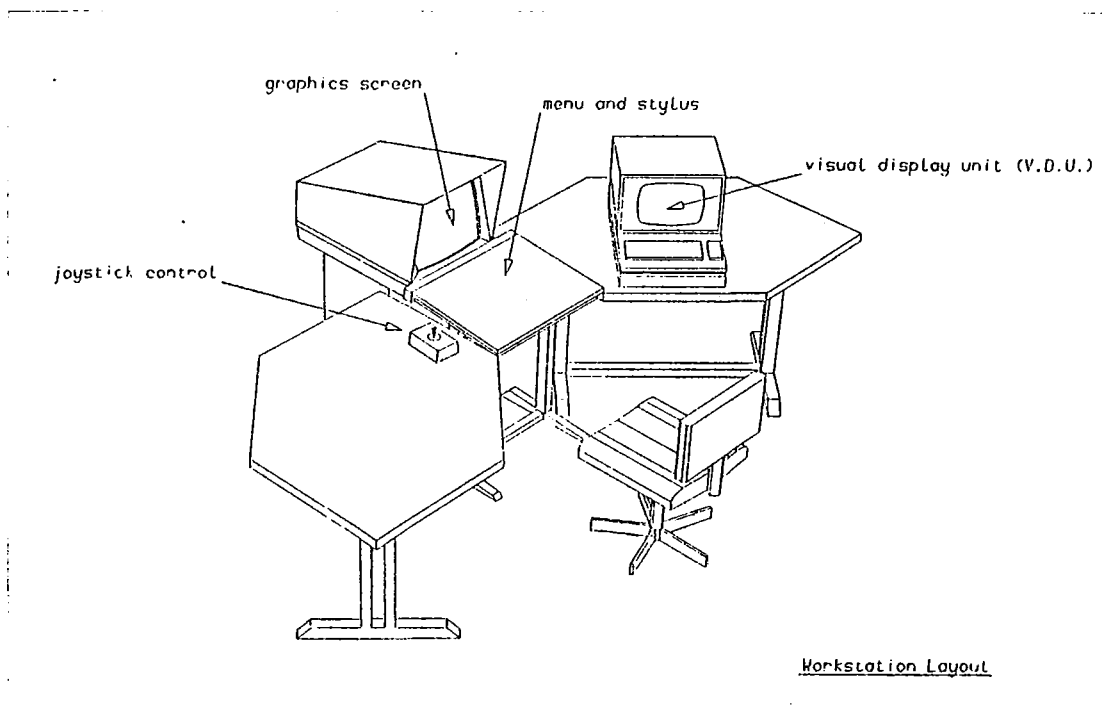
Exempel :



9. MODELLERING I MEDUSA

9.1 Allmänt

I Medusa är ritningsblanketten indelad i lager. Element kan placeras på olika lager som kan vara tända eller släckta. Denna lagerindelning passar bra för att t.ex lägga dörrar i ett lager, väggar i ett annat och armering i ett tredje. Lagerindelningen består av 1024 lager. 3D volym- och ytmodeller kan skapas.



Arbetsstation för Medusa. (/CIS82/)

9.2 Hierarkien

En datafil i Medusa består av ett antal element. Dessa element relateras till varandra under modelleringen genom en hierarkisk datastruktur till en modell av verkligheten.

Denna hierarkiska strukturen är användbar speciellt i samband med uppdatering och stycklistgenerering.

9.3 De lägre elementklasserna

9.3.1 Linjer

En linje består av ett antal linjesegment vilka kan vara raka linjer, kurvor eller cirklar. Man kan också välja olika typer av stil på linjen. Ett linjesegment kan göras osynligt.

9.3.2 Texter

En text består av en teckensträng innehållande högst 80 tecken. Det finns möjlighet att välja både stil och storlek på texten.

9.3.3 Primmar

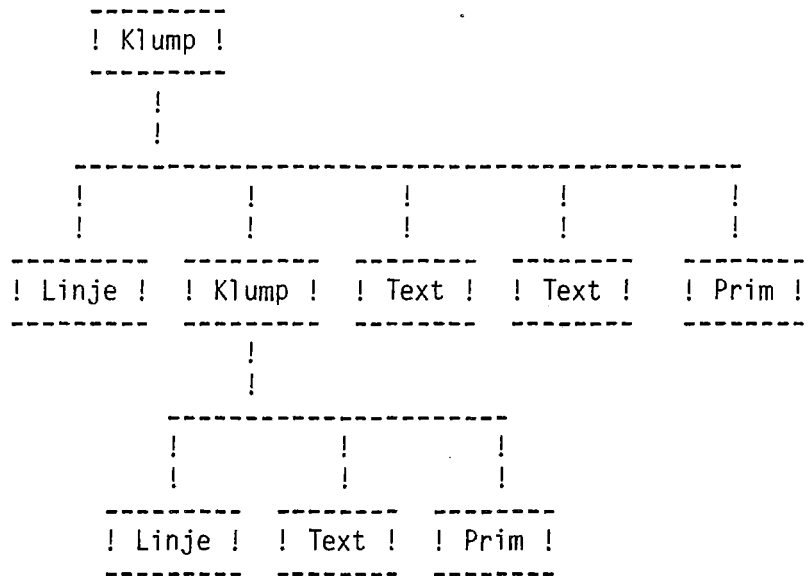
En prim är ett eller flera linjeelement vilka definierar en ofta använd figur. Primmen innehåller även en referenspunkt som används för lägesbestämning. En katalog med primmar upprättas separat från ritningsblanketten.

9.4 De högre elementklasserna

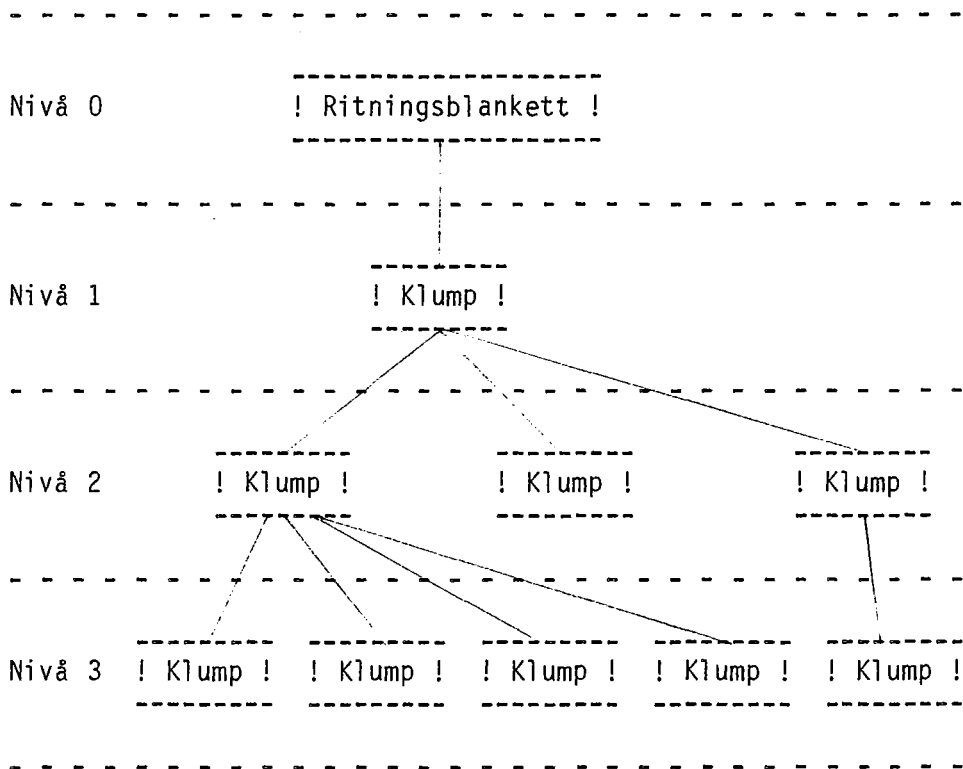
9.4.1 Klumpar

En klump är en samling av en eller flera linjer, texter och primmar, och möjligtvis andra klumpar. Samlingar av sådana element kan man referera till som en enkel enhet, exempelvis i samband med uppdatering och stycklistgenerering. Klumparna har inga namn men är av en viss typ. Till en klump relaterar man ett textelement, när man sedan söker efter den aktuella klumpen så kan man söka efter en klump som har ett visst textelement knutet till sig och på så sätt hitta den klump man vill ha tag på. En klump kan innehålla andra klumpar i upp till tre nivåer. Antalet klumpar som kan finnas i en ritningsblankett är obegränsat.

Exempel :



Exempel :

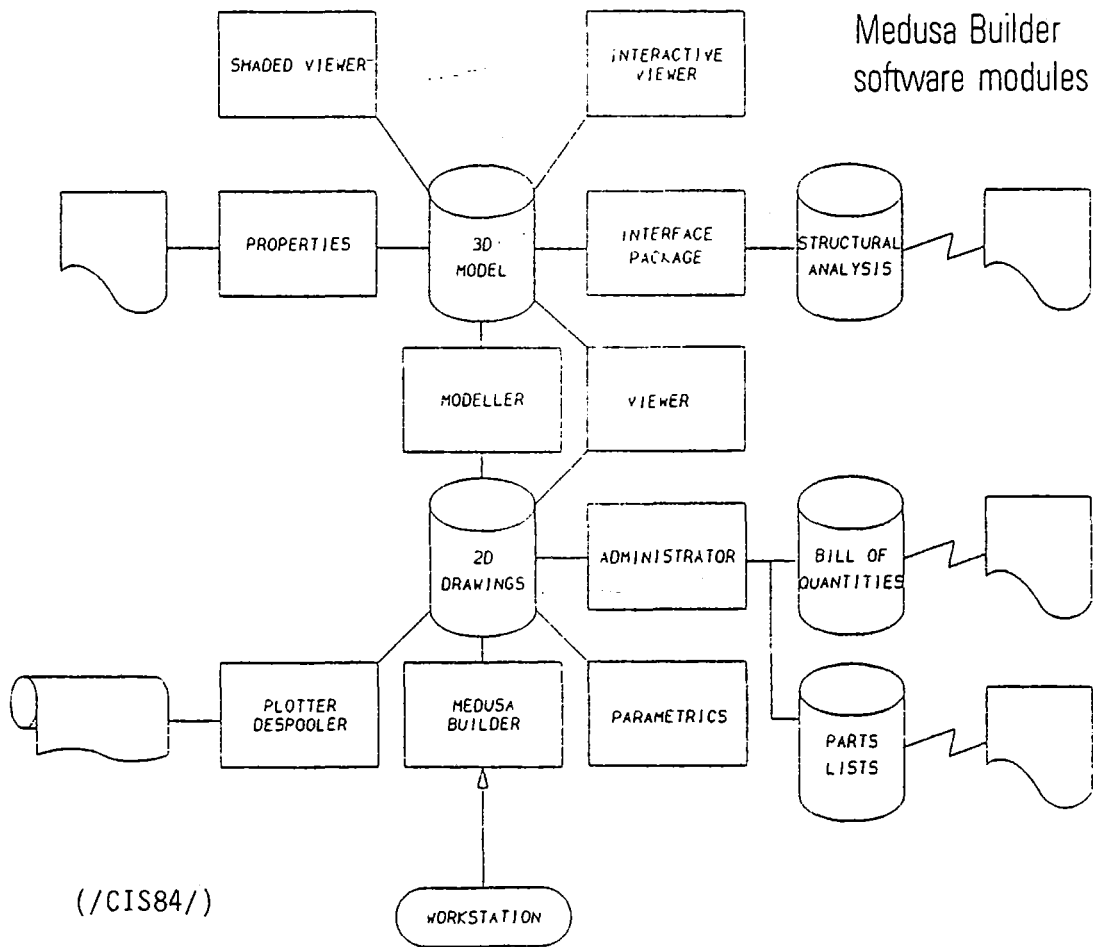


9.4.2 Ritningsblanketter

Ritningsblanketten är den högsta nivån av element, den kan vara indelad i lager och det får bara finnas en ritningsblankett i varje datafil. Ritningsblanketten innehåller alla andra element.

9.4.3 Symboler

Speciella klumpar som hör direkt till ritningsblanketten refererar man ofta till som symboler. De kan t.ex. representera sådana komponenter som är grundläggande och användes ofta. Generellt är symbolerna lagrade på filer en för varje symbol och kan läggas in i huvuddatafilen varje gång de behövs. Gruppen av filer som innehåller dessa symboler kallas för symbolbiblioteket.

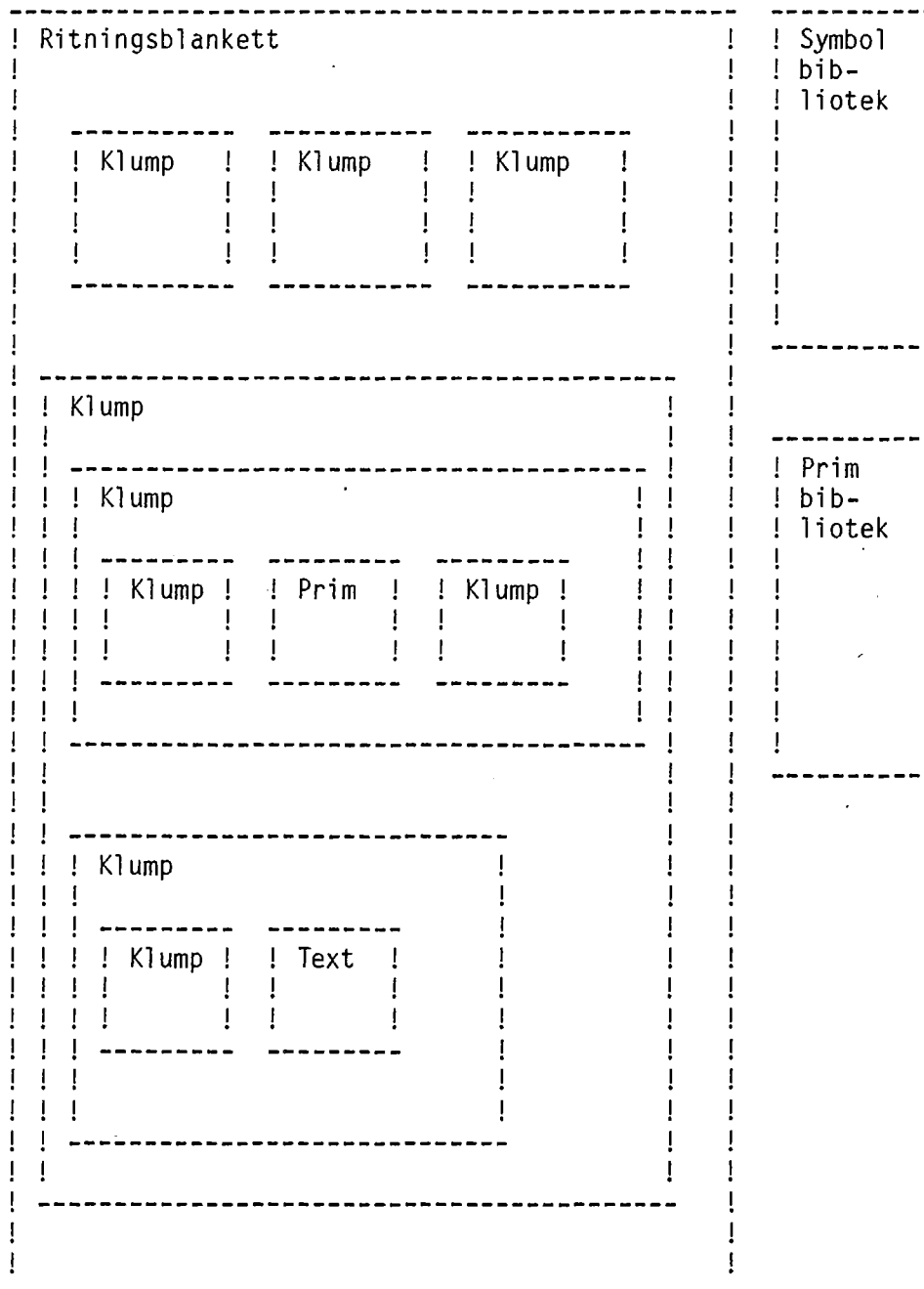


9.5 Avsökning av datastrukturen

Elementen är ordnade sekvensiellt i datafilen. Hela tiden under det att ritsystemet är i arbete finns det en pekare på ett element i filen. Detta element kallas för det aktuella elementet. Om det aktuella elementet är en klump så är alla elementen i klumpen att betrakta som aktuella.

Det är möjligt att göra en avsökning av en datafil i sekvensiell ordning på jakt efter ett element av en speciell klass, typ eller position. Det är möjligt att flytta till nästa element, eller till ett element av en speciell typ o.s.v.

Exempel:



10. MODELLERING I INTERGRAPH

10.1 Allmänt

I Intergraph har man två databaser en s.k. egenskapsdatabas och en grafikdatabas. Grafikdatabasen är indelad i lager på samma sätt som projektdatabasen i Medusa. När det gäller hänvisningar till egenskapsdatabasen så arbetar man enligt modellen med kopiering, d.v.s. man lagrar samma element på många ställen.

I Intergraph bestämmer användaren själv hur mycket minnesutrymme som skall reserveras till olika enheter, attribut o.s.v.

10.2 Egenskapsdatabasen

Egenskapsdatabasen i Intergraph baseras på en nätverksdatamodell med en hierarkisk datamodellsgrund. Databasen är i grunden en rak-fil struktur. Denna filstruktur görs sedan till en hierarkisk modell med hjälp av pekare. Antalet hierarkiska modeller du kan ha i din databas är obegränsat.

I egenskapsdatabasen finns det enheter dessa enheter kan jämföras med Medusas klumpar. Enheterna relateras till varandra i en hierarki genom att man för varje enhet anger vilken enhet som är dess förälder. Varje enhet består av en mängd attribut som beskriver enhetens egenskaper. Enheterna och dess attribut måste tilldelas nummer, man kan även tilldela dem namn om man så önskar.

Att man kan ange en enhet som förälder till en annan enhet innebär att man kan göra en indelning i nivåer. Det finns ingen begränsning av antalet nivåer man kan dela in egenskapsdatabasen i.

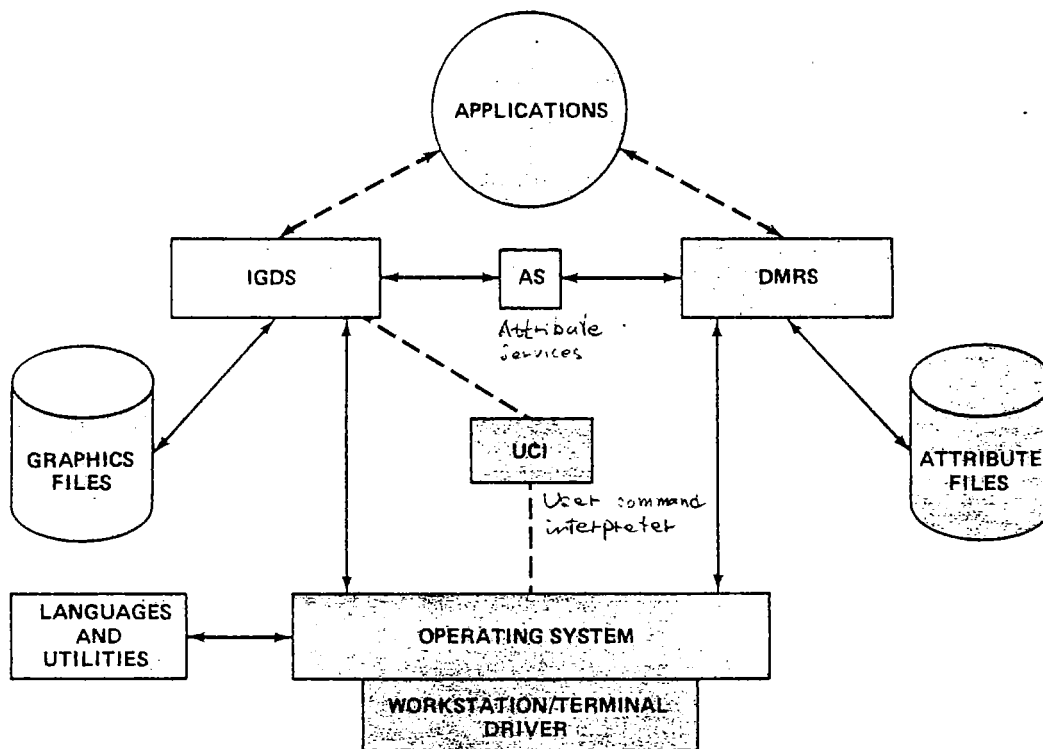
10.3 Grafikdatabasen

I Intergraph är grafiken helt skild från den "textmässiga" beskrivningen. Det finns alltså en databas där man lagrar "text", som är den egentliga projektdatabasen, denna databas kan du använda från en vanlig dataterminal som vilken databas som helst. När du sedan vill göra Intergraph till ett CAD-system så kopplar du till grafikdatabasen. I grafikdatabasen ritar du på precis samma sätt som i något av de andra systemen, sedan kopplar du figuren med hjälp av en pekare till egenskapsdatabasen.

10.4 Arbetsmetodik

Då man arbetar i Intergraph är det lite mer komplicerat än vad det är i de andra systemen. Det första man måste göra är att bestämma vilka enheter man ska använda och vilka attribut dessa enheter skall innehålla. Sedan ska du bestämma vad attributen skall vara av för typ och dess längd. Du ska även bestämma hur många enheter du ska ha på varje nivå och bestämma föräldrar till alla enheter.

Du ska alltså som användare inledningsvis bestämma hela strukturen på egenskapsdatabasen, vilket är ganska komplicerat. Det medför i sin tur att du oftast senare blir tvungen att göra ändringar i strukturen, detta är inte heller lätt att göra och blir ofta tidsödande eftersom syntaxen är krånglig.

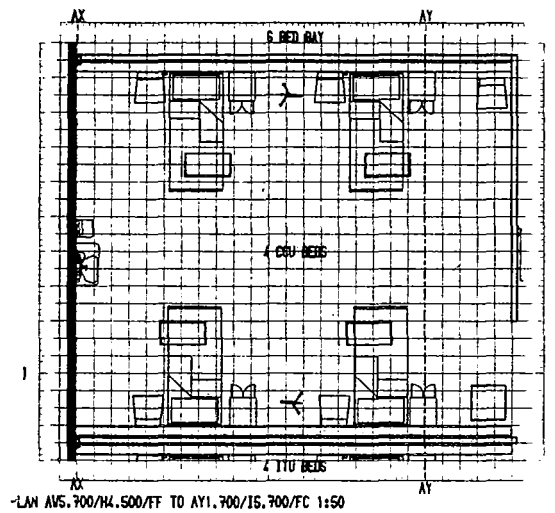


(/BET82/)

DETAILED INTERGRAPH SOFTWARE STRUCTURE

10.5 Sammanfattning

Intergraphs databas består av två huvuddelar egenskapsdatabasen och grafikdatabasen. Denna uppdelning medför att man får en databas som inte innehåller någon grafik, d.v.s. man får en databas som har ett betydligt vidare användningsområde. Intergraph har en osmidig och klumpig syntax, det är definitivt inte bra, men det kompenseras ganska väl genom att det är mycket lätt att skriva egna program till systemet i Fortran.



Exempel på ritning utförd med CAD-system.(/ARC82c/)

11. SKISSNING OCH ÄNDRINGAR I DETALJERINGSGRAD

11.1 Varför skissa i CAD-system?

Innan man påbörjar själva konstruktionsarbetet är det ofta önskvärt att kunna skissa med grova linjer i tre dimensioner. (Tvådimensionell skissning är sällan intressant om man har möjlighet till tredimensionell skissning, eftersom tvådimensionell skissning normalt sett medför mer arbete.) Den tredimensionella skissningen skulle medföra arbetsbesparingar eftersom man i ett CAD-system lätt kan lägga till en extra våning, ändra en taklutning, eller vrida huset i terrängen. Just det sistnämnda är något som kan vara mycket attraktivt. Det finns ju inte någon större möjlighet att flytta runt ett hus i en terrängmodell utan hjälp av CAD-system eller något annat motsvarande datorprogram.

11.2 Skissning i terrängmodell

Önskar man göra en skissning av ett hus i ett CAD-system så går det redan i dagens läge. Systemen är inte avsedda för skissning men med vissa åtgärder kommer de att fungera någorlunda.

Det är av intresse att kunna skissa ett hus i ett CAD-system men intresset är inte så stort, eftersom den som idag skissar hus på papper är väl förtrogen med detta förfarande och snabbt drar upp huvudlinjerna för ett hus. Det man däremot idag har svårigheter med och där man kan göra en klar tidsbesparing är om man kan skissa ett hus i en terrängmodell.

Problem dyker upp då man vill sätta in sitt hus i en terrängmodell. Det är rent principiellt inga problem att istället för byggnadsdelar lagra en markyta i databasen, men vill man lagra en stor yta med hög noggrannhet så fordras ett icke oansenligt minnesutrymme. Men om man inte slösar med minneskapaciteten så kan man även få detta att fungera. (Se /JON84/)

11.3 Ändringar i detaljeringsgrad

Nästa problem dyker upp då man vill göra ändringar i detaljeringsgrad, d.v.s. när man vill ändra sin grovt ritade figur till en figur med mycket detaljer.

Möjligheterna till ändringar i detaljeringsgrad skiljer sig en del mellan de tre olika systemen.

11.3.1 GDS

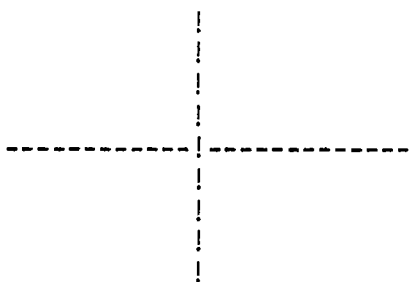
Vill man göra en ändring av detaljeringsgraden i GDS så kan man göra det genom ändringar i objektfilen.

Man kan t.ex. först definiera en vägg med ett enda attribut, väggens utsträckning som definieras som ett streck alternativt ett plan. När man sedan vill höja detaljeringsgraden så tilldelar man väggen nya attribut. Man kan t.ex. göra den tredimensionell och tala om vad den är gjord av. Vissa problem kan dock uppstå med denna metod.

Exempel:

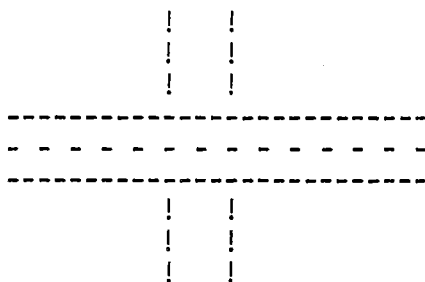
Man har definierat väggar som linjer. Nu önskar man göra dem tredimensionella.

LAG DETALJERINGSGRAD



Detta beskriver alltså ett hörn mellan fyra väggar.

HÖG DETALJERINGSGRAD



Detta är alltså samma figur men med höjd detaljeringsgrad. Här får man en yta, precis där dom två väggarna möts, på vilken det finns två väggar.

Detta var bara ett exempel på vad som kan inträffa. Naturligtvis är det inte omöjligt att lösa både detta eller andra problem, men svårigheterna kan ofta vara stora.

11.3.2 Medusa

Om man använder sig av primmar så finns det möjlighet att göra ändringar på samma sätt som i GDS, genom att göra ändringar i primbiblioteket. Om man inte använder sig av primmar så finns det ingen möjlighet att göra ändringar i katalogen och på så sätt få ändringar i ritningen eftersom man kopierar ur katalogen till ritningen.

Däremot har man lager och det kan man också arbeta utifrån. Man går då t.ex in och ändrar alla element i ett lager, eller alla element som är ritade med en viss linjetyp i ett visst lager. Denna metoden är betydligt långsammare än GDSs metod eftersom man måste ändra på många ställen. Har man dock en väl fungerande rutin för detta arbete så bör det fungera bra.

11.3.3 Intergraph

Intergraph kopierar från katalogen till projektdatabasen. Någon motsvarighet till Medusas primmar finns inte i Intergraph, så här är man hänvisad till lagerindelningen som arbetsmetod, d.v.s. samma metod som i Medusa.

11.4 Säkerhet

Ändringar i detaljeringsgrad kan inte betraktas som riskfritt, eftersom det är stor chans att man gör någon felaktig ändring eller "glömmer" att ändra någonstans.

Ändrar man enligt den metod som beskrivits för GDS är riskerna stora för fel eftersom man inte har någon som helst koll på att gjorda ändringar är korrekta. Det finns en möjlighet att öka säkerheten genom en tvåstegs ändring. I första steget gör man då en ändring som är mycket tydlig, exempelvis kan alla ändringar skrivas ut i rött, så att det går att titta igenom alla ritningarna och se att de är korrekta. Men det är inte säkert att detta är tillräckligt hög säkerhet i alla sammanhang.

Ändringar enligt den metod som beskrivits för Medusa är säkrare. I den rutin som gör ändringarna kan man lägga in många kontroller. Kontroller som t.ex. ser till att inte väggar blir dubbla. Här har man också möjlighet att låta datorn visa alla ändringar som görs. Att den visar alla ändringar som görs hindrar dock inte att oändrade felaktig-

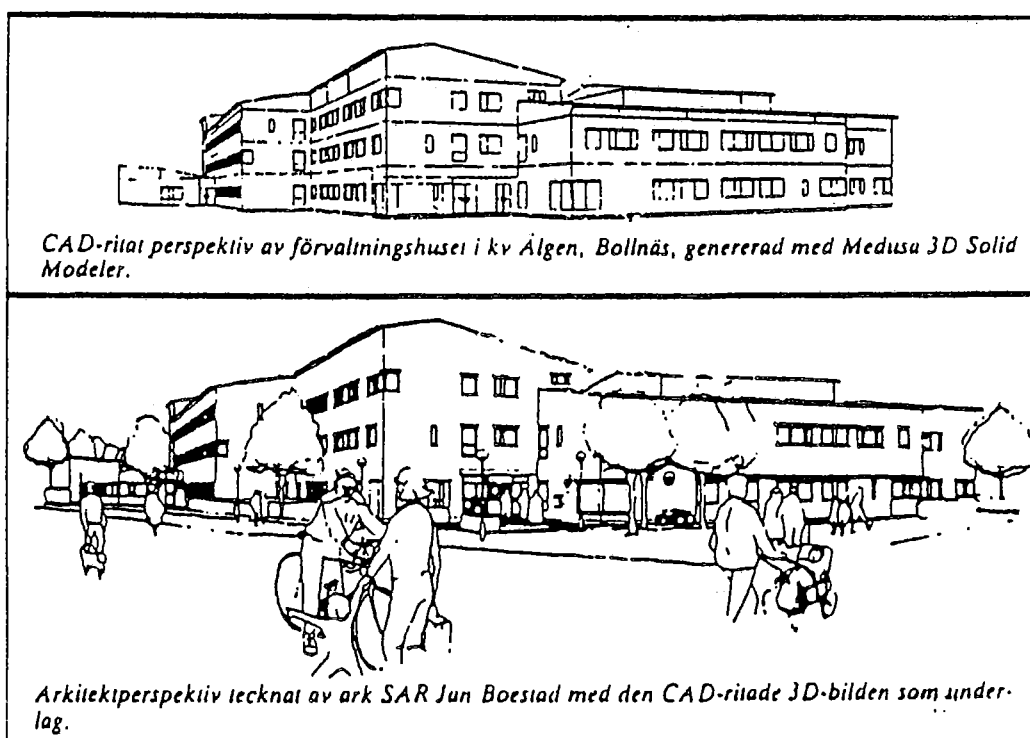
heter finns kvar på ritningen.

11.5 Sammanfattning

De två metoder (ändringar i objektfilen, sökning i lager) för ändring som här nämnts är inte felfria lösningar. Att säga att den ena lösningen är klart bättre än den andra är inte riktigt. Ett visst mått av säkerhet kan uppnås då man gör ändringar, men definitivt inte någon hög säkerhet.

11.6 Kommentar

En del av intresset för ändringar i detaljeringsgrad bottnar i en vilja att rita fort, detta kan man även uppnå på annat sätt t.ex genom dubbelritning, d.v.s. man kan istället för att rita ett streck i taget rita flera stycken på en gång. Denna metod kan användas i alla tre systemen och gör man det så behöver man inte göra ändringar i detaljeringsgraden och får därmed en betydligt säkrare arbetsmetodik.



(/BYG84/)

12. KONVERTERINGSFORMAT

12.1 Allmänt

När det gäller att överföra information mellan olika datorer, CAD-system, så är inte problemet att överföra en färdig ritning. Det är nämligen ganska enkelt och görs redan idag mellan flera existerande system. Ett färdigt standard konverteringsformat för denna typ av informationsöverföring finns redan.

Problemet är istället överföring av "intelligens", d.v.s. CAD-systemet till vilket man gjort överföringen ska "begripa" vad det har fått. Ett exempel: Gör man en överföring av en byggnad eller del av densamma så skall det mottagande systemet "begripa" att det har fått ett hus och inte tolka det som en bunke linjer.

När man ska dra upp riktlinjer för hur överföringen ska ske är det inte bara en fråga om vilket system som är bäst utan även en fråga om på vilket sätt man vanligtvis lagrar en viss typ av intelligens.

Här berörs bara några av de väsentligaste skillnaderna mellan systemen och görs en jämförelse med andra här inte nämnda system för att ge en inblick i hur ett framtida konverteringsformat kan se ut.

Man får inte glömma att det idag inte finns något övergripande format för överföring av en modell mellan flera olika system. (Det finns konverteringsformat som fungerar för överföring av vissa modelltyper mellan två system men inte mellan flera.) Målet är att överföring ska kunna ske mellan så många system som möjligt med så små förluster av intelligens som möjligt. Dessutom ska konverteringsformatet vara "hanterbart". (Se /BER84/)

12.2 Lagerindelning

GDS använder inte lagerindelning, Medusa och Intergraph använder lagerindelning. Det är just vid överföring mellan system som använder lager och system som inte använder lager som det dyker upp problem. Då man skall göra dylika överföringar kommer man antagligen att använda något slag av enlagersprincip, d.v.s. det som lagrats i ett system utan lagerindelning kommer att lagras i ett lager i ett system med lagerindelning. Det som lagrats i flera lager i ett system med lagerindelning kommer att lagras i ett lager i ett system utan lagerindelning.

12.3 Kopiering

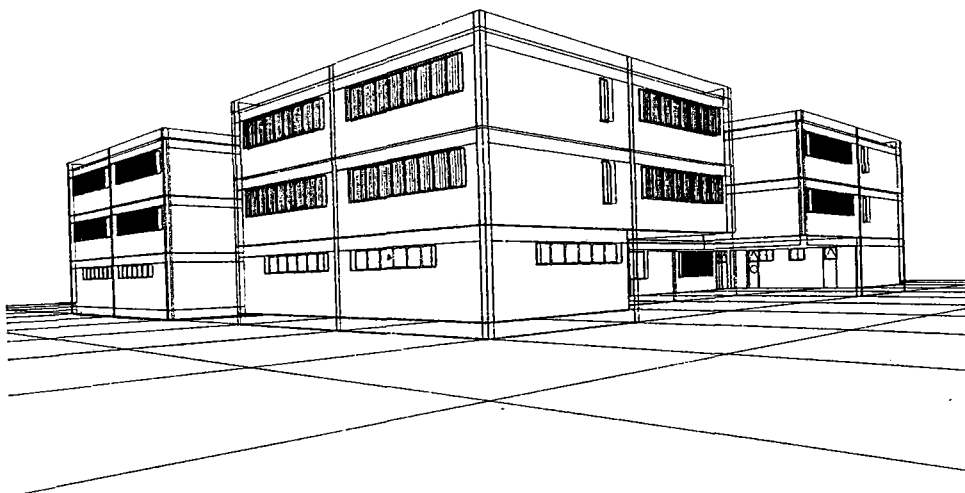
I de flesta systemen förekommer det både kopieringar och hänvisningar då man kommunicerar mellan projektdatabaser och kataloger. Detta innebär att bägge systemen måste tillgodoses i ett konverteringsformat.

12.4 Sammanfattning

Inget av systemen är bättre än något av de andra om man jämför dem med tanke på ett framtida konverteringsformat. Vill man trots allt välja något så rekommenderas Medusa och Intergraph. Detta endast med tanke på att de har lagerindelning och lagerindelningen är vanligare.

I övrigt är de tre systemen väl lämpade för ett framtida konverteringsformat.

Det man även bör beakta är i detta sammanhang är att CAD-systemen idag har en avskrivningstid på högst 2-3 år. Hur stor är sannolikheten att ett konverteringsformat blir färdigt och fungerande inom en så nära framtid?



tipo de centro docente	FP
PERSPECTIVA	fecha 11 nov 78

Exempel på perspektivritning utförd i CAD-system. (/CHR83b/)

13. ARMERINGSRITNING (Exempel)

13.1 Inledning

Antag att du har ritat ett stort hus helt färdigt och vill ta ut en armeringsritning. Hur gör du då?

Observera att detta bara gäller möjligheten att lösa problemet med avseende på databasens struktur.

13.2 GDS

I GDS använder man normalt sett objektsindelningen till att rumsindela. D.v.s. ett objekt är lika med ett rum. Använder man denna indelning får man söka igenom samtliga objekt efter armering, detta kommer i praktiken inte att fungera. Istället använder man en objektsindelning där man låter exempelvis ett bjälklag vara ett objekt.

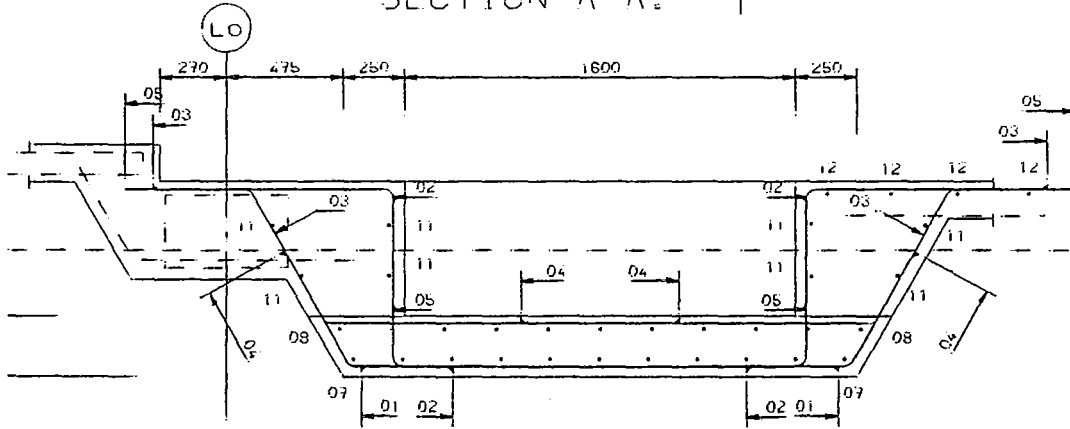
Vill du fortsätta arbetet i GDS får du lägga in armeringsjärnen manuellt, med mycket liten hjälp från systemet. Det finns ett armeringsprogram till GDS, men det fungerar inte i Sverige eftersom programmet är specialiserat för England och USA. (Se bild nästa sida.) En generellare programvariant är dock under utarbetande och väntas bli färdig under 1985. I dagsläget arbetar man med en koppling till Berit. Denna koppling fungerar i korthet så att man skapar t.ex. en bjälklagsplatta enbart genom att ange dess begränsningsytor, därefter skickar man den till Berit som skickar tillbaka den med all armering färdigt ilagd.

13.3 Medusa och Intergraph

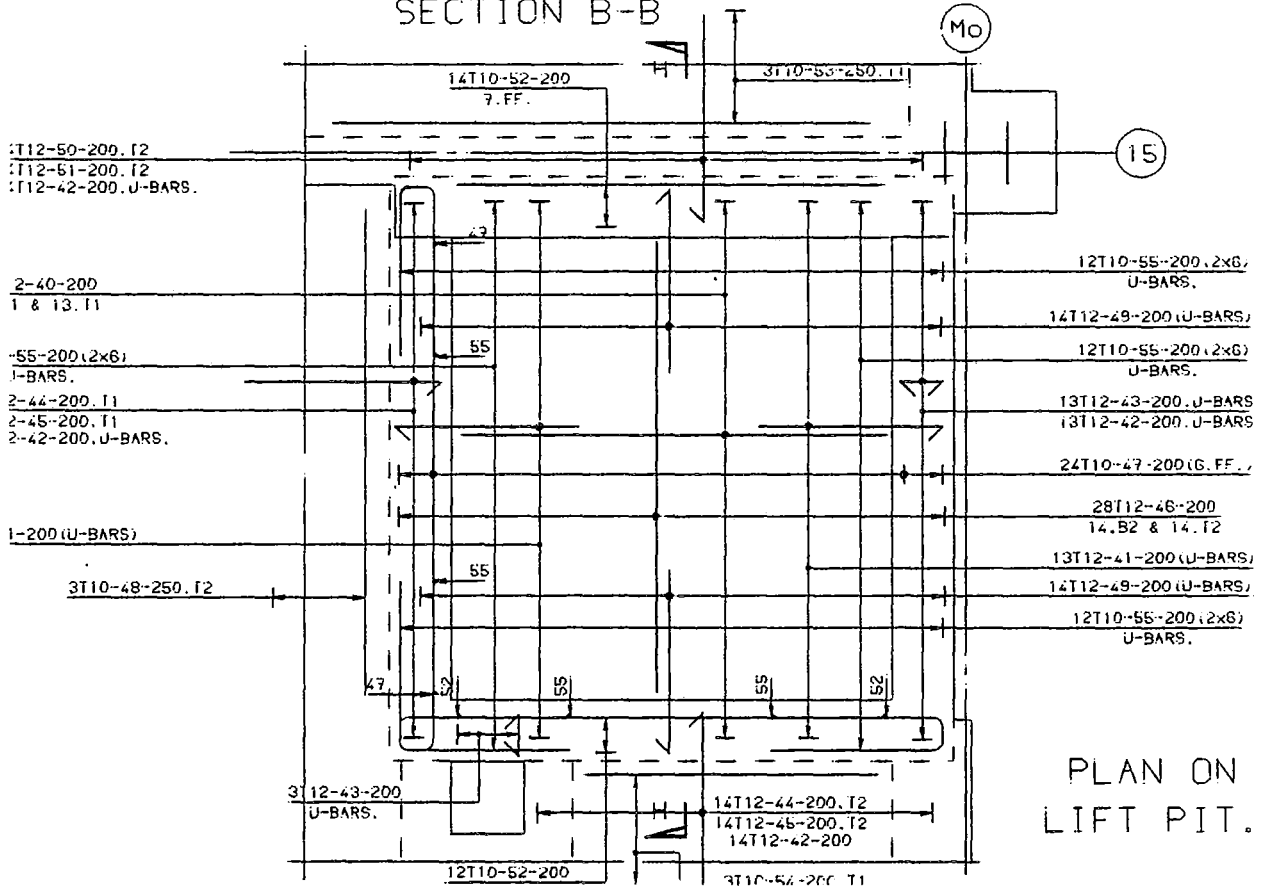
Både i Medusa och Intergraph arbetar man med lager. Metodiken för att ta ut en armeringsritning blir den samma i bägge CAD-systemen.

Man lagrar under ritandet t.ex. all armering i ett lager, ytterkonturer av väggar golv och dylikt i ett annat och kanske kompletterande text i ett tredje sedan tänder man bara dessa tre lager och gör en utskrift. Mycket enkelt och smidigt.

SECTION A-A.



SECTION B-B



PLAN ON
LIFT PIT.

Armeringsritning framställd med hjälp av GDSs ännu inte generaliserade armeringsprogram, RCDS. (Se /ARC83c/)

14. BYTE AV DÖRRAR (Exempel)

14.1 Inledning

Du har ritat ett sjukhus och i det satt in standard dörrar 90cm breda. Nu vill byggherren att hela huset skall anpassas för rullstolar, alla dörrar skall bytas till 105cm's bredd.

Eftersom du inte från början visste att dörrarna skulle bytas är elementidentifikation eller något motsvarande inte möjligt. Här måste du helt förlita dig på en bra databas med en uppdelning som gör denna typ av förändringar enkla.

14.2 GDS

I GDS är inte detta något problem eftersom du i ritningen bara gör hänvisningar till katalogen. Det enda du behöver göra är en liten ändring i katalogen. Detta under förutsättning att du har satt in dörrarna i huset på ett sådant sätt att en breddning inte medför att en dörr hamnar mitt i en vägg eller något liknande.

14.3 Medusa

I Medusa gör man kopior av katalogen. Detta medför att man inte kan göra lika smidiga ändringar som i GDS. Här blir man tvungen att söka igenom ett helt lager. T.ex. det lager som innehåller alla dörrar och fönster. I detta lager får man sedan gå från dörr till dörr och ändra alla som är fel.

Trots att systemet är mycket omständigare har det en stor fördel nämligen ur säkerhetssynpunkt. Eftersom man får gå från dörr till dörr så vet man att man inte ändrar något så det blir helt vansinnigt.

Standardprocedurer för ändringar av denna typen är inte helt lätta att göra, men så användbara att det bör vara värt besväret. En sådan procedur kan lämpligen förses med kontroller så att man inte gör ändringar som medför att databasen blir inkonsistent. Man bör även ha möjligheten att se varje ändring som görs.

14.4 Intergraph

I Intergraph har man lagerindelning och kopierar ur katalogen. Detta innebär att man måste ändra enligt den metod som beskrivits under Medusa.

15. SLUTSATSER

15.1 Allmänt

Alla tre systemen har bra databaser med en enkel strukturering.

Den största skillnaden mellan systemen är att GDS använder objektsindelning och Medusa & Intergraph lagerindelning. Praktisk innebär det att GDS är mer lämpat för små projekt medan de andra två passar bättre till stora arbeten.

Vid en jämförelse bör man tänka på att CAD-systemens avskrivningstider idag ligger på högst 2-3 år. Det är även intressant att titta på vilka möjligheter man har att använda den fasta information som finns i systemet då det uppdateras eller man byter till ett annat fabrikat.

15.2 GDS

GDS är det klart billigaste av de tre CAD-systemen. Det är ett system som är mycket enkelt uppbyggt, det finns objekt som innehåller grafik och text, dessa objekt kan du kopiera, vrida och byta skala på. Det är en stor fördel att systemet är enkelt eftersom det då är lätt att lära och lätt att utnyttja systemets hela kapacitet. Dock har systemet även nackdelar den största nackdelen är avsaknaden av lagerindelning. Att man saknar lagerindelning innebär att arbetet blir svårt att överblicka speciellt då datamängderna blir stora.

Några ytterligare nackdelar ser man redan då man tittar på de yttre förutsättningarna för arbetet. Man kan bara ha en vy på skärmen, vilket nog ofta är ett stort handikapp. Dessutom bör nog inte systemet rekommenderas till den som tänker arbeta frekvent i tre dimensioner eftersom man inte kan bygga upp någon tredimensionell modell.

GDS kan rekommenderas till den som vill ha ett CAD-system utan att ruinera sig. Det är definitivt inget dåligt system. Det bör kunna fylla sin funktion väl på ett mindre byggföretag och även i sådana applikationer som ligger utanför byggsektorn. Ett annat lämpligt användningsområde torde vara undervisning.

15.3 Medusa

Medusa är det utav systemen som är mest komplicerat utan att för den skull vara svårt att använda. Dock lär det ta lång tid att lära sig utnyttja systemet fullt ut. Vilket ju inte heller är nödvändigt för alla användare.

Lagerindelningen som man har i Medusa gör att man ganska lätt kan få stora datamängder överskådligt strukturerade, det medför att systemet fungerar bra för stora objekt. Den enkla grundläggande hanteringen medför att Medusa även passar bra till mindre objekt.

Medusa är det utav de tre systemen som vänder sig till flest olika typer av användare.

15.4 Intergraph

I Intergraph väljer man själv i högre grad än i de andra systemen hur datastrukturen skall se ut, detta är en ganska komplicerad affär, det medför att systemet är tungarbetat då man arbetar med mindre objekt. Strukturen i sig är däremot inte komplicerad, den skiljer sig från de andra systemen eftersom grafiken lagras för sig men mer komplicerad är den inte.

Eftersom egenskapsdatabasen är fri från grafiken kan den användas för vad för data som helst, detta i sin tur innebär att systemet är betydligt allmännare än något av de andra systemen. Vilket även borde innebära att systemet skulle vara bättre förberett för framtida utvidgningar och förändringar.

Systemet har en komplicerad syntax, detta kompenseras genom att man mycket lätt gör egna fortranprogram till systemet. Två bildskärmar är en lyx som man nog inte vill vara utan om man en gång har haft det.

Intergraph är ett system som passar bäst till stora objekt och då man vill ha möjlighet att arbeta med en separat databas.

15.5 Slutord

De tre CAD-systemen är alla bra men vänder sig till olika användare och användningsområden. Ser man till priset är GDS den givna vinnaren, ser man till prestanda är det svårt att avgöra om Medusa eller Intergraph är bäst. Det som är viktigt är att systemet har lagerindelning, alla de övriga attributen förutom egenskapsdatabaser är av mindre betydelse. Medusa vill jag dock utnämna som vinnare, då med tanke på att

systemet är modernare och mer lättarbetat. Utvecklingen hos Intergraph kommer dock att innebära att grafiken integreras i projektdatabasen. En god gissning är att Intergraphs CAD-system modell 1985 kan erbjuda ungefär samma finnesser som Medusa gör, för det har hunnit hända mycket sedan Intergraph lanserade sitt system, det system som jag har tittat på, 1973.

16. LITTERATURFÖRTECKNING

- /ARC78/ Applied Research of Cambridge, 1978. BDS Building Design Systems, Proven computer back-up for building designers and contractors.
- /ARC81a/ Applied Research of Cambridge, 1981. BDS Building Design Systems, Basic principles and sample results.
- /ARC81b/ Applied Research of Cambridge, 1981. GDS Reference Manual.
- /ARC82a/ Applied Research of Cambridge, 1982. GDS Arbitrary Data.
- /ARC82b/ Applied Research of Cambridge, 1982. General Drafting System.
- /ARC82c/ Applied Research of Cambridge, 1982. Building Design System.
- /ARC82d/ Applied Research of Cambridge, 1982. GDS Macro Basic.
- /ARC82e/ Applied Research of Cambridge, 1982. GDS/GNC Graphical Numerical Control.
- /ARC82f/ Applied Research of Cambridge, 1982. ABS Oldacres Bills of Quantities System.
- /ARC83a/ Applied Research of Cambridge, 1983. Fact Sheet.
- /ARC83b/ Applied Research of Cambridge, 1983. Beskrivning för EROS.
- /ARC83c/ Applied Research of Cambridge, 1983. Reinforced Concrete Detailing System.
- /ARC/ Applied Research of Cambridge. Systems for building design.
- /BER84/ Berglund, A, Bergström, A, Olsson, M, 1984. Konverteringsformat för CAD-system, Framtagning av programvara. Datorstödd Installationsprojektering AB, Skärholmen.
- /BET82/ Bechtel power corporation, 1982. Computer aided design and drafting. San francisco power division.

- /BYG84/ Byggmästaren 4, 1984. Pilotprojekt i Bollnäs utvärderat: CAD kan bli effektivt redskap.
- /BÖ84/ Bö, Ketil, Direktör SIU, 1984. Standardisering innen CAD/CAM. Newsletter Nr 2 jan 84, Skandinaviska CAD/CAM forskning.
- /CHR82b/ Christiansson, P, 1982, Kommunikation med GDS.
- /CHR83a/ Christiansson, P, 1983 Datorstödd projektering, CAD i tillämpning. Tekniska Högskolan, Lund.
- /CHR83b/ Christiansson, P, Östlund, L, 1983. Synpunkter på beräkningsredovisning. Tekniska högskolan, Lund, Rapport TVBK-3017.
- /CHR84a/ Christiansson, P, 1984. Datorstödd projektering, CAD, Besök vid företag och universitet, USA hösten 1983.
- /CHR84b/ Christiansson, P, 1984. Intergrated computer aided design. Present and future data structures. Lund Institute of Technology.
- /CHR84c/ Christiansson, P, 1984. Internationell konferens om datorstöd i arkitektarbetet, London oktober 1983.
- /CHR84d/ Christiansson, P, 1984. Utvecklingstendenser för användande av datateknik i byggandet. Nordiskt byggforskningsmöte 12-14 september, Lejondal. Tekniska Högskolan i Lund.
- /CIS82a/ Cambridge Interactive Systems, 1982. User manual 2-D System.
- /CIS82b/ Cambridge Interactive Systems, 1982. General Drafting System, An overview.
- /CIS84/ Cambridge Interactive Systems, 1984. MEDUSA, CIS announces new multi-discipline tool for the UK building industry.
- /DAT84/ Datornytt 12/84. CAD/CAM översikt.
- /EKM81/ Ekman, T, Karlsson, J, 1981. Pascal för dig som kan programmera. Studentlitteratur, Lund.
- /HOS79/ Hoskins, E, 1979. Design development and description using 3D box geometries. IPC Business Press.

- /INT82a/ Intergraph Corporation, 1982. Data definition language, users guide, DIXD1480.
- /INT82b/ Intergraph Corporation, 1982. Compendium of DMRS 8.7 software, DIXD2400.
- /INT82c/ Intergraph Corporation, 1982. VAX, DMRS Worksheet Editor (DWE), users guide (8.7), DIXD2280.
- /INT82d/ Intergraph Corporation, 1982. VAX DMRS, Host operations language, users guide (8.7), DIXD2300.
- /INT83/ Intergraph Corporation, 1983. Company Overview.
- /JON84/ Jonsson, J, E, 1984. Datoriserad projektering och arbetsplanering. Väg och Vattenbyggaren 9-10 1984.
- /MIT77/ Mitchell, W, 1977. Computer-Aided Architectural Design. Petrocelli/Charter, New York.
- /NIL83/ Nilsson, A, Blomberg, H, 1983. CAD för byggnads-konstruktioner, Jämförelse av tre ritningssystem. Kungliga Tekniska Högskolan, Stockholm.
- /NOR82/ NordCad AB, 1982. Kommunikation med GUS.
- /PAG83/ Page, E, Wilson, L, 1983. Information Representation and Manipulation using Pascal. Cambridge University Press.
- /RIC/ Richens, P. Dis-integrated models for building design. ARC Cambridge.
- /ULF82/ Ulfsby, S, 1982. Tilgjengelig databaseteknologi, Datstrukturer og databaser i GPM. Sentralinstitutt for industriell forskning, Oslo, Norge.
- /UNR83/ Unruh, E, 1983. Val av CAD-miljö ett praktikfall. Högskolan i Luleå.
- /USA83/ US Army Corps of Engineers, 1983. Computer-Aided Engineering and Architectural Design System (CAEADS).