

Gränssnittsprogrammet ELVAR

Gränssnitt mot Cad-komponenter

på NICK-format och Autocad

Vidar Jónsson

Arkiv

Konstruktionsteknik



Rapport TVBK-5071
ISSN 0349-4969
ISRN: LUTVDG/TVBK--5071-SE

Gränssnittsprogrammet ELVAR

Gränssnitt mot Cad-komponenter
på NICK-format och Autocad

EXAMENSARBETE TVBK-5071

Handledare: Per Christiansson
LUND april 1994

Vidar Jónsson

Förord.

Denna rapport är resultatet av ett examensarbete som utförts vid Avdelningen för Bärande Konstruktioner, KBS-Media Lab, Lunds Tekniska Högskola. Det är utfört av Viðar Jónsson underHandledning av Per Christiansson.

Jag vill dessutom tacka Anders Follin för hans hjälp

Lund, Mars 1994

Viðar Jónsson

Sammanfattning.

Arbetet har omfattat ett studium av hur Cad-komponenter på NICK-format lätt skall kunna hanteras och sändas till ett Cad-system (Autocad).

Första delen av rapporten är en beskrivning av NICK-formatet. Sedan följer beskrivning av Cad-komponentfilerna och rekommendationer för dessa. Kapitel 5 handlar om tillgängliga elektroniska varuinformationssystem och innehåller en kort beskrivning av dessa.

I senare delen av rapporten beskriver jag gränssnittsprogrammet Elvar som har utvecklats. Det tar hand om sökning av Cad-komponenter i ett komponentbibliotek, visning av komponenten på en skärm (figur och tillhörande attribut) och export av komponenten till Autocad. Programmet har utvecklats i Visual Basic för Windows i PC-miljö. Databasen Access har även integrerats.

Innehållsförteckning.

1. Inledning.....	1
1.1. Bakgrund.....	1
1.2. Syfte.....	1
1.3. Modellering av gränssnittsprogrammet ELVAR.....	2
2. NICK-standarden.....	3
2.1. Syfte med NICK standarden.....	3
2.2. NICK - filens uppbyggnad.....	5
2.3. För och nackdelar med NICK-formatet.....	11
2.3.1 Fördelar.....	11
2.3.1 Begränsningar.....	12
3. Cad-komponenter på NICK-format.....	13
3.1. Grafik.....	14
3.2. Klassifikationer och relationer till BSAB.....	14
3.3. Attribut.....	16
4. Rekommenderade egenskaper för Cad-komponenter.....	17
4.1. Orientering:.....	17
4.2. Vyer.....	18
4.3. Grafiknivåer.....	19
4.4. Linjekaraktär (linjetyp).....	19
4.5. Linjebredder.....	20
4.6. Färg.....	20
4.7. Typsnitt.....	21
4.8. 3D-grafik.....	21
4.9. Parametrisk grafik.....	21
4.10. Attribut.....	22
4.11. Exempel på Cad-komponentfil.....	22
5. Datoriserade varuinformationssystem.....	26
5.1. Syfte med datoriserade varuinformationssystem.....	26
5.2. Kort beskrivning av några tillgängliga system.....	26
5.2.1. Windows for Windows, demonstratör.....	26
5.2.2. AMVI, demonstratör.....	28
5.2.3. Svensk Byggtjänst byggvaruregister.....	28

6.	Kommunikation emellan Cad-komponentfiler och datoriserade varuinformationssystem.	30
6.1.	Syfte med kommunikationen.....	30
6.2.	Kravanalys.....	30
6.3.	Begränsningar i Cad-komponentformatet.....	34
7.	Uppbyggnad av gränssnittsprogrammet ELVAR.....	37
7.1.	Inledning	37
7.2.	Sökning av Cad-komponenter.....	38
7.2.1.	Sökning efter filnamn:	38
7.2.2.	Sökning i sökfil	38
7.3.	Läsning av Cad-komponentfil och sortering av informationen.	40
7.4.	Skrivning av attribut och annan text i textrutorna.....	42
7.5.	Uppritning av figurer.	42
7.6.	Export till Autocad.....	44
7.7.	Import av Cad-komponent till Autocad (lisp-programmet Elvar).....	45
8.	Användning av programmet ELVAR.	47
8.1.	Sökning.....	47
8.1.1.	Sök i sökfil:.....	47
8.1.2.	Sök efter filnamn:.....	49
8.2.	Attribut och figurer.....	49
8.3.	Export.....	51
8.4.	File menyn.....	52
9.	Slutsatser.....	54
10.	Ordlista.	56
11.	Referenser	58

1. Inledning.

1.1. Bakgrund.

På senare år har datorn blivit använd mer och mer för att göra ritningar med hjälp av Cad-system. Den bjuder på möjligheter som inte var så enkla att göra när ritningar ritades för hand. Dessa möjligheter är till exempel att inkludera produktinformation i ritningsfilen, som är tillgänglig i datorn, men som inte kommer fram på själva ritningen. När det gjorts uppstod problem med kommunikation mellan olika Cad-system. För att lösa det problemet har det satts igång några projekt för att ta fram neutralt filformat som klarar av att överföra produktinformation med ritningen. Dessa projekt är bl. a. STEP [9] som är samarbete inom Europa, och NICK [2] som är svenskt projekt, men är tänkt att anpassas till STEP när det är färdigt.

För att göra det enklare att skapa ritningar som innehåller produktinformation har det satts igång projekt för att ta fram rekommendationer för Cad-komponenter på NICK-format. Dessa komponenter är symboler som beskriver en vara, till exempel dörr eller tvättställ och innehåller både produktinformation och grafik.

1.2. Syfte.

Syfte med examensarbetet är att belysa krav som kan komma att ställas på elektronisk varuinformation, då denna skall kommunicera mot standarder för kommunikation under utarbetande inom byggbranschen (Cad-komponenter på NICK-format).

Vid KBS-Media Lab har prototyper till elektroniska varuinformationssystem framtagits. Dessa system är 'AMVI' [10] och 'Windows for Windows' [11].

Ett program har utvecklats som innehåller gränssnitt mot Cad-komponenter som är datafiler med standardiserad information enligt NICK-standarderna, samt gränssnitt mot Cad-system. Programmet har utvecklats i PC-miljö i språket Visual Basic. I samband med programmeringen har utvärdering skett av existerande standardföreslag med kritiska synpunkter på fördelar och nackdelar vid denna integration.

Resultatet presenteras i form av denna rapport och demonstratorn ELVAR (se kapitel 7).

1.3. Modellering av gränssnittsprogrammet ELVAR.

Programmet Elvar är tänkt att utgöra ett gränssnitt mot Cad-komponenter och Autocad. Det skall vara ett hjälpmedel för sökning i komponentbibliotek, man skall snabbt kunna rita figurer av komponenterna på skärmen och se tillhörande attribut samtidigt. När man har valt Cad-komponent för att lägga in på en ritning (i Autocad) skall man kunna exportera komponenten till Autocad. Programmet skall köras i Microsoft Windows miljö, på IBM-kompatibla PC-datorer.

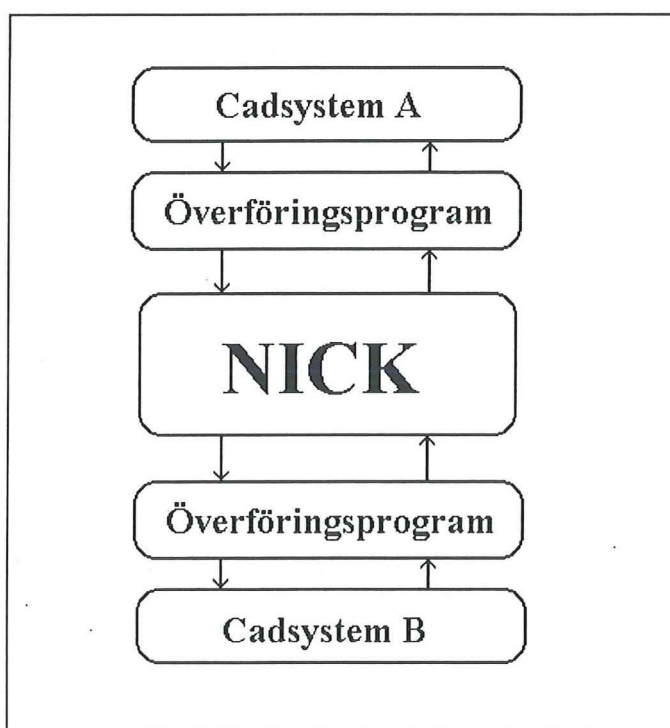
Den programvara som används för att skriva Elvar är följande:

1. Microsoft Visual Basic för Windows Version 3.0 för Windows.
2. Microsoft Windows 3.1 API-funktionbibliotek (Application Programming Interface).
3. Microsoft Access 1.0, Databas som kopplas direkt till Visual Basic.
4. Autolisp Release 11, för import av Cad-komponent till Autocad.

2. NICK-standarden

2.1. Syfte med NICK standarden

Vi är nu på väg mot en mer intelligent hantering av CAD-ritningar, där man ritat upp byggdelar och definierar tillhörande information, som tjocklek, höjd, typ, brandklass och ljudklass för en vägg. Därför har NICK-standarden blivit framtagen, för att överföra alfanumeriska data, mellan olika CAD-system, tillsammans med grafiken, så att 'intelligensen' behålls i ritningen. NICK står för Neutral Intelligent Cad-Kommunikation och medför att information kan utbytas på ett standardiserat sätt oberoende av vilket Cad-system man har valt.



Figur 2.1. NICK-standardens princip.

Andra standarder t. ex IGES (Initial Graphics Exchange Specification) och DXF (Drawing Interchange File) är inte användbara därför att de är endast gjorda för att överföra grafik. Senare versioner av dessa överföringsprogram klarar av att överföra information om lager, färg, och vissa andra egenskaper men inte någon produktinformation. IGES- och DXF-filer blir ganska stora, i de flesta fallen större än ritningsfilen från Cad-systemet.

Det finns också ett Finskt system för överföring av ritningar. Det kallas för BEC (betongelement-CAD) och är utvecklat av Matti Hannus i samarbete med Betongindustriförbundet i Finland. Det systemet kan hantera grafik och information om lager, färg och vissa andra egenskaper. Överföringsfilen är mycket kompakt men inte direkt läsbar.

När utvecklingen av NICK-standarden började, bildades en grupp bestående av 10 företag. Denna grupp tog fram en lista med de krav som projektet skulle uppfylla. [1]

- Hitta format för arkivering, där de ursprungliga ritningarna kan återskapas. Symboler ska kunna lagras som referens till speciell ritning/fil eller så måste de finnas lagrade för varje förekomst med hjälp av enkel grafik.
- Överföra 'hela ritningsgrafiken' och intelligent information enligt byggdelskonceptet mellan parterna i integrerad projektering. Överföringen ska bygga på ett antal 'grundtyper' för geometrierna.
- Alla discipliner och faser i ett byggprojekt ska betraktas.
- Både ritnings- och modellorienterade CAD-system ska hanteras.
- All information ska vara 'läsbar' dvs i ASCII-format.
- Projektet ska ta till vara internationell utveckling och om möjligt även påverka densamma. Det är viktigt att så mycket som möjligt görs för att få med svenskt byggtänkande i det format som kommer att bli standard i Europa. Risken finns annars att de svenska byggföretagen måste arbeta med betydligt 'primitivare' hantering av information och mängder än i dag.
- Ett önskvärt mål vore om det *Neutrala formatet* kunde bli något som skrevs in i standardavtal för projektering och överlämnande till byggherren.
- Formatet ska kunna överföras med hjälp av EDIFACT-syntaxen (Electronic Data Interchange For Administration, Commerce and Transport).
- NICK skall baseras på STEP-formatet för geometrier så snart det finns framme och har accepteras.

2.2. NICK - filens uppbyggnad

NICK-filen består av 'records' som delas upp i följande fyra grupper.

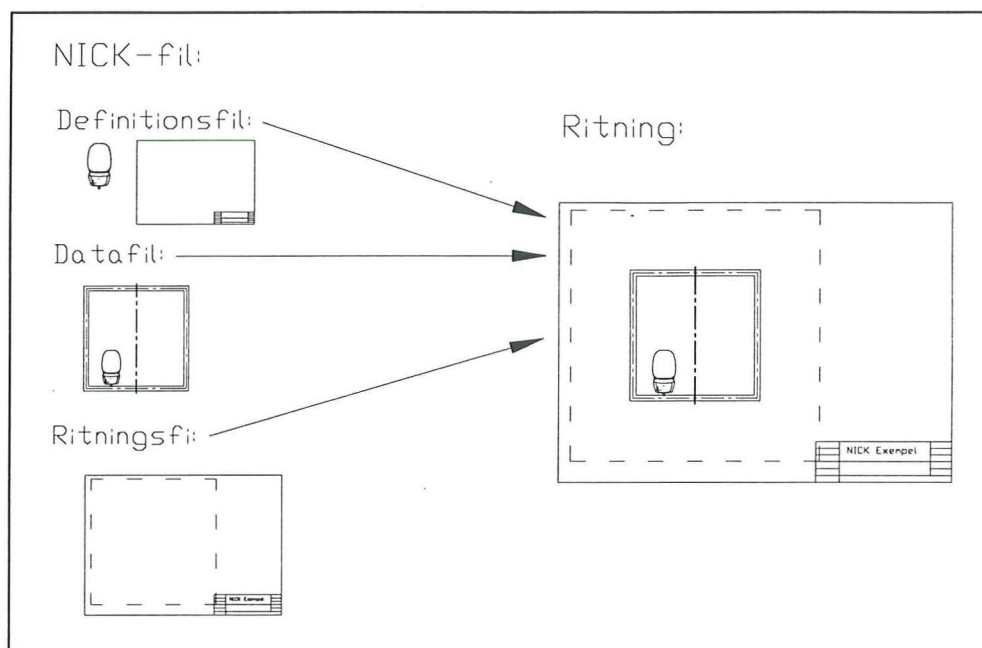
Administrativa records är alla record som börjar på bokstaven H, t. ex. HFI, HPR, HCO. De inleder filer eller markerar början av en recordgrupp.

Definitionsrecords som definierar linjer, texter, koder, m.m.

Datarecords som beskriver byggdelar, system, utrymmen och grafiska grupperingar.

Ritningsrecords som beskriver innehållet i ritningen.

I NICK-formatet finns det tre filtyper, **definitionsfil**, **datafil** och **ritningsfil**, som alla består av administrativa records som är obligatoriska och därefter definitionsrecords, datarecords och ritningsrecords efter behov.



Figur 2.2. Nickfilens uppbyggnad.

Definitionsfil:

NICK har en officiell definitionsfil där koder, typer, namn och standardsymboler definieras. Det är också möjligt att definiera egna koder, typer, namn och standardsymboler i en egen definitionsfil som skickas över vid projektets start. Definitionsfil innehåller definitionsrecord som är av följande typ:

CXX (Code XX)

Definierar olika typer av koder, t. ex attributkod (CAT), Byggdelskod (BD-kod) (CBP), Skrafferingskod (CCH) och fonttyp (CFO).

DSY (Symbol definition)

För att definiera standardiserade symboler. Symbolerna skall finnas tillgängliga i ett symbolbibliotek och vid överföringen skickas: symbolnamn som hänvisar till symbolbiblioteket, insättningspunkt och rotation för symbolen .

00001	HFI	00.1	940120	Nick-exempel	KBS-Media lab	VJ	Def. fil
00002	HPR	EXJOB	MM	DEG	10000.	10000.	0. - 0.000 SWE
00003	HCO	Användardefinition					
00004	CAT	\$BREDD	Bredd				
00005	CBP	363	Innerväggar				
00006	CCH	1	45 grader åt höger				
00007	CFO	HELV	Helvetica fylld				
00008	CGG	300	Stomlinjer, baslinjer				
00009	CLI	1	Heldragen				
00010	CMA	7	Gips				
00011	CPO	10	Ofylld cirkel med 1 mm diameter				
00012	CSM	521	Tappvattensystem				
00013	CSP	123	Kontorsrum				
00014	CTY	\$V712	Gipsvägg 95mm				
00015	DSY	355.1	40	\$F16	Fönster		

Figur 2.3. Exempel på definitionsfil.

Datafil:

Överför byggdelar, grafiska grupperingar, system och rum. Filen kan innehålla hela projektet eller delar därav. Filen förs över under projektets gång. En datafil kan innehålla datarecord som är av följande typ:

BSM (Building system)

System med relationer till andra system, byggdelar eller utrymmen, och attribut som beskriver deras egenskaper. System har ingen grafik. Exempel på system är installationsdelar.

BPA (Building part)

Byggdelar beskriver de objekt som NICK arbetar med på den finaste nivån. Byggdelar består av grafik och attribut. Byggdel kan vara grafiska element som ritas direkt eller symbol med tillhörande attribut, insättningspunkt och rotation.

BSP (Building space)

Utrymmen, beskriver rum med grafik och attribut. Exempel är att man ritlar en linje runt ett rum för att definiera dess läge och knyter till det ett attribut för att beskriva rummets funktion och andra egenskaper.

GGR (Graphical grouping)

Grafiska grupperingar beskriver objekt som inte är byggdelar, rum eller system, exempel: skraffering och konnektionslinje som anger var i modellen skiljelinje går mellan överlappande ritningar.

Efter ovanstående record följer record som beskriver attribut eller ritlar linjer och cirkelbågar. Här följer beskrivning av några record av den typen.

ATT (Attribute)

Record som inleder ett attribut. Recorden består av nummer och typ som alla andra, grafikpåverkan (0 = ingen grafik, 1 = plan grafik, 2 = 3d-grafik), antal värderader, format på värderaderna, attributnamn och beskrivande text. Grafikpåverkan är 0 för ingen grafik, 1 för plan grafik och 2 för 3D grafik. Antal värderader visar hur många record av typen VAL som följer och som tillhör attributet. Format på värderaderna kan vara I för integer, R för real tal och C för character. Beskrivande text skrivs endast ut för attribut som inte finns i definitionsfilen.

TXT (Text)

Textdefinition som innehåller bredd och höjd på texten, skala på ritningen, justering, fonttyp, linjetjocklek och färg. Justering anger justering av textens datumpunkt i textbox. Koden 0000 står för datumpunkt i nedre vänstra hörnet, 0505 står för datumpunkt i mitten av texten, 1010 står för datumpunkt i övre högra hörnet, och så vidare. Färg är kod på 6 tecken och anger blandning av färgerna röd, gul och blå.

VAL (Value)

Värderad som följer record av typen ATT eller TXT och innehåller det värde som attributet skall ha eller texten som skall skrivas. En VAL record innehåller endast ett värde.

LGR (Line graphics)

Definierar linjegrafin och består av linjetyp, linjetjocklek och färg. Linjetyp är ett tal på maximalt 3 tecken. Linjetjockleken följer DIN-standarderna och kan vara 0.18, 0.25, 0.35, 0.5, 0.7 och 1.0 mm. Färg har beskrivits under recorden TXT.

LCO (Line coordinates)

Består av x-, y- och z-koordinater för punkter, förbindelse, synlighet och punkttyp. Förbindelse beskriver viken typ av element som ska ritas till nästa punkt, L står för rak linje, C står för cirkelbåge och + står för centrum på cirkelbågen. Synlighet kan vara V för synlig (visible), och I för osynlig (invisible).

DPO (Datum point)

Record som innehåller insättningspunkt för symboler eller text. DPO består av x-, y- och z-koordinater för insättningspunkten och rotation för symbolen.

00001	HFI 00.1	940120	Nick-exempel	KBS-Media lab	VJ	Data fil
00002	HPR EXJOB	MM	DEG	10000.	10000.	0. 0.000 SWE
00003	HVOHUS1	PLAN1	123	123	123	
00004	DPO	0	0	0	0.00	
00005	BSM 00007	521	12	N	Ny fil	SYS1
00006	ATT 1	4	I	\$MAXFLOW	Maximalt flöde m3/min	
00007	VAL 100					
00008	BPA 00006	3630	1	10	Del A	N Ny fil I \$VA716
00009	ATT 0	1	I	\$ID	ID-nummer	
00010	VAL 5					
00011	ATT 1	1	R	\$TJKL	Tjocklek	
00012	VAL 95.					
00013	ATT 2	1	R	\$HOJD	Höjd	
00014	VAL 2600					
00015	LDE					
00016	LCO 0.00	0.00	0.00	L	V	O
00017	LCO 2000.00	0.00	0.00	L	V	O
00018	LCO 2000.00	2000.00	0.00	L	V	O
00019	LCO 0.00	2000.00	0.00	L	V	O
00020	LCO 0.00	0.00	0.00	E	V	O
00021	BPA 00006	3752		30	Del A	N Ny fil I TOAL1
00022	ATT 0	1	I	\$ID	ID-nummer	
00023	VAL 9					
00024	ATT 0	1	C	\$FABR	Fabrikant	
00025	VAL Gustavsberg					
00026	ATT 0	1	C	\$FARG	Färg	
00027	VAL vit					
00028	DPO 500.00	50.00	0.00	180.00		
00029	BSP 00006	0222	1	Del A	N	Ny fil Toalett
00030	ATT 0	1	I	\$RUMNR	Rumsnummer	

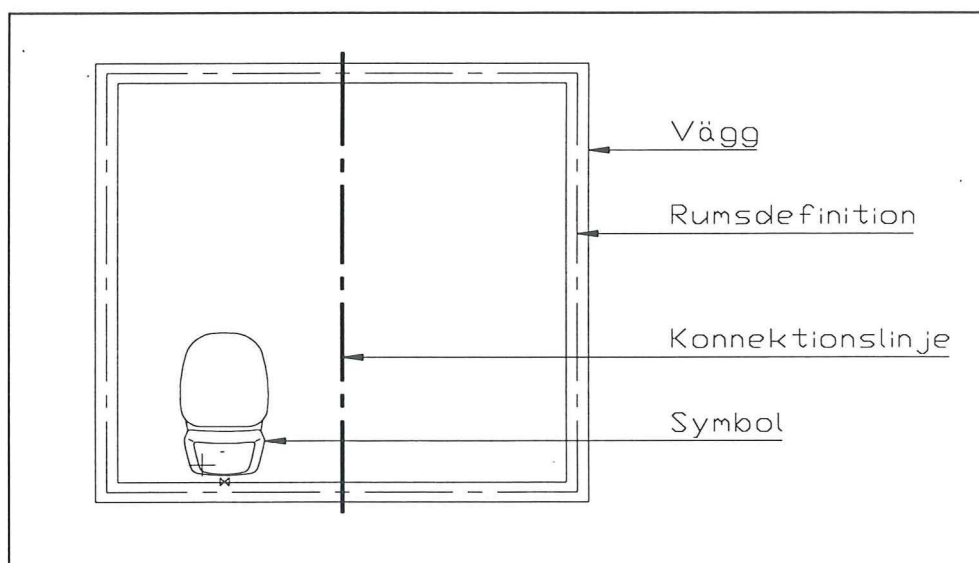
```

00031 VAL 110
00032 ATT 0      1      C      $RUMNAM      Rumsnamn
00033 VAL Toalett
00034 LDE
00035 LCO 0.00   0.00   0.00   L      V      O
00036 LCO 2000.00 0.00   0.00   L      V      O
00037 LCO 2000.00 2000.00 0.00   L      V      O
00038 LCO 0.00   2000.00 0.00   L      V      O
00039 LCO 0.00   0.00   0.00   E      V      O
00040 GGR 701    A      N      Ny fil    940120
00041 ATT 0      1      I      $ID      ID-nummer
00042 VAL 14
00043 LGR 4      0.5    000000
00044 LCO 1000.00 -100.00 0.00   L      V      O
00045 LCO 1000.00 2100.00 0.00   E      V      O

```

Figur 2.4. Exempel på datafil.

- Rad 00001 - 00003 Administrativa record, inleder filen.
- Rad 00004 Insättningspunkt för ritningen i projektet.
- Rad 00005 - 00007 Definition av byggnadssystem, som har attributet maxflow.
- Rad 00008 - 00020 Byggdelen vägg (buildingpart), attribut och grafik.
- Rad 00021 - 00028 Insättning av symbol på ritningen (buildingpart), med tillhörande attribut och insättningspunkt.
- Rad 00029 - 00039 Rumsdefinition (building space), rumsnummer, namn och linjer som definerar rummets läge.
- Rad 00040 - 00045 Definition av konnektionslinjen där ritningen skall delas på två ritningsblanketter.



Figur 2.5. Datafil, grafik.

Ritningsfil:

Definierar innehållet i ritningar med hjälp av byggdelar från modellen och detaljer som tillhör respektive ritning. Ritningsfilen innehåller ritningsrecord och datarecord för att definiera t. ex. ritningsfönstret och beskrivande text. Ett ritningsrecord är av följande typ.

HDR (Header drawing)

Ritningshuvud, anger ritningsnummer och storlek på ritningsblanketten.

DWI (Drawing window)

Ritningsfönster som definieras i modellens koordinater. Samtidigt definieras lägesbegrepp för de objekt som finns inom ritningsfönstret (t. ex. hus1, plan1), skala och datumpunkt och rotation i papperskoordinater.

WCO (Window content)

Innehållet i fönstret definieras av: 'påverkan', GG-koder (Graphical Grouping - koder) och BD-koder (Byggdels - koder). Variabeln påverkan är första tecken efter WCO och är plus (+) om objekt skall finnas med på ritningen och minus (-) om objekt inte skall finnas med på ritningen. Om posten WCO saknas tas alla objekt inom fönstret med på ritningen.

DET (Detail)

Anger grafiska element som endast finns på ritningsblanketten. De kan innehålla grafik, attribut, text och symboler och ritas i skala 1:1. Exempel på DET är en ritningsram.

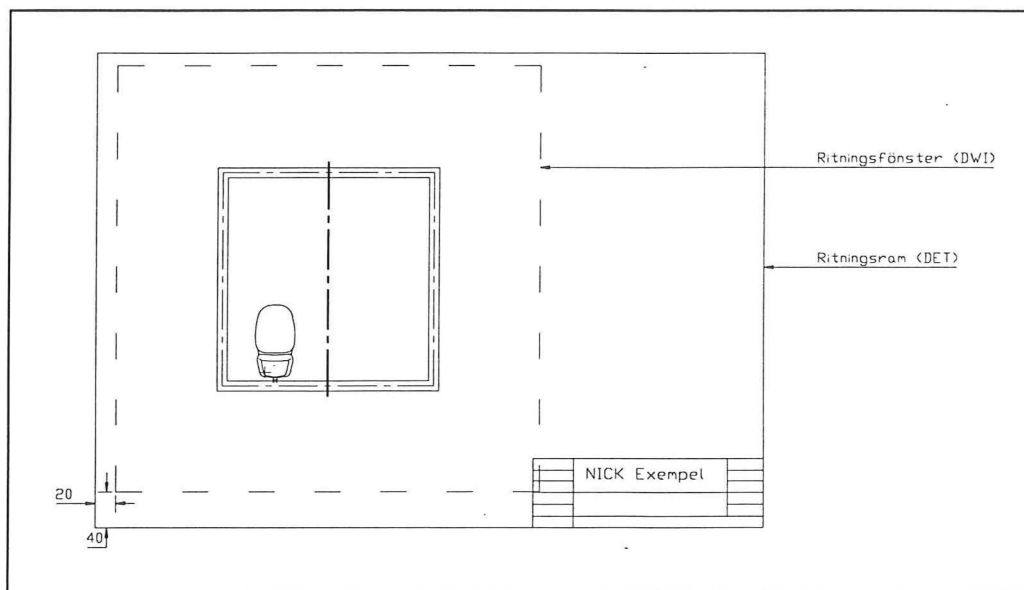
```

00001 HFI 00.1 940120 Nick-exempel KBS-Media lab VJ Ritn. fil
00002 HPR EXJOB MM DEG 10000. 10000. 0. 0.000 SWE
00003 HDR 120:1 420.00 297.00
00004 ATT 1 1 C $NAME Ritningsnamn
00005 VAL NICK Exempel
00006 DWI HUS1 PLAN1 100
00007 LCO -1000.00 -1000.00 0.00 L V O
00008 LCO 3000.00 -1000.00 0.00 L V O
00009 LCO 3000.00 3000.00 0.00 L V O
00010 LCO -1000.00 3000.00 0.00 L V O
00011 LCO -1000.00 -1000.00 0.00 E V O
00012 DPO 20.00 40.00 0.00 0.0
00013 DET 001 1.
00014 SYM RAM-A3 Ram för A3-blankett
00015 DPO 0.00 0.00 0.00 0.0

```

Figur 2.6. Exempel på ritningsfil.

Rad 00001 - 00002	Administrativa record, inleder filen.
Rad 00003	Ritningsnummer och blankettstorlek.
Rad 00004 - 00005	Ritningsnamn
Rad 00006 - 00011	Anger ritningsfönstret, allt som finns inom det skall med på ritningen, se figur 2.7.
Rad 00012	Insättningspunkten för ritningsramen på ritningsblanketten.
Rad 00013 - 00015	Ritning av ritningsram.



Figur 2.7. Ritningsfil, grafik.

De tre filtyperna kan slås ihop till en och bilda ett projekt.

Alla rader i en NICK-fil är numrerade för att lättare kunna kontrollera överföringen och referera till tidigare information.

Ovanstående beskrivning av records omfattar inte alla recordtyper men ger en översikt över de viktigaste. För närmare beskrivning av recorderna visas till NICK-rapporten R70, 1991.^[2]

2.3. För och nackdelar med NICK-formatet

2.3.1 Fördelar.

Normalt är NICK-filen mycket mindre än motsvarande ritningsfil, till skillnad från DXF som normalt ger en större fil. NICK - formatet är därför en ganska kompakt form för lagring av ritningar.

Filen är på ASCII-format och det är därför enkelt att se vad ritningen innehåller utan att rita upp den i något ritningsprogram. Utveckling av överföringsprogram blir också enklare.

Största fördelen, och kanske anledningen till att NICK-projektet började är att kunna överföra produktinformation mellan olika Cad-system, tillsammans med ritningen, som andra format såsom DXF inte klarar.

2.3.1 Begränsningar.

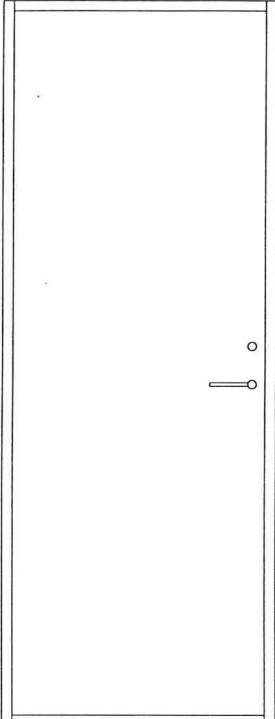
I NICK är linjer och cirkelbågar de enda grafiska element som går att använda, det går inte att beskriva komplicerade geometrier t.ex. 'nurbs' och 'spline'.

Det finns en del begränsningar i definitioner i första versionen av NICK. När NICK börjar användas kommer det att visa sig vilka fler definitioner som behövs, och eventuellt vilka nuvarande som är onödiga.

3. Cad-komponenter på NICK-format.

Skillnaden mellan NICK-filer och Cad-komponenter är att NICK-filen är avsedd för överföring av färdiga ritningar, medan en Cad-komponent kan behandlas som en styckvara för att överföra information om komponentbibliotek. Cad-komponenten är en textfil på ASCII-format (samma format och uppbyggnad som NICK-filen) som beskriver grafiken för en vara och också innehåller produktinformation i form av strukturerad text. En Cad-komponent kan också innehålla delkomponenter, som definieras som referenser.

Ett exempel på Cad-komponent är en dörr med dörrhandtag som delkomponent och brandklass, ljudklass, varukod och leverantör som produktinformation.

Cad-komponentfil på NICK-format.	Cad-komponent grafik.
<pre> Inledning av fil: 00001 HFI 001 0225 swdo-nsl-t-8 CC wrn Ny f4 : Attribut: 00007 DSY 5250 45 SWDO MS1-T-BH 101 Enkl massvärn trökn 00008 ATT 1 1 C IVARUKDD 00009 VAL XE222 00010 ATT 1 2 C IBSAB_P2 00011 VAL 36.5 00012 VAL - : Grafik: 00041 VIE FRONT 00042 LGR 301 0.35 000000 00043 LCD 0 0 2090 L I 0 00044 LCD 0 0 0 L V 0 : 00072 VIE BAKSIDA 00073 LGR 301 0.35 000000 00074 LCD 0 92 2090 L I 0 00075 LCD 0 92 0 L V 0 : 00103 VIE TOPP 00104 LGR 301 0.35 000000 00105 LCD 0 92 2090 L I 0 00106 LCD 0 0 2090 L V 0 : 00137 VIE VSIDA 00138 LGR 301 0.35 000000 00139 LCD 0 0 2060 L I 0 00140 LCD 0 0 15 L V 0 : 00179 VIE HSIDA 00180 LGR 301 0.35 000000 00181 LCD 690 0 2060 L I 0 00182 LCD 690 0 15 L V 0 : Texti: 00221 HDR swdo-nsl-t-8n 00222 TXT 40 35 30 0010 NDRM2 0.35 000000 00223 VAL swdo-nsl-t-8n 00224 DPO 0 0 0 00 </pre>	

Figur 3.1. Exempel på Cad-komponent, NICK-fil och grafik.

3.1. Grafik

Grafiken för en Cad-komponent består av raka linjer och cirkelbågar mellan punkter angivna i ett 3-dimensionellt koordinatsystem. Det är ett normerat ortogonalt koordinatsystem. I plan riktas x-axeln åt höger, y-axeln uppåt och z-axeln mot betraktaren. Dess origo är detsamma som Cad-komponentens insättningspunkt och x-axeln utgör riktning för komponentens rotation. Varje linje har sin linjetyp, som är linjebredd och färg. Linjetypen kallas för grafiskt attribut och tillhör varje linje.

Texter beskrivs som enskilda textrader med de grafiska attributen texthöjd och typsnitt. Insättningspunkten för en text medger också justering av texten i en textbox.

Cad-komponentens punkter koordinatsätts i mm i full skala och är därför oberoende av ritningens skala. Vissa komponenter väljer man dock att rita schematiskt, t. ex. komponenter i installationssystem såsom strömbrytare, ventiler m. m. För en sådan komponent väljer man en storlek som skall passa ritningens skala.

Grafiken kan innehålla beskrivningar av komponenten sedd i 6 olika vyer, d.v.s. topp, botten, front, baksida, vänster- och högersida. Origo är gemensam för alla vyerna.

3-dimensionell grafik beskrivs separat från de 2-dimensionella vyerna, men med samma insättningspunkt.

3.2. Klassifikationer och relationer till BSAB.

En Cad-komponent kan tänkas ingå i ett produktbibliotek som omfattar alla typer av byggvaror och installationsvaror. För att kunna söka och sortera komponenter som ingår i produktbiblioteket eller ett byggprojekt behövs ett klassifikationssystem. Här redovisas några klasser i ett klassifikationssystem. Samtliga BSAB tabeller /12/ finns i version från 1983, men det pågår arbete med BSAB-9x som är en ny version av BSAB och kommer att innehålla nya versioner av tabellerna till vilka hänvisning görs nedan.

Byggdel, installationssystem:

Byggdel är 'Planerad eller verklig fysisk, större enhet med bestämd huvudfunktion i byggnadsverk'. Exempel på byggdel är innervägg med huvudfunktionen avskiljande, eller pelare med huvudfunktionen bärande.

Inom installationsområdet är motsvarigheten till byggdel, installationssystem, som är benämning på samverkande installationsdelar i en byggnad.

För byggdelar och installationssystem gäller tabell BSAB-P2.

Byggdelstyp:

Byggdelstyp är en byggdel specificerad med avseende på teknisk lösning eller material. Exempel på byggdelstyp är armerad betong eller Bjälklag av lättbetong.

Klassificering av byggdelstyper kommer troligen att ingå i nästa version av BSAB men i NICK-projektet har gjorts en klassificering av byggdelar med tillhörande byggdelstyper (BD-koder).

Byggobjekt:

Komponent som är utplacerad i byggnaden kallas för byggobjekt. Varje byggobjekt kan uppdelas i en eller flera byggobjektdelar.

NICKs BD-koder används för klassificering av byggobjekt.

Varor:

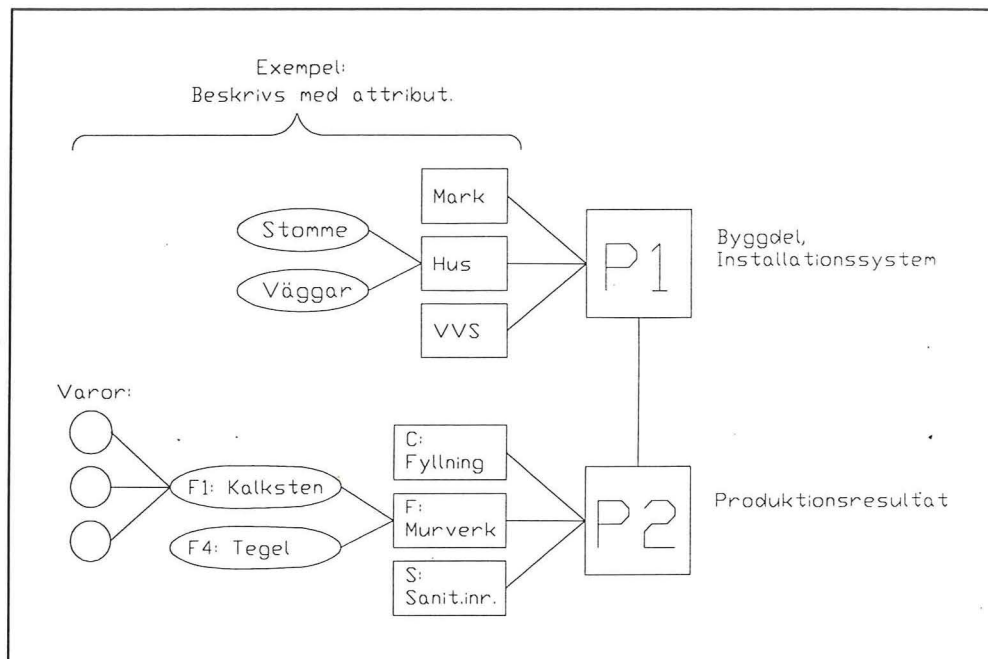
Varor är en av de ingående resurserna i produktionsresultat.

BSAB produkttabell 1 kan användas för klassificering av varor.

Produktionsresultat:

Är ett bestämt resultat av en aktivitet på byggplatsen för produktion av byggnad eller anläggning

För produktionsresultat gäller tabell BSAB-P1.



Figur 3.2. Byggdelar, varor och produktionsresultat i BSAB.

3.3. Attribut.

Attribut är information om komponentens egenskaper i form av text.

Attribut kan delas upp i två grupper, typgemensamma attribut och variantattribut. Typgemensamma attribut är inte geometrianknutna, men det är möjligt att knyta valmöjligheter till variantattribut, t.ex. färg, geometri, m.m.

Vissa attribut är obligatoriska, t.ex. varukod för varor, vissa är rekommenderade, t.ex. fabrikant och färg och andra är frivilliga. Obligatoriska och rekommenderade attribut är standardattribut i NICK-formatet och definieras av att de inte börjar med utropstecken (!).

4. Rekommenderade egenskaper för Cad-komponenter

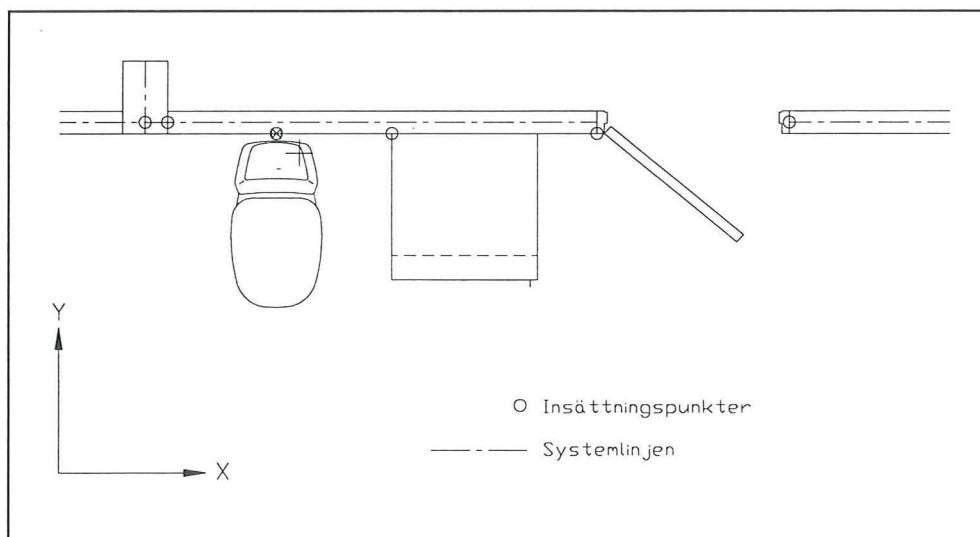
Det är viktigt att egenskaperna för Cad-komponenter är desamma oberoende av från vilken fabrikant de kommer. Därför har rekommendationer blivit framtagna för dessa egenskaper. I följande kapitel försöker jag beskriva viktiga delar av dessa rekommendationer.

4.1. Orientering:

Insättningspunkter för Cad-komponenter definieras enligt byggdelsklass. Stomkomponenter såsom väggar och pelare har en insättningspunkt som utgår ifrån systemlinjen. Kompletteringar placeras med utgångspunkt från stomkomponenten. I oroterat läge ligger utgångspunkten i övre vänstra hörnet av figuren, t. ex. för skåp. Undantag från denna regel utgör bl.a. öppningskompletteringar, där insättningspunkten ligger i samma väggliv som för skåpet, men i undre vänstra hörnet av figuren. Ett annat undantag är sanitetsporslin där insättningspunkten ligger på centrumlinjen.

Rotationsvinkeln är 0 grader när komponentens framsida ligger nedåt i toppvyn.

Komponenter i installationssystem följer samma regel som för stommar, d.v.s. insättningspunkten ligger i centrumlinjen.

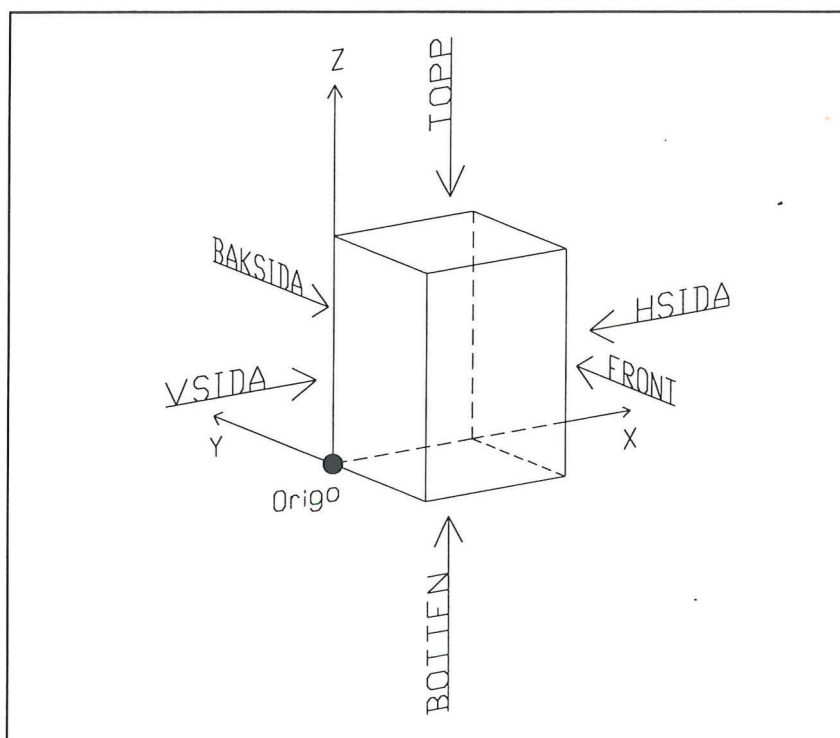


Figur 4.1. Insättningspunkter för olika typer av komponenter.

4.2. Vyer

Vyerna läggs så att de tillsammans utgör sidor i en kub med komponentens maximala utsträckning i alla leder, x-, y- och z- led.

Vyer anges med posten VIE i NICK-filen och föregår all grafik som tillhör vyn.



Figur 4.2. Vyer för Cad-komponenter

Det finns också 8 isometriska vyer definierade.

Vy	Normalens koordinater (x,y,z)
NVOVER	(÷1,1,1)
NVUNDER	(÷1,1,÷1)
NOOVER	(1,1,1)
NOUNDER	(1,1,÷1)
SVOVER	(÷1,÷1,1)
SVUNDER	(÷1,÷1,÷1)
SOOVER	(1,÷1,1)
SOUNDER	(1,÷1,÷1)

4.3. Grafiknivåer

Grafiknivåer beskrivs som första tecken i posten LGR. Tabell 1 ger en översikt över olika grafiknivåer.

1xx	Kontur	Enklaste representation, avsett för skala 1:200 och 1:500, t.ex. översiktsritningar.
2xx	Översikt-planering	Kontur kompletterad med detaljering anpassad för skala 1:100.
3xx	Översikt-bygg	Kontur kompletterad med detaljering anpassad för skala 1:20 och 1:50.
4xx	Detalj	Detaljerad representation, avsedd för detaljritningar i skala 1:1, 1:5 och 1:10.
5xx	Skalberoende figur	Schematisk symbol avsedd för såväl översiktsritningar som schemor och uppställningar. Skalberoende figur innebär att definitionen anger grafik utskriven i en given skala, som definieras av attributet !TSKALA.
6xx	Betjäningsområden	Grafik som illustrerar utrymmesbehov för användning och service.
7xx	Anslutningar	Punkter och ytor för anslutning till olika tekniska system t. ex. vatten, ventilation, kraft, tele.
8xx	Montering/fixering	Punkter, linjer och ytor för montering eller fixering av komponenter.
9xx	(reserverad)	

1xx - 5xx utesluter varandra men 6xx - 8xx kan användas med en av 1xx - 5xx

4.4. Linjekaraktär (linjetyp)

Linjetyp beskrivs som tredje tecken i fältet Linjetyp i posten LGR, enligt nedanstående uppställning.

xx0	Osynlig	
xx1	Heldragen	-----
xx2	Prickad	-----
xx3	Streckad	--- --- --- --- ---
xx4	Punktstreckad	--- - --- - --- - ---
xx5	Dubbelpunktstreckad	--- - - --- - - ---
xx6-xx9	(reserverade)	
x80-x99	Användardefinierad	

Andra tecknet i posten LGR är reserverat för lagerdelning av komponenter. Standardvärde är 0.

4.5. Linjebredder

I första hand används linjebredderna 0.18, 0.35 och 0.7 mm. Linjebredd anges efter linjetypen i posten LGR, (exempel: LGR 301 0.18).

4.6. Färg

Färg för linjer beskrivs med ett 6-siffrigt tal som anger procentuell mättnadsgrad av färgerna rött, grönt och blått.

Röd	990000
Grön	009900
Blå	000099
Gul	999900
Cyan	009999
Magenta	990099
Vit	999999

Färgen anges efter linjebreddens i posten LGR. Exempel, linje för detaljeringsgrad 1:50, bredd 0.18 mm och färg röd:

LGR 301 0.18 990000

4.7. Typsnitt

Texthöjder skall följa serien 2.5, 3.5, 5, 7, 10 mm där 3.5 mm är normalhöjd för beskrivande text, och större texthöjder används för rubriker.

Cad-komponenter bör endast innehålla nedanstående standardtypsnitt:

SS-A	Texttyp A enl. SS 03.22 14
SS-B	Texttyp B enl. SS 03 33 14
HELV	Helvetica fylld
HELV-OUT	Helvetica outline

4.8. 3D-grafik

3D-grafik föregås av record MOD och har samma grafiska attribut som 2D linjegratik. Det finns tre typer av 3d-grafik, trådmodell (WIRE), ytmodell (SURFACE) och volymmodell (SOLID).

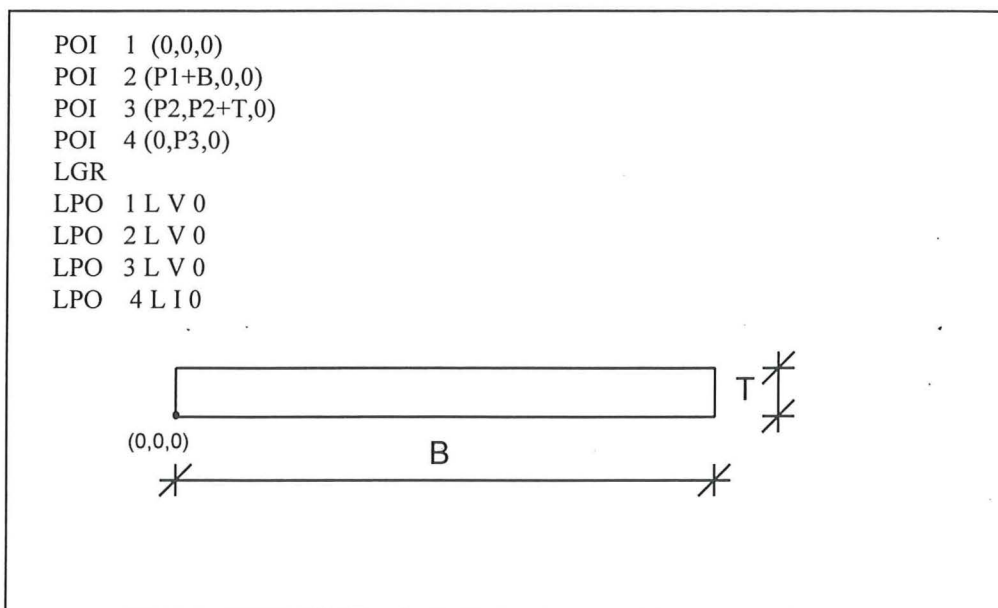
WIRE	Består av linjer mellan punkter i rymden och kallas för trådmodell. Den ger en enkel genomsynlig bild av komponenten.
SURFACE	Ytmodell som består av plana polygoner i rymden. Polygonens brytpunkter skall anges i ritning motsols, sedda från framsidan. Ytmodellen medger borttagning av skymda linjer, skuggning och materialbehandling.
SOLID	Volymmodell består av definierade volymer, vardera med ett bestämt antal beskrivande punkter. Varje volym anges med typ och ingående punkter. Volymmodellering medger addition och subtraktion av volymer, bestämning av snittytor m.m.

4.9. Parametrisk grafik

Parametrisk grafik har geometrin definierad med hjälp av variantattribut. Det används för komponenter som har samma form men har olika storlekar, t. ex. dörr som kan ha bredden 80 cm eller 90 cm.

I NICK-filen definieras punkter med posten POI som innehåller namn på punkten och koordinater. För att rita grafiken används därefter posten LPO,

som innehåller punktnummer i stället för koordinater i posten LCO (Line Coordinates), se kapitel 2.2. Datafil.



Figur 4.3. Exempel på parametrisk grafik.

4.10. Attribut.

I NICK-filen behövs minst två rader för varje attribut, en för definition av attributet och en för värdet på attributet.

Exempel:

```

ATT 1 1 1 !BREDD
VAL 200

```

Typgemensamma attribut beskrivs med posten ATT och tillhörande värderad VAL. Variantattribut beskrivs med posten VAT. Efter följer värderaden VAR. Värderaderna kan vara en eller fler.

4.11. Exempel på Cad-komponentfil

```

00001 HFI 01.0 930222 N-3.NIC Marbodal AB SAX Ny fil
00002 HPR MarboNick mm deg0.00 0.00 0.00 0.0000 SWE
00003 HCO Global definition av Marbodals symboler
00004 HSY Marbodals standardsymboler 9301
00005 HVO 0. 0. 0.
00006 DPO 0. 0. 0. 0.0
00007 DSY 3772 45 MARB N-3.NIC 01.0 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
00008 ATT 1 1 1 !BREDD
00009 VAL 200
00010 ATT 1 1 1 !DJUP

```

4. Rekommenderade egenskaper för Cad-komponenter

00011 VAL 580
00012 ATT 1 1 C !FABR
00013 VAL Marbodal AB
00014 ATT 1 1 I !HOJD
00015 VAL 870
00016 ATT 1 1 C !ID
00017 VAL B2-7-6-h1L
00018 ATT 1 1 C !SLAGN
00019 VAL V
00020 ATT 1 1 C MARBO-NR Marbodal beställningsnummer
00021 VAL 051802
00022 ATT 1 4 C FRI-SIDA Kod för fria sidor
00023 VAL 0F
00024 VAL VF
00025 VAL HF
00026 VAL 2F
00027 VIE BAKSIDA
00028 LGR 301 0.35 000000
00029 LCO 0. 0. 0. L I 0
00030 LCO 200. 0. 0. L V 0
00031 LCO 200. 0. 870. L V 0
00032 LCO 0. 0. 870. L V 0
00033 LCO 0. 0. 0. E V 0
00034 LGR 301 0.35 000000
00035 LCO 0. 0. 150. L I 0
00036 LCO 200. 0. 150. E V 0
00037 VIE VSIDA
00038 LGR 301 0.35 000000
00039 LCO 0. 0. 0. L I 0
00040 LCO 0. 0. 870. L V 0
00041 LCO 0. -580. 870. L V 0
00042 LCO 0. -580. 150. L V 0
00043 LCO 0. -500. 150. L V 0
00044 LCO 0. -500. 0. L V 0
00045 LCO 0. 0. 0. E V 0
00046 LGR 301 0.35 000000
00047 LCO 0. -580. 850. L I 0
00048 LCO 0. -600. 850. L V 0
00049 LCO 0. -600. 150. L V 0
00050 LCO 0. -580. 150. E V 0
00051 LGR 301 0.35 000000
00052 LCO 170. -600. 820. L I 0
00053 LCO 170. -630. 820. L V 0
00054 LCO 170. -630. 720. L V 0
00055 LCO 170. -600. 720. E V 0
00056 VIE HSIDA
00057 LGR 301 0.35 000000
00058 LCO 200. 0. 0. L I 0
00059 LCO 200. 0. 870. L V 0
00060 LCO 200. -580. 870. L V 0
00061 LCO 200. -580. 150. L V 0
00062 LCO 200. -500. 150. L V 0
00063 LCO 200. -500. 0. L V 0
00064 LCO 200. 0. 0. E V 0
00065 LGR 301 0.35 000000
00066 LCO 200. -580. 850. L I 0
00067 LCO 200. -600. 850. L V 0
00068 LCO 200. -600. 150. L V 0

4. Rekommenderade egenskaper för Cad-komponenter

00069	LCO	200.	-580.	150. E V	0
00070	LGR 301 0.35 000000				
00071	LCO	170.	-600.	820. L I	0
00072	LCO	170.	-630.	820. L V	0
00073	LCO	170.	-630.	720. L V	0
00074	LCO	170.	-600.	720. E V	0
00075	VIE TOPP				
00076	LGR 301 0.35 000000				
00077	LCO	0.	0.	870. L I	0
00078	LCO	200.	0.	870. L V	0
00079	LCO	200.	-600.	870. L V	0
00080	LCO	0.	-600.	870. L V	0
00081	LCO	0.	0.	870. E V	0
00082	LGR 303 0.35 000000				
00083	LCO	0.	-500.	150. L I	0
00084	LCO	200.	-500.	150. E V	0
00085	LGR 301 0.35 000000				
00086	LCO	170.	-600.	820 L I	0
00087	LCO	170.	-630.	820 E V	0
00088	VIE BOTTEN				
00089	LGR 301 0.35 000000				
00090	LCO	0.	0.	0. L I	0
00091	LCO	200.	0.	0. L V	0
00092	LCO	200.	-600.	0. L V	0
00093	LCO	0.	-600.	0. L V	0
00094	LCO	0.	0.	0. E V	0
00095	LGR 301 0.35 000000				
00096	LCO	0.	-500.	0. L I	0
00097	LCO	200.	-500.	0. E V	0
00098	LGR 301 0.35 000000				
00099	LCO	170.	-600.	820 L I	0
00100	LCO	170.	-630.	820 E V	0
00101	VIE FRONT				
00102	LGR 301 0.35 000000				
00103	LCO	0.	-600.	150. L I	0
00104	LCO	0.	-600.	870. L V	0
00105	LCO	200.	-600.	870. L V	0
00106	LCO	200.	-600.	150. L V	0
00107	LCO	0.	-600.	150. E V	0
00108	LGR 301 0.35 000000				
00109	LCO	0.	-500.	150. L I	0
00110	LCO	0.	-500.	0. L V	0
00111	LCO	200.	-500.	0. L V	0
00112	LCO	200.	-500.	150. E V	0
00113	LGR 301 0.35 000000				
00114	LCO	0.	-600.	850. L I	0
00115	LCO	200.	-600.	850. E V	0
00116	LGR 301 0.35 000000				
00117	LCO	170.	-630.	820. L I	0
00118	LCO	170.	-630.	720. E V	0
00119	HDR N-3				
00120	TXT 25. 3.5 20. 0010 NORM2			0.35 000000	
00121	VAL Handduksskåp vänster 20 cm				
00122	DPO	0.	0.	0.	0.0

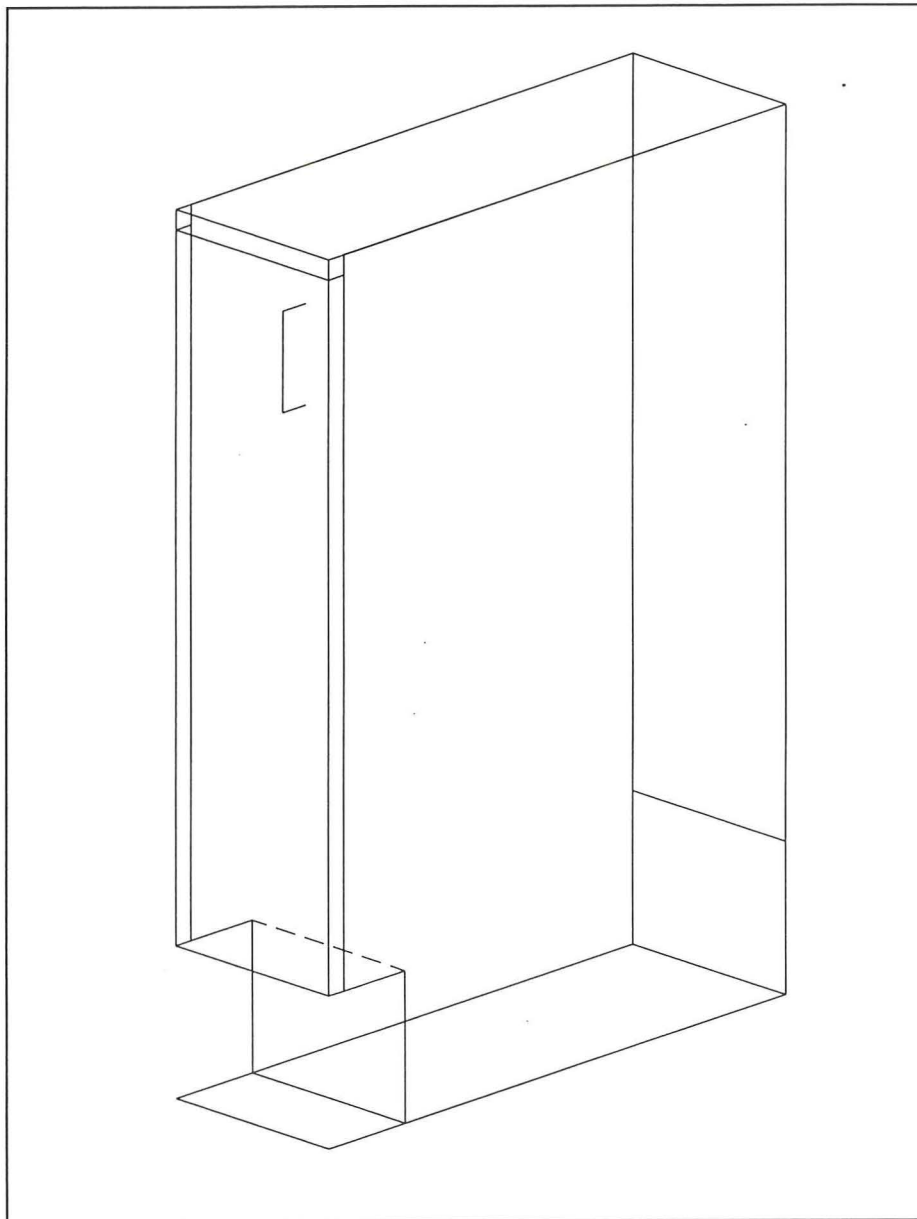
Rad 00001 - 00005 är administrativa record och inleder filen.

Rad 00006 - 00007 är definitionsrecord och definierar insättningspunkt och namn för komponenten.

Rad 00008 - 00026 är attributdel. Där alla attribut för komponenten, t. ex bredd, djup, fabrikant, id-nummer, m.m.

Rad 00027 - 00118 beskriver komponentens grafik för olika vyer.

Rad 00119 - 00122 är text som skal skrivas på ritningen.



Figur 4.4. Exempel på Cad-komponent.

5. Datoriserade varuinformationssystem.

5.1. Syfte med datoriserade varuinformationssystem.

Huvudanledningen till att datoriserade varuinformationssystem har börjat utvecklas är bl. a. att underlätta sökning av symboler i stora bibliotek och att göra 3D-modellering och visualisering så enkel och så snabb som möjligt. Hittills har det varit ganska komplicerat och tidskrävande att göra 3D-modellering och visualisering med fotorealismteknik och det har skrämt folk ifrån att använda den tekniken. Genom möjligheten till en kraftfull presentation av industriprodukter i en elektronisk katalog och direkt import av produktinformation, samt visualiseringsinformation i 3D blir det betydligt enklare att prova olika lösningar.

5.2. Kort beskrivning av några tillgängliga system.

Följande beskrivningar är tänkta att ge en uppfattning av vad för slags system som finns i användning eller är under utveckling, men inte att ge fullkomlig beskrivning av systemen.

5.2.1. Windows for Windows, demonstratör.

Windows for Windows [11] är elektronisk katalog som är gjord för visualisering av fönsterritningar och är tänkt ersätta en traditionell produktkatalog. Med programmet man kan alltid visa ny information och dessutom olika produktkombinationer. Namnet Windows for Windows har tillkommit för att programmet visar bilder av fönster (Windows) i Microsoft Windows miljö.

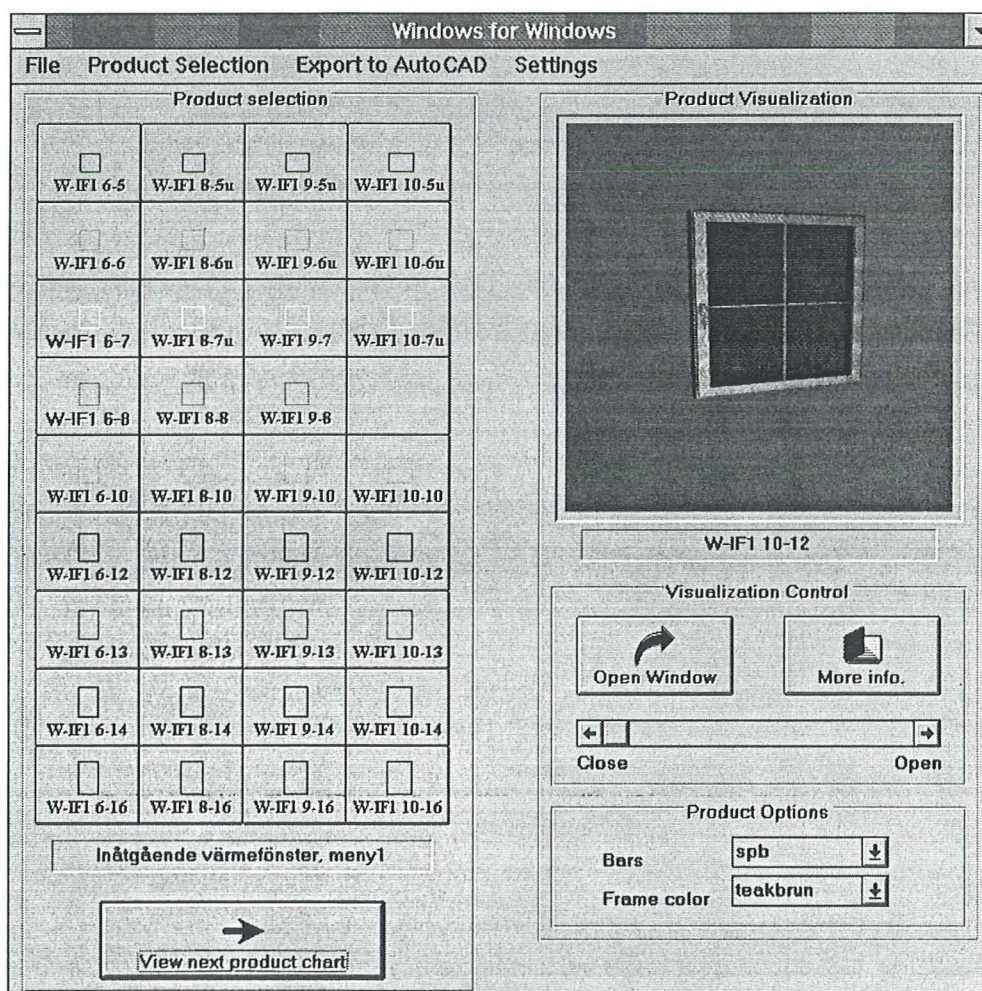
Programmet är utvecklad av Anders Follin vid Avdelningen för Bärande Konstruktioner, KBS-Media lab L.T.H. i Lund. Det är utvecklat i programspråket Visual Basic, med en koppling till dBase som är en databas där all information om fönstren är lagrade. För att klara av den kommunikationen behövdes det tredje systemet CodeBase som är ett programmeringsspråk gjort för kommunikation mellan Visual Basic och databasen.

Programfönstret i Windows for Windows består huvudsakligen av en del 'knappar' och en bildruta. Varje knapp står för en typgrupp fönster. När man trycker på en knapp kan man välja dimensioner på fönstret och sen ritas en

tre-dimensionell bild med fotorealistisk presentation av detta i bildrutan. Man kan också välja att se olika produktkombinationer. Genom att trycka på en knapp nedanför bildrutan kan man också se fönstrets mekaniska funktion, d.v.s. se hur det öppnas och stängs, och eventuellt andra rörliga detaljer.

Genom att öppna ett annat programfönster får man på skärmen teknisk information om fönstertypen i form av text och fönstrets uppbyggnad i form av en scannad bild.

Till sist är det möjligt att överföra information till Autocad för Windows i form av en 2-, eller 3-dimensionell ritning samt visualiseringsinformation (renderingsinformation). Detta sker med hjälp av DDE (Dynamic Data Exchange) som är Windows sätt att flytta information mellan två applikationer.



Figur 5.1. Windows for Windows, programfönster.

5.2.2. AMVI, demonstratör.

AMVI (Advanced Material and Vendor Information System) [10] är resultatet av samarbete mellan Svensk Byggtjänst i Stockholm och Avdelningen för Bärande Konstruktioner, L.T.H. i Lund, med Per Christiansson som projektledare.

Systemet är utvecklat i HyperCard för Macintosh med koppling till relationsdatabasen Oracle, rörliga videobilder och '(AI)-tools', Super Expert och MacBrain. Systemet är kört på Macintosh med koppling till bl. a. scanner, och videokamera.

I systemet kan man med hjälp av navigationspaletter söka efter önskad vara i ett bildbibliotek, visa bilder av varan på skärmen och sedan visa videobilder på monitorn som visar hur montering av varan går till, eller t. ex. en reparation. Biblioteket är en del av Byggvaruregiseret från Svensk Byggtjänst, som beskrivs i nästa kapitel, med extra information i form av ritningar och videobilder. Demonstratorn innehåller också material och prisinformation, och kan göra en utskrift för beställning.

5.2.3. Svensk Byggtjänst byggvaruregister.

Varudatabasen, Sveriges Bygg och Installationsvaror, är en PC-databas utarbetad av AB Svensk Byggtjänst. Utvecklingen av PC-versionen har skett i samarbete med AU System Communication AB.

Varudatabasen innehåller alla Sveriges bygg- och installationsvaror. Varubeskrivningen innefattar handelsnamn, beskrivning, eventuella referenser till Svensk Bygg- och VVS-katalog samt eventuella typgodkännanden från Boverket. För företagen redovisas adress, telefon- och faxnummer. Varudatabasen innehåller inga bilder.

Innehållet i Varudatabasen uppdateras och administreras av AB Svensk Byggtjänst. På startbilden 'Innehåll' finns en knapp för att aktivera Uppdatering av databasen. Kommandot Uppdatera kan endast användas i samband med att Du erhållit en uppdatering från AB Svensk Byggtjänst.

Du har möjlighet att söka information via 6 sökingångar; företag, varor, sökord, BSAB-kod, varugrupper och nyheter.

Utskrifter kan göras på godtyckligt vald fil, eller via kopiering till urklipp, av företagsrapport och varugrupsrapport. Företagsrapporten innehåller uppgift om ett företag samt alla deras varor. Varugrupsrapporten innehåller alla varor och företag som finns under respektive varugrupp.

Varudatabasen är förberedd för att använda direktfaxfunktionen eller utskrifter för att nå en leverantör eller tillverkare med frågor. Med programmet Winfax Pro och ett faxmodem kan Du faxa direkt från Varudatabasen till en leverantör eller tillverkare för att få ytterligare information. Har Du inte ett faxmodem och programmet Winfax Pro kan Du välja att skriva ut meddelandet.



Figur 5.2. Svensk byggtjänst varudatabas, huvudmeny.

6. Kommunikation emellan Cad-komponentfiler och datoriserade varuinformationssystem.

6.1. Syfte med kommunikationen.

Genom att utveckla ett elektroniskt varuinformationssystem som använder standardiserade indata av typen neutralt filformat, till exempel Cad-komponenter på NICK-format är det möjligt att bilda ett bibliotek med varor från olika tillverkare. Detta gör det enklare för arkitekter och konstruktörer att prova olika lösningar med produkter från olika tillverkare, och att ställa samman dessa på en ritning som även innehåller produktinformation i form av attribut. Genom att öka informationen i komponenterna, d.v.s. lägga in 3D-grafik, blir det relativt enkelt att ställa samman många komponenter i en visualiseringsmodell. Tillverkarna behöver då endast skicka ifrån sig en typ av komponenter, d.v.s. på det neutrala formatet som man bestämmer sig att kommunicera med, och konstruktören behöver endast ett system för sökning av komponenter och för att göra en ritning eller en visualiseringsmodell med produkter från olika tillverkare.

Elektroniskt varuinformationssystem är därför ett verktyg för att göra det enklare att söka i ett komponentbibliotek och ta fram de komponenter som behövs varje gång och lägga in dessa på en ritning som sedan kan användas för t. ex. mängdavgivning eller i en visualiseringsmodell.

6.2. Kravanalys.

I början var det meningen att försöka utveckla ett program som var ett gränssnitt mot Windows for Windows, men efter en diskussion med Anders Follin, som har utvecklat programmet Windows for Windows insåg jag att det inte var så enkelt att göra ett gränssnitt som kommunicerar mellan Cad-komponenter och Windows for Windows. Dessutom är Windows for Windows speciellt anpassat för visualisering av fönster, och alla variabler och strukturuppbyggnad i programmet anpassad till det. Det är också så att de Cad-komponenter som hittills har varit gjorda inte innehåller 3-dimensionell grafik och inte heller information om mekanisk funktion. Det är därför kanske inte så intressant med import till Windows for Windows eftersom där endast ritas en 3-dimensionell bild av produkten. Jag har därför beslutat att försöka göra ett eget program som utnyttjar de möjligheter som Cad-komponenterna bjuder på.

Det som ett sådan program skall kunna göra är t. ex. att:

- Enkelt kunna söka de komponenter som man behöver varje gång. Det finns två sätt att göra det på. Det ena är att använda en sökfild som innehåller en kort beskrivning av vad komponenten innehåller i form av text. Det andra är att söka efter filnamn som i många fall inte säger någonting om innehållet i komponenten.
- Rita figurer på skärmen av de vyer som komponenten innehåller, eventuellt också isometriska vyer och 3-dimensionella bilder.
- Rita måttsättningar av komponenten.
- Ta hand om attribut som tillhör komponenten och skriva dessa på skärmen.
- Ändra färg och eventuellt ytbehandling på komponenten.
- Klara av att exportera figurer med tillhörande färg, ytbehandling och attribut till något Cad-system, t. ex. Autocad.

Jag har via Svensk Byggtjänst fått en diskett med Cad-komponenter från 6 olika företag, 2-4 från varje, med undantag från Marbodal AB som skickat ifrån sig 173.

Det första man stöter på är frågan om vilket som är det bästa sättet att söka bland komponenter som man vill använda, särskilt i ett bibliotek innehållande flera hundra komponenter. Ett sätt är att ordna komponentfiler så att varje företag har sin katalog. För att söka inom katalogen har Marbodal AB gjort en sökfild som består av filnamn, ID-nummer och en klartext som beskriver innehållet i filen. Användning av sökfild av den typen kan underlätta sökning av komponenter i ett stort bibliotek. Namnet på sökfild kan lämpligen vara leverantörens NICK-kod, MARB för Marbodal, SWDO för Swedoor o.s.v. För att kunna söka komponenter i bibliotek där det inte finns någon sökfild måste man också kunna söka efter filnamn, men det har den nackdelen att filnamnet ofta inte säger någonting om innehållet i filen. Den metoden passar därför endast för bibliotek med få komponenter.

Filnamn	ID-nummer	Klartext
N-1	B2-7-6-q1L	Brickfack vänster 20 cm
N-2	B2-7-6-q1R	Brickfack höger 20 cm
N-3	B2-7-6-h1L	Handduksskåp vänster 20 cm
N-4	B2-7-6-h1R	Handduksskåp höger 20 cm
N-5	B3-7-6-q1L	Bänkskåp utan inrede vänster 30 cm
N-6	B3-7-6-q1R	Bänkskåp utan inrede höger 30 cm

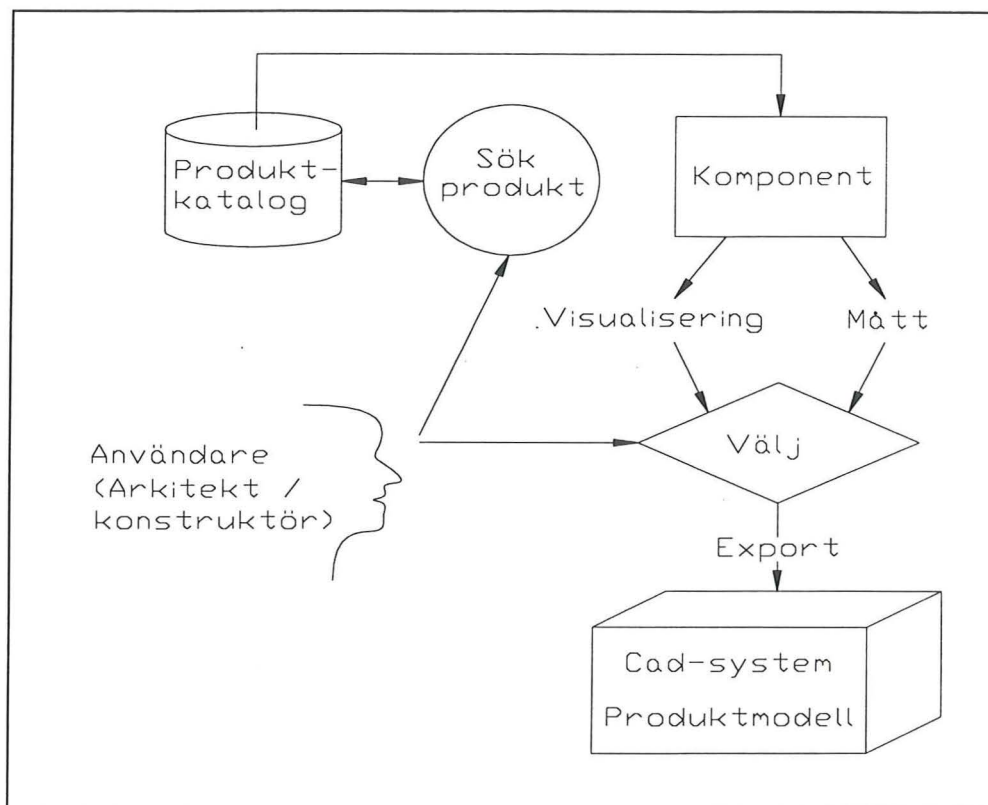
Figur 6.1. Exempel på sökfil (från Marbodal AB):

När man har hittat en komponent som skall användas är det troligen snabbast att ladda in hela filen i minnet på en gång och läsa igenom informationen och sortera efteråt. På det sättet slipper man vänta medan datorn läser från disk varje gång man gör något. Metoden har nackdelen att vara minneskrävande men jag tror inte att det är ett stort problem på en dator som kör Microsoft Windows.

Efter sortering av informationen kan attributlistan och annan text som tillhör komponenten, t. ex. symboldefinition och annan beskrivande text, skrivas på skärmen.

Nu är det klart för uppritning av figurer. Det är lämpligt att lägga in 'knappar' i programmet att trycka på, en för varje vy, och en för att ändra färg på ritningen. På det sättet kan man visa önskad vy så ofta man vill. När figuren av vyn ritas är det lämpligt att också rita måttsättning på komponentens största utsträckning i det plan som vyn visar. Man kan ha nytta av att se måtten på komponenten när man försöker välja mellan flera komponenter.

Färg och ytbehandling är en viktig visualiseringsinformation, och det är en fördel att kunna pröva olika färger och ytbehandlingar i varuinformationssystemet, som sedan skickas över till Cad-systemet med komponenten. Detta är mer intressant att göra för 3D-bilder än för 2D-bilder. 2D-bilder används inte ofta i samband med visualisering.



Figur 6.2. Att välja produkt i en elektroniskt lagrad varuinformationsdatabas.

Nästa steg är att exportera figuren till något Cad-system. Jag utgår här ifrån Autocad, men man kan tänka sig att senare lägga till möjlighet att exportera till något annat system. Det finns några sätt att arbeta med Autocad som ger olika alternativ för att exportera symboler från varuinformationssystem.

1. När man arbetar med Autocad för Windows är det möjligt att samtidigt köra Autocad och något varuinformationssystem och därför kan man använda DDE (Dynamic Data Exchange) i Windows för att överföra symbolen till Autocad.
2. När man arbetar med Autocad för Dos behöver man ofta starta om datorn för att gå mellan Autocad och Windows och därför är det inte möjligt att starta Autocad från varuinformationssystemet för att direkt lägga in symbolen på en ritning. Här kan man i varuinformationssystemet skapa en fil som man kan kalla för projektfil, d.v.s. en fil som innehåller namn och eventuellt annan information på en eller flera symboler som skall läggas in på en ritning. Man måste då göra ett Lisp-program i Autocad som läser in projektfilen, och ritlar de symboler som man har valt ut. För att göra Lisp-programmeringen enklare kan man också i projektfilen lägga tillhörande grafik på något reducerad sätt.

3. En möjlighet är att använda ett operativsystem där man kan köra många program samtidigt, t. ex. Windows NT eller OS/2. Det gör det möjligt att köra Autocad i ett fönster och Elvar i ett annat samtidigt. På det sättet slipper man gå ur ena programmet och starta det andra. Arbetsgången blir densamma som i alternativ 2.

6.3. Begränsningar i Cad-komponentformatet.

Jag har fått en del problem med Cad-komponentfilerna, och stött på några begränsningar och problem i praktisk användning av komponenterna. I detta kapitel skall jag försöka peka på några av dessa problem och ge förslag till förbättringar.

Cad-komponenterna har alla samma uppbyggnad. De flesta innehåller 6 vyer, framsida, baksida, topp, botten, vänster och höger. Inga innehåller grafik för 3-dimensionell ritning. Därmed saknas en intressant del i ett varuinformationssystem.

Alla komponenterna innehåller en del attribut som beskriver olika sorters produktinformation. Det är viktigt att denna information följer med i överföringen från komponenten och hela vägen till Cad-systemet. Ett problem med attributlistan är att den inte alltid följer rekommendationerna för Cad-komponenter. Där säges att i standardattribut skall namnet inte börja på ett utropstecken (!) men det är inte i överensstämmelse mellan komponenter från olika företag. Attributnamnet skrivs också ibland med svenska bokstäver och ibland inte. Till exempel attributet höjd skrivs ibland som HOJD och ibland som HÖJD. Listan för standardattribut blir enklare att arbeta med om man bestämmer sig för om svenska bokstäver skall användas eller inte.

Programmet 'Windows for Windows' kan visa mekanisk funktion hos produkten, men i Cad-komponenterna saknas den information för att kunna göra detta. Man kan tänka sig att det är möjligt att lägga in information av den typen i komponenten, men det är mest intressant för visning av varan i en elektronisk katalog, men mindre intressant för att importera till en ritning.

I några filer saknades "end of line". Det är viktigt att detta inte inträffar för att man skall kunna använda funktion av typen Read-line i ett program. I Autolisp är det till exempel enda enkla sättet att läsa information från en fil.

Komponenterna är gjorda i Dos på ASCII-format vilket kan leda till problem när man arbetar i Windows, eftersom Dos och Windows inte använder samma teckentabell. Det leder till att svenska bokstäver som ä, ö och å ändras till andra bokstäver eller symboler för att de inte är placerade på samma ställe i teckentabellen för Dos och Windows. Det är också en skillnad mellan standard ASCII teckentabell, som en del använder, och den teckentabell som heter PC-850, vilken används på flesta datorer som levereras i Sverige i dag.

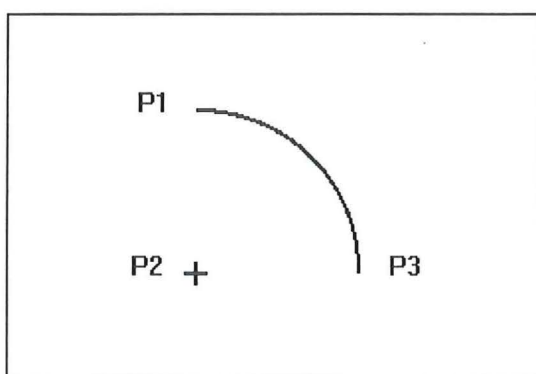
Cirkelbågar i komponenterna beskrivs av startpunkt, mittpunkt, och slutpunkt. Det är inte alltid samma avstånd från mittpunkten till startpunkten och slutpunkten, d.v.s. radien är inte den samma. Detta leder till att cirkelbågen inte alltid tangerar linjerna i början och slutet av cirkelbågen som den är tänkt göra. Man kan tänka sig att de beskriver en ellips, men det är ett av de fall som NICK-standarden inte hanterar.

Enligt rekommendationen för Cad-komponenter skall man beskriva cirkelbågar på följande sätt, se också kapitel 2.2. Datafil.

```
00022 LCO      0      0      0 C V 0  P1
00023 LCO      0     10     0 + I 0  P2
00024 LCO     10     10     0 E I 0  P3
```

Det används inte samma sätt för att beskriva cirkelbågarna i praktiken. Det är gjort på följande sätt, d.v.s. L, C, +, och E har olika betydelse.

```
00022 LCO      0      0      0 L I 0  P1
00023 LCO      0     10     0 C V 0  P2
00024 LCO     10     10     0 + V 0  P3
```



Figur 6.3. Cirkelbåge.

Det är inte heller överensstämmelse mellan rekommendationen och praktiken vad gäller beskrivning av linjer.

Det blir enligt rekommendationen:

```
00022 LCO    0    0    0 L V 0
00023 LCO    0   10    0 E I 0
```

Och i praktiken:

```
00022 LCO    0    0    0 L I 0
00023 LCO    0   10    0 E V 0
```

Det är enklare att göra program som använder Cad-komponenter om de överensstämmer med rekommendationen, och att man inte behöver tänka på att komponenterna kan skilja sig ifrån den på olika sätt.

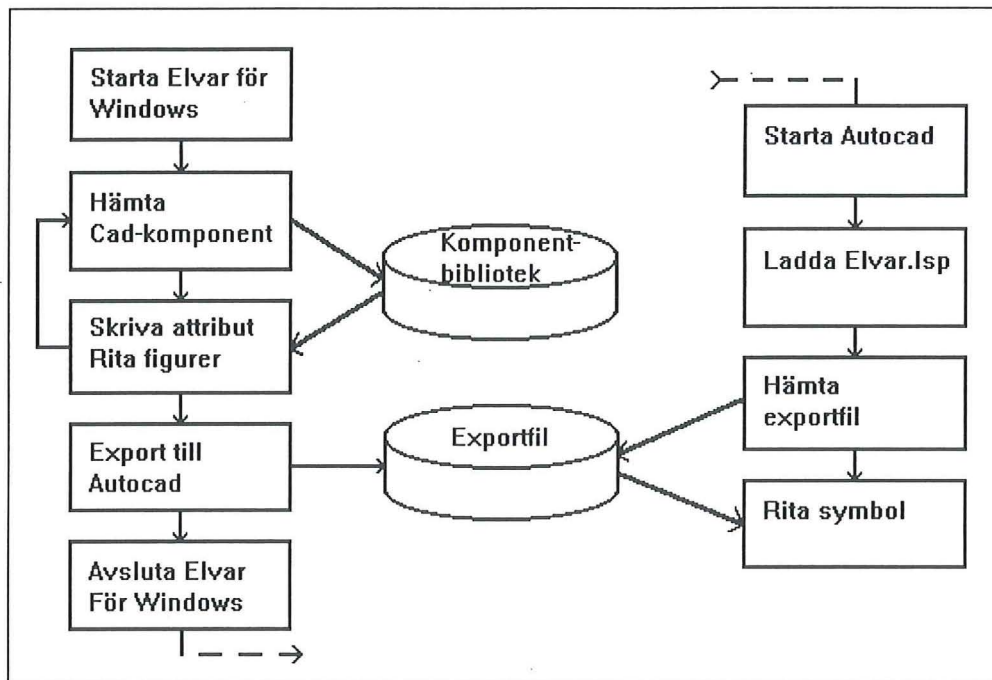
Det är möjligt att dessa begränsningar och fel har blivit rättade, men om inte så är fallet är det viktigt att det blir gjort, för om det uppstår massor med fel i komponenterna blir det till slut ingen som använder sig av dessa.

7. Uppbyggnad av gränssnittsprogrammet ELVAR.

7.1. Inledning

Programmet Elvar är tänkt som ett hjälpmedel för att söka och välja mellan Cad-komponenter i ett komponentbibliotek och att göra en exportfil till Autocad. Programmet kan delas i två delar, den ena är ett Windows-program utvecklad i Microsoft Visual Basic, och den andra är ett Autolisp-program. Windows-programmet tar hand om sökning i ett komponentbibliotek, visning av Cad-komponenterna på skärm och gör en exportfil. Autolisp-programmet körs inuti Autocad och importerar exportfilen till Autocad.

Ritningar av 3-dimensionella figurer och möjligheten att ändra färg på komponenten har lämnats kvar för eventuell senare utveckling. Detta har jag valt för att Cad-komponenterna fortfarande inte innehåller någon 3d grafik. Dessutom är färg av mindre intresse för 2d ritningar än 3d ritningar som är tänkta för överföring till visualiseringsmodeller.



Figur 7.1. Flödesschema för programmet Elvar.

7.2. Sökning av Cad-komponenter

Cad-komponenter kan sökas på två sätt, efter filnamn och genom sökning i sökfil.

7.2.1. Sökning efter filnamn:

Vid sökning efter filnamn används vanligt File Open fönster där man använder funktionerna *Drive1.Drive*, *Dir1.Path* och *File1.FileName* för att kontrollera vilken enhet, katalog och fil som är markerad. När man har markerad den fil som man vill läsa trycker man på *OK*-knappen och filnamnet och tillhörande katalog skickas till proceduren *Komppopen*, som beskrivs i kapitel 7.2.2.

7.2.2. Sökning i sökfil

Vid sökning i sökfil används möjligheten att koppla tabeller från Microsoft Access direkt till Visual Basic. Jag gjorde ett antal tabeller i Microsoft Access. En innehåller namn på leverantörerna och tillhörande NICK-kod och heter *Koder*. Andra innehåller information om Cad-komponenter från respektive leverantörer, d.v.s. sökfilen som beskrivs i kapitel 6.2. och har samma namn som leverantörens NICK-kod.

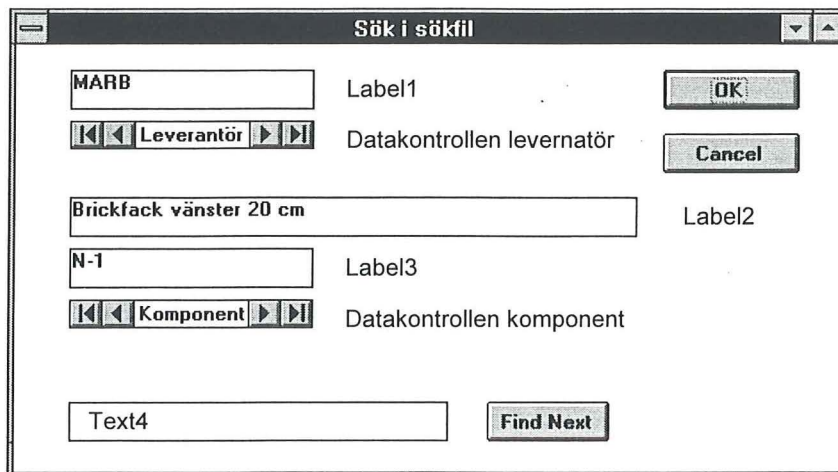
Kod	Leverantör
ECOP	
GYPR	Gyproc
IFSA	Ifö
MARB	Marbodal
SWDO	Swedoor
TEPR	Tepro

Figur 7.2. Databastabellen Koder.

ID1	Filname	ID	Klartext
1	N-1	B2-7-6-q1L	Brickfack vänster 20 cm
2	N-2	B2-7-6-q1R	Brickfack höger 20 cm
3	N-3	B2-7-6-h1L	Handduksskåp vänster 20 cm
4	N-4	B2-7-6-h1R	Handduksskåp höger 20 cm
5	N-5	B3-7-6-q1L	Bänkskåp utan inrede vänster 30 cm

Figur 7.3. Databastabellen MARB, (första 5 raderna).

I programmet (i Visual Basic) gjorde jag ett nytt fönster som innehåller ett antal 'label', två databaskontroller, textruta och knapp, se figur 7.4. Skillnaden på 'label' och textruta är att det är möjligt att editera text som står i textruta men inte text som står i 'label'.



Figur 7.4. Sökning i sökfil fönstret.

Databaskontrollen är kopplad till en databas med egenskapen *DataBaseName* och till en enskild tabell i databasen med egenskapen *RecordSource*. I det här fallet är databaskontrollen *Leverantör* kopplad till tabellen *Koder* (figur 7.2.) i databasen och databaskontrollen *Komponent* kopplad till den leverantör som är markerad i tabellen *koder*. D.v.s. genom att ändra leverantör med databaskontrollen *Leverantör* ändras automatiskt tabellkopplingen för databaskontrollen *Komponent*, se figur 7.4.

Med egenskaperna *DataSource* och *DataField* kan man koppla 'label' till en databas. Det gör man genom att egenskapen *DataSource* blir namn på den datakontroll som man vill koppla 'labelen' till och *DataField* blir namnet på det fält i databasen som man vill skriva i 'labelrutan'. Här blir *Label1.DataSource* datakontrollen *Leverantör* (figur 7.4.), *Label1.DataField* blir fältet *Kod* (figur 7.2.), *Label2.DataSource* blir datakontrollen *Komponent* (figur 7.4.), *Label2.DataField* blir fältet *Klartext* (figur 7.3.), *Label3.DataSource* blir också datakontrollen *Komponent* och *Label3.DataField* blir fältet *Filname* (figur 7.3.).

För sökning i sökfil-tabellerna används en funktion som heter *FindNext* och om den når änden av tabellen och inte hittar rätt sträng flyttas den till början med funktionen *MoveFirst*. I sökrutinen måste man göra en sträng som innehåller namn på det fält i vilket sökning skall ske, *LIKE* och den text man skall söka efter, till exempel *"Klartext LIKE " + Text4.Text + "*" "*, där

Klartext är namn på fältet i databasen där sökningen skall ske (figur 7.3.), *LIKE* är funktionen som jämför, *Text4.Text* är den text som användaren skriver i textrutan *Text4* (figur 7.4.) och '*' står för att det får vara vad som helst efter den text som användaren har skrivit in, d.v.s. användaren behöver bara skriva in en bokstav och programmet letar efter nästa *Klartext* (figur 7.3.) som börjar på den bokstaven.

När användaren har bestämt sig för att se en bild på någon komponent trycker han på *OK*-knappen och texten som står i 'label3' (se figur 7.4), i fältet filnamn i databasen, skickas till proceduren *Kompopen* samt information om tillhörande katalog.

7.3. Läsning av Cad-komponentfil och sortering av informationen.

Inläsning av Cad-komponentfilen sker i proceduren *Kompopen*. Där definierar jag ett vektorfält, *Infil*, som kan innehålla som mest 1200 element. I vektorfältet *Infil* läser programmet in filen med *Read-Line* sats. D.v.s. programmet läser en linje i filen som ett element i vektorfältet.

Vid sortering av informationen har jag i Visual Basic bildat strukturer av olika slag och här följer beskrivning av dessa samt deklARATIONERNA från programmet.

En strukturtyp innehåller symboldefinitionslinjen i Cad-komponentfilen.

```
Type Symdef
  BDKod As Integer
  UtökadKod As Integer
  Formtyp As Integer
  LevKod As String
  SymbNamn As String
  Ver As String
  Text As String
End Type
```

En typ innehåller attribut och består dels av beskrivning av attributet, format och namn, och dels av attributens text. Högsta antal textrader är 10.

```
Type Attribut
  Grpäv As Integer
  AntRad As Integer
  Format As String
  Namn As String
  Text As String
  Värde(10) As String
End Type
```

En typ innehåller en punkt och består av punktkoordinater (x,y,z), information om vad skall ritas mellan punkterna och hur punkten skall se ut.

```
Type Lcoord
  x As Single
  y As Single
  z As Single
  Förbin As String
  synl As String
  Pkttyp As Integer
End Type
```

En typ innehåller en polygon och består av inledning, som är information om linjen, typ, tjocklek och färg, och ett antal punkter, högst 20.

```
Type lgraph
  Linetyp As Integer
  Tjocklek As Single
  Färg As String
  Punkt(20) As Lcoord
End Type
```

En typ innehåller text som skall skrivas på ritningen och består av textens utseende, själva texten, insättningspunkten och rotation.

```
Type Text
  Bredd As Single
  Höjd As Single
  Skala As Single
  Just As String
  Font As String
  Tjockl As Single
  Färg As String
  Text As String
  x As Single
  y As Single
  z As Single
  Rot As Single
End Type
```

Vektorfältet *Infil* är definierat lokalt, vilket gör att när inläsningen och sorteringen är klar och programmet går ut ur proceduren *Kompopen* vektorfältet försvinner ur datorns minne.

Strukturerna som beskrives ovan är däremot definierade *Global*, d.v.s. de kan anropas var som helst och när som helst i programmet.

7.4. Skrivning av attribut och annan text i textrutorna.

När proceduren *Kompopen* är färdig med att sortera informationen anropas en annan procedur som heter *Attribtxt*. Den tar hand om att skriva attribut och annan text i textrutor på skärmen. Den skriver attributnamn eller beskrivande text om den finns definierad på en rad och attributens värde på en annan rad.

För att kunna skriva texten i en textruta måste man konvertera den till en sträng, och för att flytta texten till en ny rad måste man definiera en konstant som jag har kallat *NL* ($NL = Chr\$(13) + Chr\(10)). $Chr\$(13)$ flyttar texten till en ny rad och $Chr\$(10)$ flyttar texten främst i raden.

När programmet har gjort en sträng av attributtextern och en sträng av den beskrivande texten skickas den till en annan procedur, som heter *SvTecken*, som läser i genom texten och sätter svenska bokstäver på rätt plats. När detta är färdigt skickas texten tillbaka till *Attribtxt* som skriver den i textrutorna på skärmen (figur 7.11.).

7.5. Uppritning av figurer.

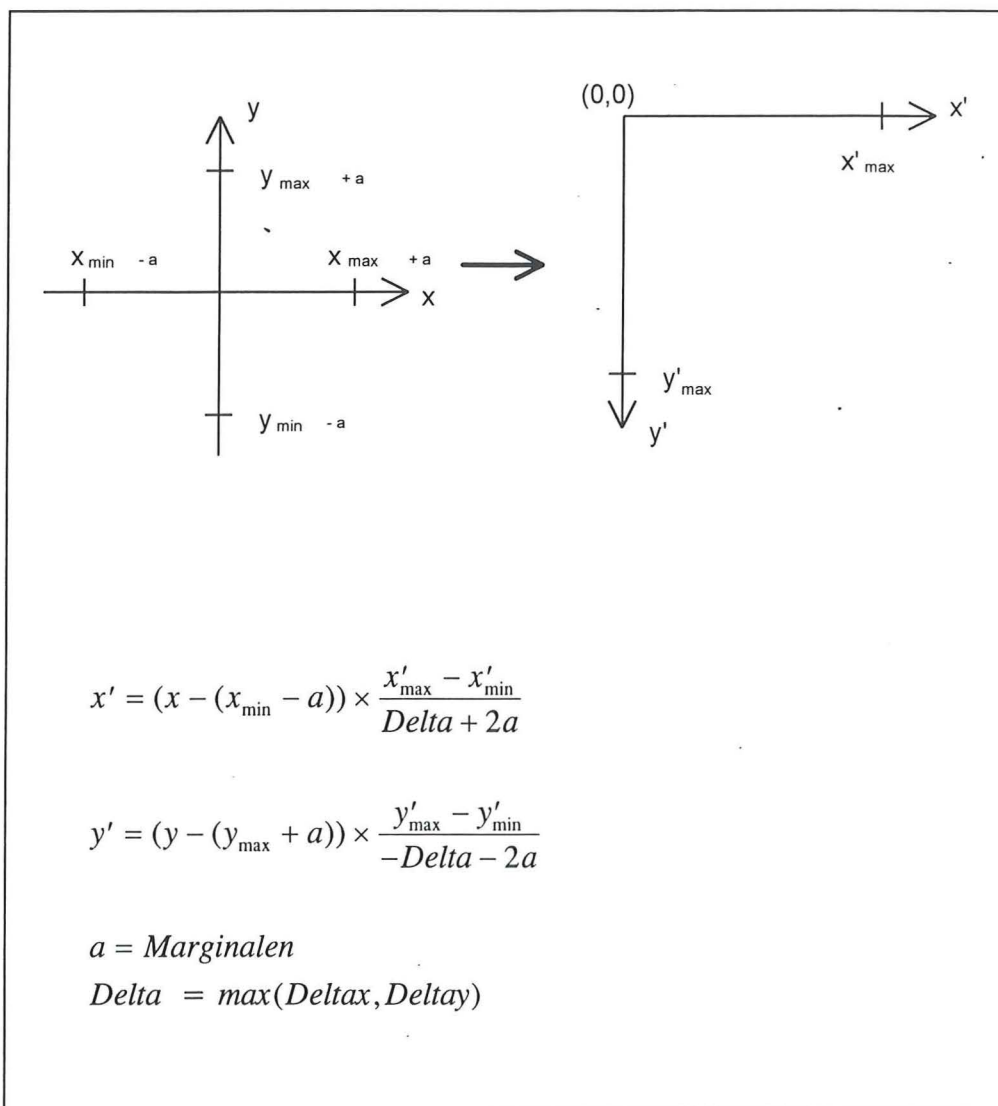
Uppritning av figurer sker i procedurer som heter *XYAPI*, *XZAPI* och *YZAPI*. Namnen står för att procedurerna använder Windows API-rutiner (Application Programming Interface) för att rita figurerna och *xy* står för att proceduren ritar i *xy*-planet, *xz* ritar i *xz*-planet, och *yz* ritar i *yz*-planet. Alla procedurerna är uppbyggda på samma sätt, skillnaden är att de inte arbetar med samma koordinatpar (*xy*, *xz* eller *yz*).

I följande beskrivning utgår jag ifrån *xy*-planet, men samma sak gäller för *xz*-planet och *yz*-planet.

Inparametrar i procedurerna är *Sida*, som är en vektor av typen *lgraph* (se 7.3.) och innehåller den vy som skall ritas, *Fj* som är heltal (integer) och representerar antalet polygoner i aktuell vy och *Sp* som är heltal (integer) och anger om bilden skall speglas ($Sp = \div 1$) eller inte ($Sp = 1$), d.v.s. från vilket håll man tittar på bilden. Variabeln *Sp* är 1 för toppvy, framsida och höger sida och $\div 1$ för botten, baksida och vänster sida.

API-rutinerna arbetar i pixel mode d.v.s. en punkt på skärmen är en enhet i koordinatsystemet, och dess origo är i övre vänstra hörnet på skärmen. Därför måste man överföra alla koordinater till skärmens koordinatsystem. Innan man börjar måste man ta reda på min. och max. värden för *x* och *y*, delta-*x*,

och delta-y som är skillnaden på max och min värdet, och marginalen a , som jag har satt till 20% av större deltavärdet. Nu har man alla ingångsparametrar i överföringsformlerna.



Figur 7.5. Cad-komponentens respektive skärmens koordinatsystem och överföringsformler.

För att ta reda på x'_{\max} och y'_{\max} används funktionen *GetClientRect* som ger max och min koordinater för den form eller bildruta som hänvisas till.

Nästa steg är att ställa in rätt linjetyp. Det görs genom att ändra egenskapen *DrawStyle* för bildrutan (*Form1.Picture1.DrawStyle = A*). Fältet *LineTyp* i strukturen *Sida* innehåller den informationen (*Sida(i).LineTyp*).

Nu är det klart att rita figuren och där måste man börja med att kontrollera om det är en linje eller cirkelbåge som skall ritas. Det gör man genom att

kolla fältet 'förbindelse' i strukturen *Sida (Sida(i).Punkt(j).forbin)*. För att rita linjer används API-funktionen *PolyLine* och för att rita cirklar används API-funktionen *Arc*. Funktionen *Polyline* har som indata *hDC* som är "handtag" till den bildruta som man vill rita i, *Points()*, som är strukturfält som innehåller x och y koordinater för polygonen som skall ritas, och n som är antalet punkter i fältet *Points()*. Funktionen *Arc* har som indata handtaget *hDC*, koordinater för övre vänstra hörnet och nedre högra hornet av den rektangel som cirkelbågen skall ritas inuti och start och slutpunkt för cirkelbågen. Cirkelbågen ritas alltid motsols och därför måste man vända start och slutpunkt om man behöver rita medsols.

Därefter ritas måttsättningslinjer för komponentens största utsträckning. Där används också funktionen *PolyLine* för att rita linjerna och API-funktionen *DrawText* för att skriva måtten på skärmen. Funktionen *DrawText* fungerar på samma sätt som *Arc*, d.v.s. man anger den rektangel som texten skall skrivas inuti. För att räkna ut måtten används x_{\max} , x_{\min} , y_{\max} och y_{\min} som blivit uträknade tidigare i proceduren.

7.6. Export till Autocad

I proceduren *Export* skrivs en reducerad fil på disken som innehåller Cad-komponentens attribut och grafik tillhörande den vy som skall exporteras. Detta är gjort för att göra Autolisp-programmet, som läser komponenten till Autocad enklare. Formatet på exportfilen följer ungefär NICK-formatet.

För att exportera filen öppnar man ett vanligt File Save fönster, som använder samma funktioner som är beskrivna i kapitel 7.2.1., sök i sökfäl. När man väljer filnamn har man två möjligheter, dels att skapa en ny fil och dels att välja en fil som existerar och göra en fortsättning på den. Det sker automatiskt, när man använder kommandot *Append* i stället för *Output* när filen är öppen (*Open "filnamn" For Append As #1*), att programmet skriver fortsättning på filen, men inte över den information som finns i filen.

För att formatet på utskriften skall bli rätt använder man funktionen *Format\$* som är till för att skriva textsträngar på skärm, skrivare eller i fil på ett definierat sätt.

```
00001 DSY 3772 0 45 MARB N-8 01.0 xxxxxxxxxxxxxxxxxxxxxxxxxxxx
00002 ATT 1 1 I !BREDD Bredd
00003 VAL 400
00004 ATT 1 1 I !DJUP Djup
00005 VAL 580
```

```
00006 ATT 1 1 C !FABR Fabrikant
00007 VAL Marbodal AB
00008 ATT 1 1 I !HOJD Höjd
00009 VAL 870
00010 ATT 1 1 C !ID ID-nummer
00011 VAL B4-7-6-q1R
00012 ATT 1 1 C !SLAGN Dörrslagning
00013 VAL H
00014 ATT 1 1 C MARBO-NR Marbodal best,,llningsnummer
00015 VAL 052104
00016 ATT 1 4 C.FRI-SIDA Kod f'r fria sidor
00017 VAL 0F
00018 VAL VF
00019 VAL HF
00020 VAL 2F
00021 LGR 301 0,35 000000
00022 LCO 0 0 870 L I
00023 LCO 400 0 870 L V
00024 LCO 400 -600 870 L V
00025 LCO 0 -600 870 L V
00026 LCO 0 0 870 E V
00027 LGR 303 0,35 000000
00028 LCO 0 -500 150 L I
00029 LCO 400 -500 150 E V
00030 LGR 301 0,35 000000
00031 LCO 30 -600 820 L I
00032 LCO 30 -630 820 E V
```

Figur 7.6. Exempel på exportfil.

7.7. Import av Cad-komponent till Autocad (lisp-programmet Elvar).

För att importera Cad-komponenterna till Autocad från programmet Elvar har jag gjort ett program i Autolisp. Det läser exportfilen, gör ett block av symbolen med tillhörande attribut, varefter användaren placerar blocken på sin plats i ritningen på vanligt sätt.

Programmet börjar med att läsa exportfilen till en lista som heter *infil*, en rad i filen som ett element i listan. Sista elementet i listan får värdet *nil* innan inläsningen avslutas.

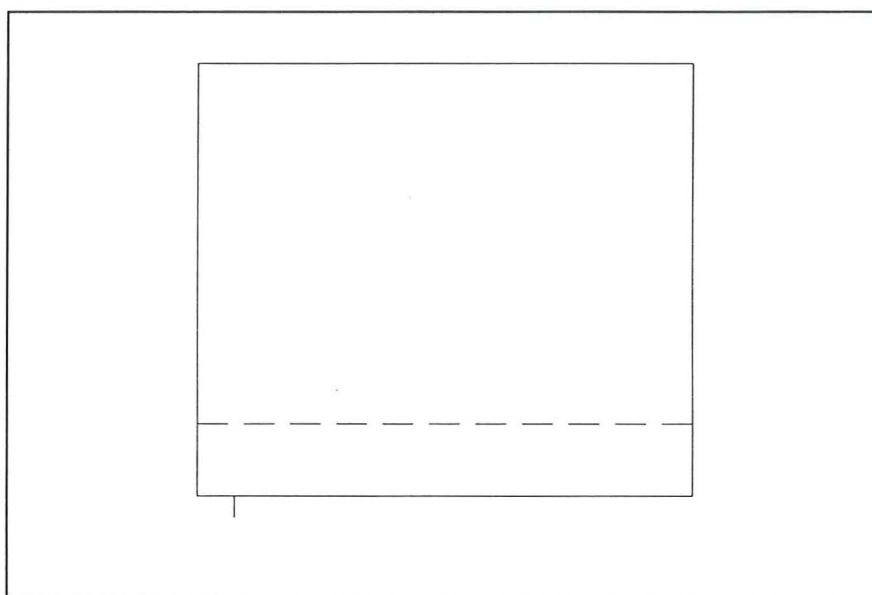
Resten av programmet läser igenom listan, kontrollerar vilken information som finns i respektive rad och använder denna på rätt sätt.

Om raden är av typer ATT som står för attribut anropas Autocad-kommandot ATTDEF och attributet får det namn och värde som står i nästa rad. Om det finns fler än en värderad som tillhör ett attribut skrivs de på ett attribut med komma (,) emellan. Alla attributen är *Invisible* och *Present*, som betyder att de visas inte på ritningen och att Autocad inte frågar efter nytt värde på

attributet när blocken är placerad på ritningen, men att det är möjligt att ändra värden senare, t.ex. med kommandot *DDATTE*.

Om raden är av typen LGR börjar programmet med att ställa in rätt linjetyp. Detta görs i ett underprogram som heter *linjetyp*. Sedan går det till nästa rad och kontrollerar om det skall rita linje eller cirkelbåge (förbindelse i NICK-filen), lagrar i minnet startpunkt och slutpunkt för linje och startpunkt, centerpunkt, och slutpunkt för cirkelbåge och använder sedan *PLINE* eller *ARC* för att rita elementet. Varje gång en punkt lagras i minnet går programmet till en underprogram *Maxmin* som håller reda på största och minsta koordinaterna.

När ritning av symbolen är avslutad, d.v.s. programmet har hittat *nil* i listan infil, gör programmet ett block av symbolen. Det görs med kommandot *BLOCK WINDOW* och använder min- och max- koordinaterna från underprogrammet *Maxmin* som hörnpunkter för fönstret. Som namn på blocket används symbolnamnet i symboldefinitionslinjen i Cad-komponentfilen. Med funktionen *tblsearch* kontrollerar programmet om namnet redan finns på ritningen, och i så fall får man chansen att ge blocket nytt namn eller avsluta programmet. Insättningspunkten för blocken är punkten (0,0,0) i Cad-komponentens lokala koordinater. Funktionen *pause* används inuti kommandot *INSERT* för att användaren skall kunna markera insättningspunkten och rotationen med musen och samtidigt se blocken.

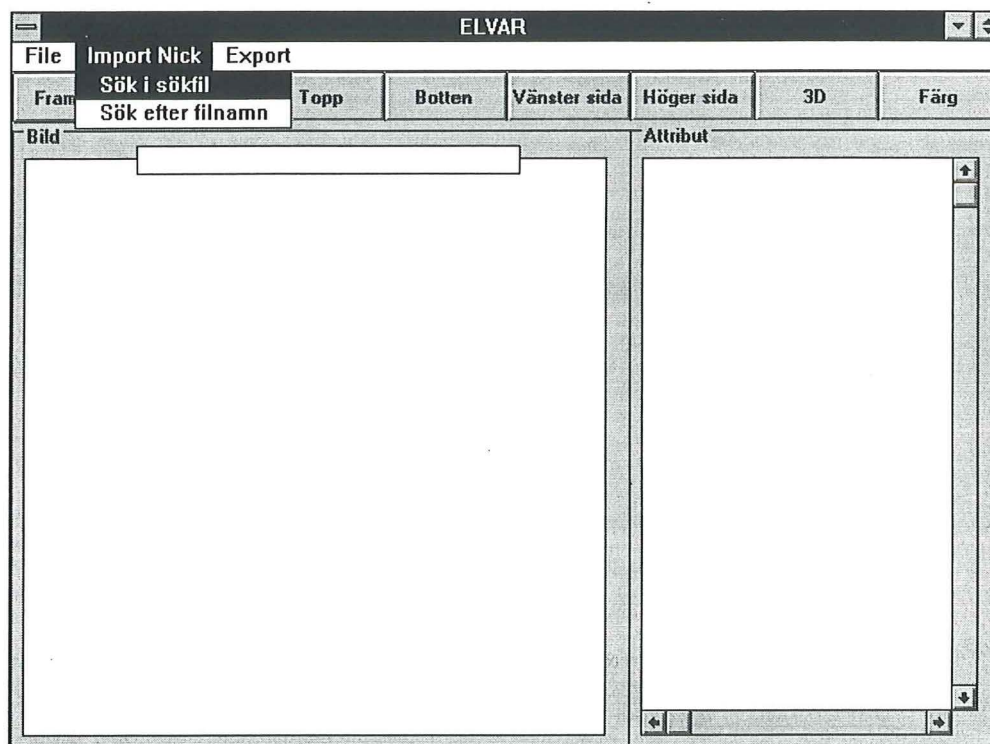


Figur 7.7. Exemplet från kapitel 7.6.

8. Användning av programmet ELVAR.

8.1. Sökning.

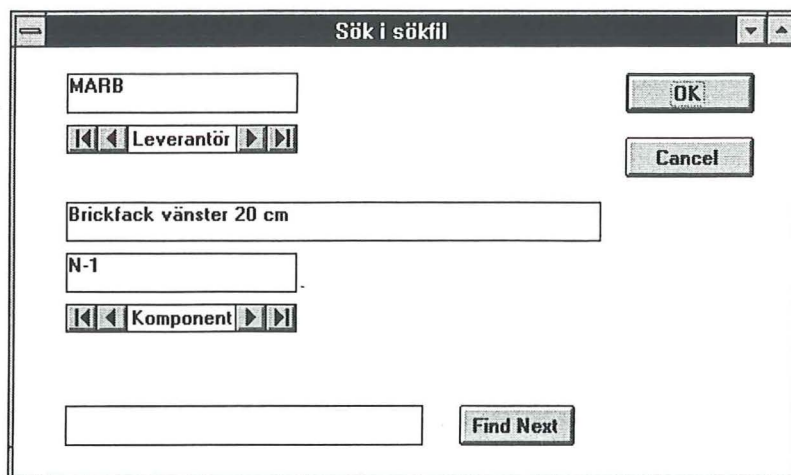
Man har möjlighet att söka komponenter på två sätt, söka efter filnamn eller använda sökfil som ligger inne i en speciell databas (se figur 7.2. och figur 7.3., kapitel 7.2.2.). Sökningen börjar med att man väljer *Import Nick* menyn i Elvars huvudfönster. Där får man två alternativ, *Sök efter filnamn* och *Sök i sökfil* (se figur 8.1.).



Figur 8.1. Elvars huvudfönster och Import Nick menyn.

8.1.1. Sök i sökfil:

När man väljer Sök i sökfil i Import Nick menyn aktiveras ett nytt fönster som visar namn på leverantör och information om tillhörande Cad-komponenter (se figur 8.2.).



Figur 8.2. Sök i sökfil fönstret.

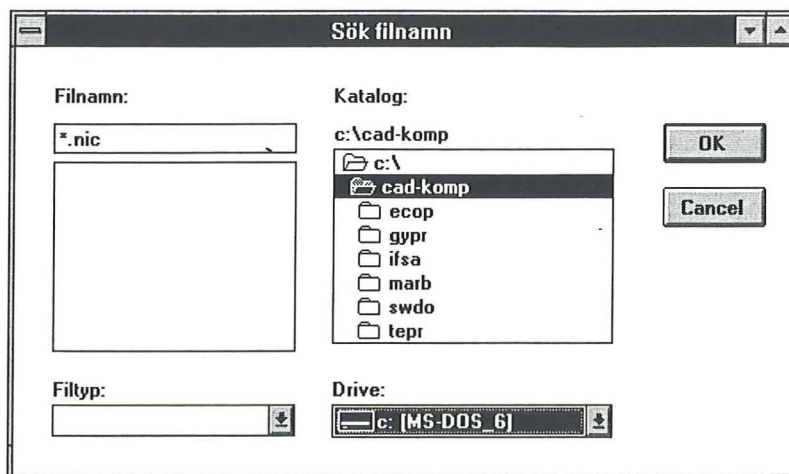
Där måste man börja med att välja leverantör. För att göra det använder man de övre pilarna. Om man trycker på en pil med streck visar programmet det sista elementet i databasfilen för pil till höger respektive första elementet för pil till vänster. Om man trycker på pil utan streck visar programmet nästa element.

När man har valt leverantör, får man i textrutan nedanför information om en komponent och dess filnamn. I början är det första komponenten i sökfilen som tillhör den leverantör som man har valt. För att välja en annan kan man bläddra i sökfilen genom att använda de nedre pilarna, som fungerar på samma sätt som beskrivits ovan, eller använda sökningen "Find Next". Där skriver man in en eller fler bokstäver i textrutan nederst i fönstret, och när man trycker på knappen *Find Next* får man information om nästa komponent som börjar på dessa bokstäver. Sökningen sker i det fält i databasen som heter klartext och innehåller text som beskriver komponenten (se figur 7.3.).

När man har hittat någon komponent som man vill se en figur av trycker man på *OK* - knappen och programmet öppnar tillhörande komponentfil. Samtidigt stängs *Sök i sökfil* - fönstret. Det är också möjligt att stänga fönstret utan att välja någon komponent genom att trycka på *Cancel* - knappen. *Cancel* - knappen har alltid funktionen att stänga ett aktivt fönster utan att göra något annat.

8.1.2. Sök efter filnamn:

När man väljer *Sök efter filnamn* i *Import Nick* menyn aktiveras ett vanligt "File open" fönster som man känner igen ifrån alla Windows program (se figur 8.3.).



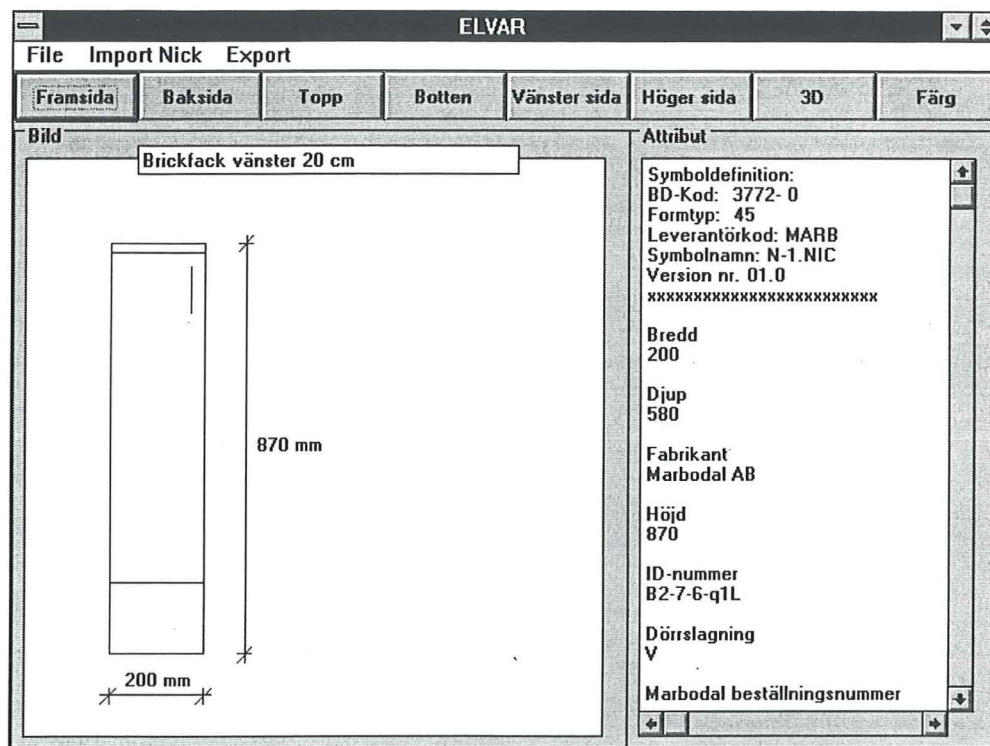
Figur 8.3. Sök efter filnamn fönster.

Där kan man öppna filer var som helst på hårddisken eller en diskett. När fönstret aktiveras är standard suffix på filnamnen .nic, men man kan ändra det genom att ändra texten i textrutan *Filnamn* till något annat av typen *.xxx eller gå till listan i *Filtyper* och där välja emellan *.* och *.nic. När man trycker på *OK*-knappen och texten i textrutan *Filnamn* börjar på * ändrast fillistan, men fönstret stängs inte förrän man har markerat någon fil i fillistan och tryckt på *OK*-knappen.

Det går inte att öppna filer av någon annan typ än filer på Nick-format.

8.2. Attribut och figurer

När man har valt en komponent läser programmet in hela filen i minnet och när detta är färdigt skrivs symboldefinitionsinformation och attribut i en textruta på skärmen och text som tillhör komponenten i en annan textruta (se figur 8.4.).



Figur 8.4. Huvudfönster i programmet Elvar med attributlista och beskrivande text och figur av framsida.

För att rita figurer i bildrutan på skärmen trycker man på någon av knapparna nedanför menyraden, *Framsida*, *Baksida*, *Topp*, *Botten*, *Vänster sida* eller *Höger sida*. Om information om den vy som man vill rita inte finns i Cad-komponenten får man 'message box' på skärmen som talar om detta, (se figur 8.5.).

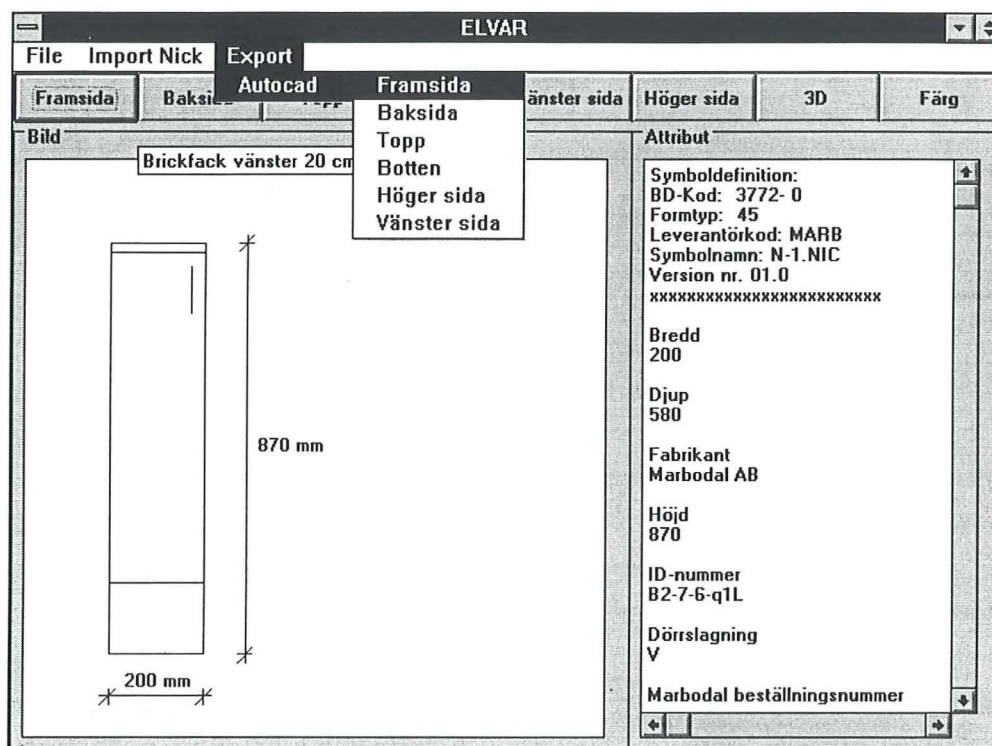


Figur 8.5. Message box.

8.3. Export

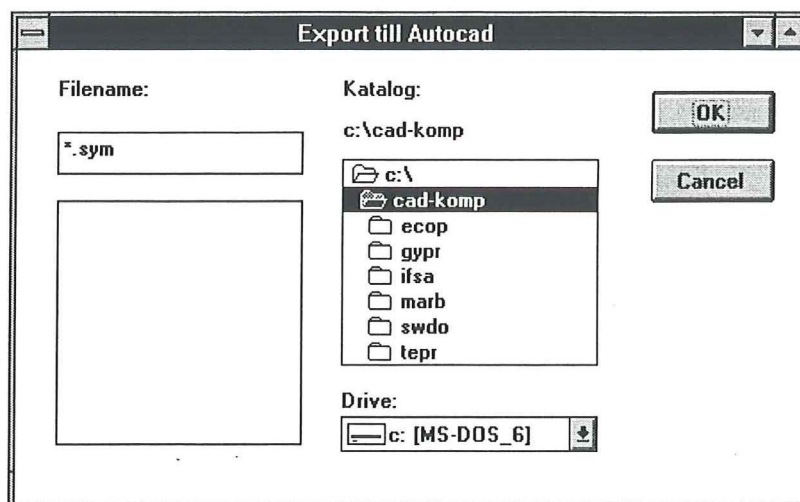
I Windows-programmet Elvar har man möjlighet att skriva en fil som innehåller grafik för en eller flera Cad-komponenter på hårddisken. Filen innehåller komponentens attribut och grafik i en vald vy. Denna fil kan man sedan läsa in i Autocad med Autolisp-programmet Elvar.

För att skapa filen väljer man *Export* i menyraden i Elvars huvudfönster. Då får man en undermeny där man väljer vilken vy man vill exportera (se figur 8.6.).



Figur 8.6. Export meny

När man har valt vy får man ett vanligt "File open" eller "File save" fönster på skärmen (se figur 8.7.). Där kan man skriva in ett nytt filnamn eller använda ett gammalt. Om man väljer ett filnamn som redan finns skriver programmet efter sista raden i filen, men skriver inte över information som redan finns i filen. På detta sättet kan man exportera fler än en symbol i en fil. Filerna har som standard suffixet .sym.



Figur 8.7. Export fönstret.

För att läsa in symbolfilen i Autocad måste man ladda in Autolisp-programmet Elvar till Autocad. Det kan man göra med kommandot (*load "elvar"*) varje gång man startar Autocad eller skriva detta kommando i filen Acad.dwg, då laddas Elvar.lsp automatiskt när Autocad startar.

När man sen vill läsa in en symbol i Autocad skriver man *Elvar* i kommandoraden och lisp-programmet startar. Först frågar programmet efter namn på symbolfil, och där måste man skriva namn, efternamn och katalog om symbolfilen inte ligger på samma katalog som Autocad arbetar. Därefter ritlar programmet symbolen och gör ett 'block' av den. Symbolnamnet i Cad-komponentfilen används i första hand för namn på blocken, men om det redan finns en symbol med samma namn får man chansen att ge symbolen ett annat namn eller avsluta programmet. Till sist frågar programmet efter insättningspunkt och rotation för symbolen. Där kan man använda Autocads *objekt snap* för att välja insättningspunkten.

När symbolen är på plats kan man med kommandot *ddatte* editera attributen, och i Autocad version 12 får man ett fönster på skärmen som innehåller en lista över alla attribut som tillhör blocken.

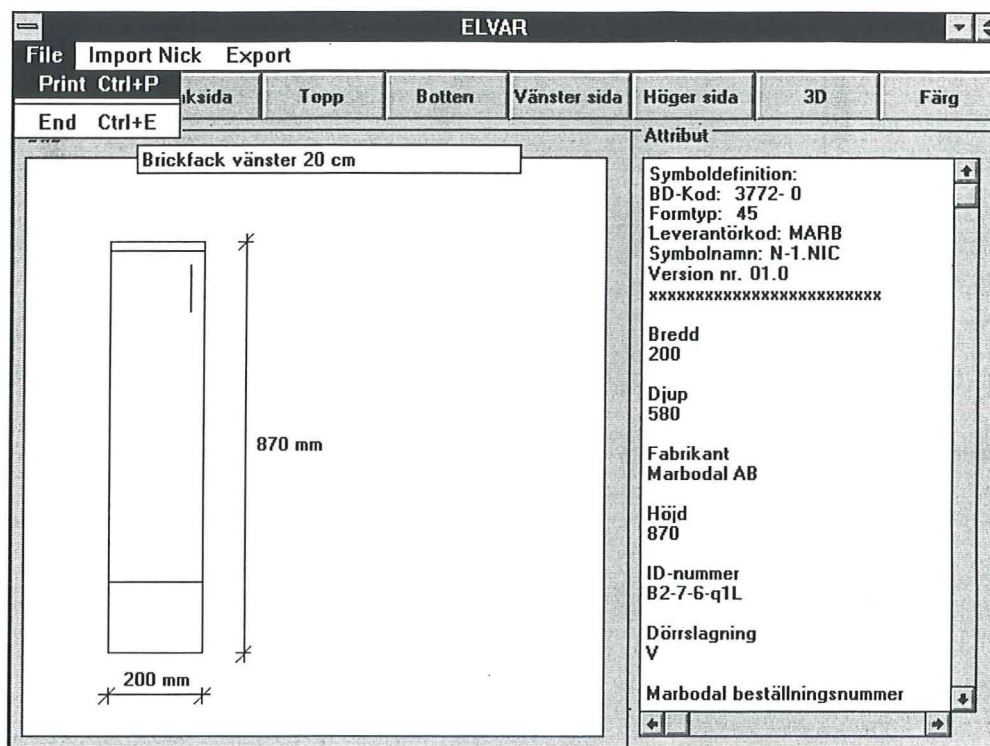
8.4. File menyn.

Under *File* menyn i huvudfönstret finns det två kommandon, *Print* och *End* (se figur 8.8.).

Med *Print* kommandot kan man på skrivare skriva ut figuren som visas på skärmen, d.v.s. den vy som bildrutan visar. Då används den skrivare som är

installerad i Windows. I stället för att använda menyn kan man använda förkortningen *Ctrl-P*.

End kommandot avslutar programmet. I stället för att använda menyn kan man använda förkortningen *Ctrl-E*.



Figur 8.8. File menyn

9. Slutsatser

Nu när nästan alla ritningar görs med hjälp av Cad-system kan man fråga sig om vi är på rätt väg. Är det bättre på något sätt att använda Cad än att rita för hand?

Bland fördelarna med att använda Cad är när man kan använda sig av ett symbolbibliotek för att lägga in objekt på ritningen och möjligheten att lägga produktinformation på ritningen i form av text som inte visas på ritningen men är tillgänglig i ritningsprogrammet. Denna information kan användas på många sätt, till exempel följande.

- Som hjälpmedel vid beställning. Det blir enkelt att ta fram de egenskaper som arkitekten tänkte att produkten skulle ha och skicka till leverantören.
- Produktinformation kan användas vid automatisk mängdavgivning i ritningsprogrammet. Vid mängdavgivning med hjälp av ritningar ritade för hand är risken stor för fel, men den blir mindre om mängdavgivningen kan göras med hjälp av datorprogram.
- Det kan underlätta, kontroll under byggtiden, drift och förvaltning av byggnaden, att ha tillgång till produktinformation på ritningen.
- Det kan också underlätta vid försäljning av fastigheter om säljaren har något mer att visa än bara ritningar. Det är då fördel att ha en beskrivning av produkterna som skall installeras i byggnaden.

Genom att ha tillgång till ett Cad-komponentbibliotek får man två fördelar samtidigt, man har tillgång till ett symbolbibliotek, och man får med symbolen i form av attribut, produktinformation för respektive symbol. Av det kan man dra slutsatsen att behovet finns för Cad-komponenter. Det finns däremot vissa förutsättningar för att man skall kunna använda Cad-komponenter.

- Någon måste ha intresse av att göra Cad-komponenter. Det blir i första hand leverantören eller tillverkaren som gör Cad-komponenter, och det gör han om hans varor säljs mer på grund av detta. Det gör det också lättare för honom att göra anbud om informationen från Cad-komponenten används i en förfrågan.

- Det behövs databaser som projektörer har tillgång till på något sätt, t. ex. på diskett, eller genom telenät.
- För att kunna använda Cad-komponenter behöver projektören datorprogram för sökning i databasen, och överföring av Cad-komponenten till sitt Cad-system.

Om dessa förutsättningar finns tror jag att Cad-komponenten kan leva vidare. Samtidigt måste en anpassning göras till de generella produktmodeller i 3D som hålles på att formuleras internationellt

En del av detta examensarbete har varit att göra ett datorprogram som klarar av sökning i databas och överföring till ett Cad-system (Autocad). Programmet som utvecklades har jag kallat Elvar. Det är kanske inte tänkt som någon färdig programpaket, men det visar att det inte behöver vara så komplicerat att använda Cad-komponenter. I framtiden kan man tänka sig att programmet kan kopplas till Svensk Byggtjänst Byggvaruregister, som beskrivs i kapitel 5.2.3.

10. Ordlista.

AMVI	Advanced Material and Vendor Information system
API	Application Programming Interface. Bibliotek som levereras med Microsoft Windows och innehåller programfunktioner.
Attribut	Information tillhörande en symbol eller en komponent i form av text som beskriver egenskaper eller funktion.
BSAB	Gemensamt klassificeringssystem för svensk byggsektor.
Cad-Komponent	Cad-komponenten är en textfil på NICK-format, som beskriver grafiken för en vara och också innehåller produktinformation.
DDE	Dynamic Data Exchange. Metod som Microsoft Windows använder för att kommunicera mellan två applikationer.
Elektronisk varuinformationssystem	System som presenterar produktinformation för varor på en datoriserad form.
NICK	Neutral Intelligent Cad-Kommunikation.
Polygon	Linje sammansatt av många delar. En polygon kan vara öppen eller sluten.
STEP	Standard for the Exchange of Product model data.

Struktur	I programmering, en post som består av element av olika typ
Sökfil	Fil som innehåller information som gör det lättare att söka i ett bibliotek.
Teckentabell	Tabell där tecken och symbol har sin plats och representeras med ett nummer. Ex. ASCII.
Windows for Windows	Elektronisk katalog för fönster för att köra i Microsoft Windows.

11. Referenser

- [1] Kristiansen, Sven, 1990, Neutral byggproduktmodell för integrerad projektering, produktion och förvaltning - systembeskrivning, Byggnadssyrelsens informationer T:123, Stockholm.
- [2] Tarandi, Väino, 1991, NICK - Neutralt format för intelligent CAD-kommunikation (Statens råd för byggforskning) Rapport R70:1991, Stockholm.
- [3] SB-Recommendationer 7, Cad-komponenter, 1994, Svensk byggtjänst
- [4] Hansson, Pål, Karsson, Göran, Pärletun, Lars Göran, 1991, Kom i gång med AutoCad, Studentlitteratur, Lund.
- [5] Hansson, Pål, Karsson, Göran, Pärletun, Lars Göran, 1990, Från CAD till CIM, Studentlitteratur, Lund.
- [6] Cornell, Gary, 1993, The Visual Basic 3 for Windows Handbook, Osborne McGraw-Hill, Berkeley, California, U.S.A.
- [7] Microsoft Corporation, 1993, Användarhandbok Microsoft MS-DOS vesion 6.2.
- [8] Autolisp Realease 11, Programmer's Reference.
- [9] Björk, B C, 1990, STEP internationell standard för digital överföring av produktinformation, Statens råd för byggforskning, G6:1990, Stockholm.
- [10] Christiansson, Per, 1990/1991, Advanced Material and Vendor Information System - AMVI, Avdelningen för Bärande Konstruktioner, Lund. Automation in Construction 2 (1993), (pp 109-121)
- [11] Follin, Anders, 1994, Digital Product Catalouges, An evaluation of presentation and visualization techniques, Avdelningen för Bärande Konstruktioner, Lund.
- [12] SB-Recommendationer 6, BSAB-systemet, Tabeller och tillämpningar, 1987, Stockholm.