

Sammanfattning

VRML är en högaktuell teknik för att realisera Virtual Reality omgivningar över Internet. Forskning och utveckling inom VR har pågått ett antal år, men hittills har de befintliga VR-systemen inte gått att använda för den vanlige Internetanvändaren. VRML är ett språk som funnits sedan -94 och som nu under -96 kommer att få ett ordentligt lyft. Tidigare har VRML varit alldeles för begränsat för att nå framgång, men nu i augusti -96 blir den nya förbättrade VRML 2.0 godkänd av ISO som officiell standard. Med VRML 2.0 är det möjligt att beskriva omfattande tredimensionella världar som kan användas som komplement eller alternativ till vanliga hemsidor. Användaren av en tredimensionell VRML 2.0 värld kan förutom att röra sig i världen även hantera den och dess objekt interaktivt samt aktivera länkar till andra världar och dokument. Objekten i världen kan vara animerade, röra sig, ha viss intelligens, besitta egenskaper som 3D-ljud, m m.

Det enda som behövs för att använda och betrakta befintliga VRML-världar är en VRML-läsare. Denna läsare kopplas till WWW-bläddraren varpå en tredimensionell VRML-värld uppenbarar sig då en VRML-fil påträffas. Det finns en stor mängd VRML-läsare och arbetet med att bygga ut dessa med stöd för VRML 2.0 pågår nu för fullt. Navigering i VRML-världar sker med musen och tangentbordet. I de senaste versionerna av Netscape Navigator, och så småningom även Internet Explorer, är en VRML-läsare redan inkluderad. I takt med att Internetanvändarna uppgraderar sina WWW-bläddrare kommer möjligheterna att hantera VRML-världar att växa.

Vill man skapa sin egen VRML-värld behövs någon utvecklingsmiljö för VRML. Tillvägagångssättet för att skapa VRML världar i de olika utvecklingsmiljöerna skiljer sig mycket åt. Både grafiska och textbaserade utvecklingsverktyg finns och valet beror till stor del på kunskapsnivå och vilket tillvägagångssätt man föredrar.

Användningsområdena för VRML är många. Med VRML är det möjligt att realisera i princip alla applikationer man idag använder traditionell WWW-teknik till, men med bättre och mer lättanvända användargränssnitt. Dessutom möjliggör VRML nya typer av WWW-applikationer. I många tillämpningar räcker inte två dimensioner till, t ex virtuella konferenser där deltagarna möts och har en konferens i ett virtuellt konferensrum trots att de kan befinna sig på geografiskt skilda platser. Idag har Java en stor spridning som ett medel att skapa effektfulla hemsidor i syfte att locka fler besökare. I framtiden kan man räkna med att även VRML används flitigt för detta. Med hjälp av ett externt API som kommer under hösten, VRMLs Javakoppling, CGIer och VRMLs goda möjligheter att återanvända kod är det mesta realiserbart.

Läsanvisning

- **Kap. 1 Introduktion,**
ger bakgrundsinformation och introducerar VRML. Relationen mellan VRML, traditionell VR-teknik och Internet förklaras kort.
- **Kap. 2 VRML – Språk och funktioner,**
förklarar mer ingående hur VRML fungerar med bl a demonstrationsexempel för både VRML 1.0 och 2.0. Funktionalitet som är aktuell för inkludering i framtida VRML-versioner tas upp.
- **Kap. 3 Verktyg,**
granskar de verktyg som finns för att skapa och läsa VRML-världar. Andra intressanta VR-system för Internet än VRML berörs också kortfattat.
- **Kap. 4 Applikationsområden,**
anger ett antal exempel på applikationer som med fördel skulle kunna realiseras med VRML samt utvärderar kort nyttan med VRML.

Innehåll

1. INTRODUKTION.....	5
1.1 VIRTUAL REALITY OCH VISIONEN OM CYBERSPACE	5
1.2 VIRTUAL REALITY MODELING LANGUAGE (VRML) – INTRODUKTION	7
1.3 VRML HISTORIK	8
1.4 VAD ÄR MÖJLIGT IDAG?.....	9
2. VRML – SPRÅK OCH FUNKTIONER.....	11
2.1 VRML 1.0.....	11
2.2 VRML 2.0.....	14
2.3 PÅ GÅNG INOM VRML.....	17
2.4 VIDARE LÄSNING	19
3. VERKTYG.....	20
3.1 VRML-LÄSARE.....	20
3.2 UTVECKLINGSMILJÖER FÖR VRML.....	26
3.3 ALTERNATIV TILL VRML.....	32
4. APPLIKATIONSOMRÅDEN.....	34
4.1 GRUPPSAMARBETE.....	34
4.2 VISUALISERING.....	35
4.3 FÖRSÄLJNING	38
4.4 UNDERHÅLLNING.....	39
4.5 INFORMATIONSTJÄNSTER	41
4.6 VANLIGA SIDOR.....	43
4.7 KOMMER VRML ATT ANVÄNDAS?.....	43

1. Introduktion

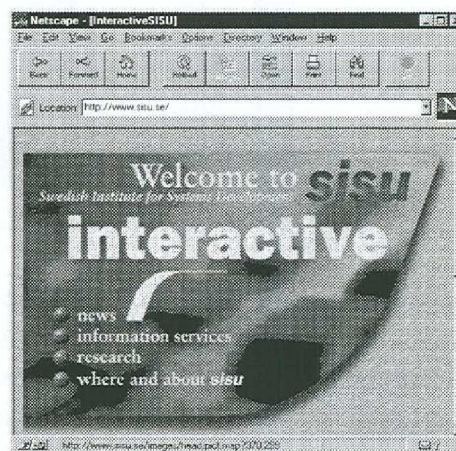
Internet är ett område där utvecklingen sker i rasande fart. När Internet började bli känt för allmänheten i början av 90-talet var det e-post som de flesta tänkte på när man pratade om Internet. Ville man överhuvudtaget navigera sig fram på Internet användes tangentbordet i en ganska svårtillgänglig endimensionell värld av FTP, Telnet etc. Huvuddelen av all överföring skedde via FTP-protokollet. Idag tänker allmänheten främst på att "surfa" när man säger Internet. D v s att använda en WWW-bläddrare för att med musens hjälp navigera sig fram i en tvådimensionell World Wide Web (WWW) värld.

```
ftp ftp.cdrom.com
Connected to wv.archive.cdrom.com.
220 wv.archive.cdrom.com FTP server (Version wu-2.4-dg(29))
User (wv.archive.cdrom.com:(none)): anonymous
331 Guest login ok, send your complete e-mail address as password.
Password:
230-Welcome to wv.archive - home ftp site for Walnut Creek CDROM.
230-There are currently 625 users out of 1200 possible.
230-
230- Guest login ok, access restrictions apply.
ftp> cd pub/misc
250 CWD command successful.
ftp> dir
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
total 6
-rw-r--r-- 1 201 wheel 210 Apr 19 1995 .cache
-rw-r--r-- 1 201 wheel 1012 Apr 18 1995 .cachet
----- 1 root wheel 0 Feb 10 11:30 .netar
lrwxrwxr-x 1 root wheel 23 May 7 09:11 MacSciTech
drwxrwxr-x 13 root wheel 512 Jul 29 13:07 vmach
226 Transfer complete
310 bytes received in 0.05 seconds (6.20 Kbytes/sec)
ftp>
```

```
telnet minsk.doc.su.se
Trying...
Connected to minsk.doc.su.se.
Escape character is '^]'.
Local flow control off
SumOS UNIX (Minsk)

login: tomyri
Password:
Last login: Tue Jul 30 08:46:29 from dhcp6.sisu.se
SumOS Release 4.1.3 (MINSK) #5: Mon Apr 24 14:21:49
terminal type vt100
~/: bashrc on Minsk (sparc)
BASH version 1.14.4(1)

[Minsk] tomyri > ls
#_X11Startup#  RMMAIL-   kurser      phinger.irc
ctx            Calendar  bin         mail
private       xjob     RMMAIL     emacs
manual.ps     proj     RMMAIL.old  fizzy.irc
public.html   RMMAIL.old-  irc.help    outgoing
tmp
[Minsk] tomyri > emacs
```



Figur 1 – Internetnavigering med FTP och Telnet till vänster, med Netscapes WWW-bläddrare till höger.

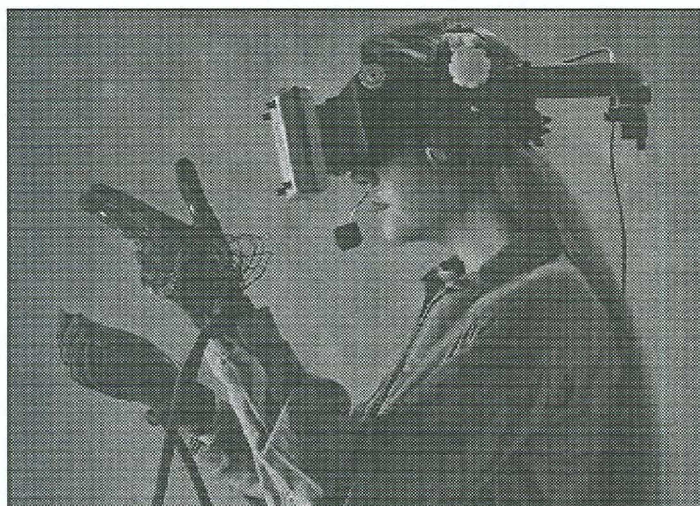
Om ett par år har kanske uppfattningen om vad Internet är ändrats åter igen. Det är numera möjligt att via Virtual Reality Modeling Language (VRML) navigera i en tredimensionell värld. Enligt VRML-entusiasterna kommer detta att ge en ny Internet-boom. Fördelarna med att ha ett tredimensionellt gränssnitt är framförallt två. För det första kan gränssnitt göras mer intuitiva eftersom de kan göras mer lika verkligheten än tvådimensionella WWW-sidor. För det andra kan man möjliggöra saker som tidigare inte gick att realisera med WWW.

1.1 Virtual Reality och visionen om Cyberspace

Cyberspace är ett uttryck myntat av science fiction författaren William Gibson för att beskriva hans vision av ett världsomspännande globalt nätverk där alla personer, datorer och informationskällor är sammankopplade. I Gibsons cyberspace kan man röra sig i denna virtuella rymd och hantera de olika objekten i rymden. Cyberspace är inte bara en "passiv" rymd, utan verkligheten kan styras från cyberspace eftersom de är hopkopplade. Att röra sig i cyberspace är en upplevelse för hela kroppen och alla sinnen, det känns och är som att röra sig

i verkligheten. Om man t ex använder ett föremål i cyberspace känns det som om man verkligen använder *föremålet*, inte en bild eller representation av föremålet.

Virtual Reality (VR) är en tredimensionell rymd genererad och hanterad av datorer. I denna virtuella verklighet ska man kunna röra sig och kommunicera med andra personer som befinner sig i rymden. Allt som görs i virtuella verkligheten ska fungera på samma sätt som det gör i verkliga livet. Det Virtual Reality eftersträvar är att realisera Cyberspace så långt som det är möjligt.



Figur 2 – Navigering i VR-värld med datahandskar och headmounted display.

Virtual Reality är inget nytt begrepp. Forskning och utveckling inom VR har pågått under många år. De VR-system som har funnits hittills har generellt sett krävt kraftfulla datorer och snabba kommunikationskanaler. De huvudsakliga användningsområdena har varit avancerade simuleringar och efterlikningar av verkligheten. Flygpiloter genomgår idag en stor del av sin flygträning på marken i simulatorer där verkligheten har försökt efterliknas så långt som möjligt. Öresundsbron har modellerats med VR-tekniker för att göra det möjligt att med egna ögon nästan exakt se hur den kommer att påverka landskapet innan den ens har börjat byggas. Kockums har tagit hjälp av VR-teknik för att sälja U-båtar som ej var färdigbyggda till Australien. U-båtarna avbildades helt enkelt i VR så att de australiensiska spekulanterna fick en mycket bra uppfattning om hur de skulle se ut och fungera. I USA finns det exempel på planerade köpcentrum som till en början mött på motstånd hos allmänheten, men efter de sett hur de verkligen kommer att se ut och påverka omgivningen (genom att modellera byggnaderna med VR-tekniker) har uppfattningen ändrats. Dessutom har man kunnat testa ett antal olika designalternativ och upptäcka brister med projekten som man förut inte såg förrän det var för sent att korrigera. På senare tid har VR även hittat lite mera vardagliga tillämpningsområden som troligtvis ligger närmare vad VRML m fl siktar på. Vissa svenska frisörer har VR-system för att låta kunden få sitt ansikte avbildat på en datoriserad provdocka och i förväg få en uppfattning om hur olika frisyrier kommer att passa. IKEA erbjuder sina kunder att se hur deras soffa kommer att se ut med olika sofftyger genom att lägga det valda mönstret på en datoriserad soffa inuti datorn och låta kunden se hur det ser ut innan han bestämmer sig.



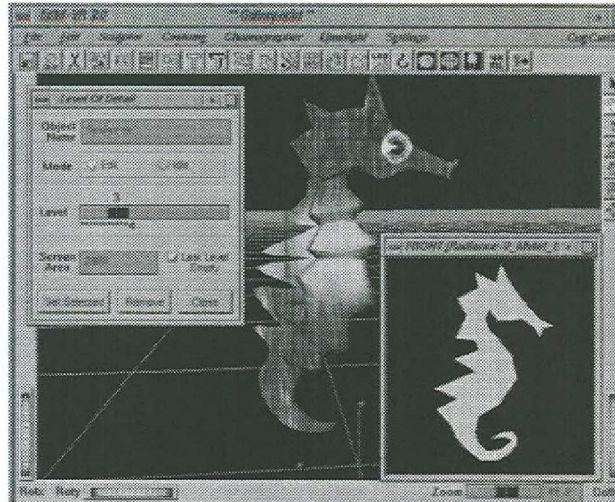
Figur 3 – En avancerad VR-flygsimulator. Till vänster cockpit och till höger en vy utifrån.

Den explosionsartade Internetutvecklingen har naturligtvis skapat ett intresse för att använda VR-systemen i denna miljö. Men då förutsättningarna på Internet vad gäller nätkapacitet och datorer oftast är långt ifrån vad systemen är byggda för är det svårt att direkt använda befintliga VR-system. Istället har intresset för VR-system som är gjorda för Internet och WWW börjat växa. I dessa försöker man minimera både kommunikationen och beräkningsmängden, något som är en svår balansgång eftersom den ena tenderar att öka när den andra minskar. Bland de Internetspecifika VR-systemen är det framförallt VRML och dess varianter som är mest aktuella och intressanta i dagsläget.

1.2 Virtual Reality Modeling Language (VRML) – Introduktion

Virtual Reality Modeling Language är ett språk för att i textform beskriva en enkel tredimensionell värld. Principen i VRML är liksom i HTML att en textfil som beskriver världarna/dokumenterna överförs från servern till klienten. När klientens WWW-bläddrare får ett dokument som den känner igen som en VRML-värld vidarebefordras den till den VRML-läsare som är kopplad till WWW-bläddraren. VRML-läsaren tolkar sedan textfilen och genererar sedan själv världen utifrån beskrivningen. Detta innebär att det inte blir några större bandbreddsproblem, en stor del av flaskhalsen flyttas över till klient-datorernas VRML-läsare.

Med VRML är det möjligt att förflytta sig i den tredimensionella världen och, i viss mån, interagera med föremålen i världen. Fördelen med VRML jämfört med många andra sätt att försöka implementera VR är framförallt att den är konstruerad för att fungera ihop med WWW-teknik. Alla som är uppkopplade mot Internet kan med sin WWW-bläddrare navigera även i VRML världar. Vilket operativsystem och vilken dator användaren har är av mindre betydelse. VRML är direkt anpassat och optimerat för just de förutsättningar som råder på Internet. Såväl de som sitter uppkopplade med en vanlig PC via ett modem som de som sitter på en kraftfull arbetsstation via höghastighetslänkar ska kunna delta. Effektiva grafikhanteringsalgoritmer och diverse trick som t ex att kunna komprimera VRML-filer och möjlighet att ställa in detaljnivå på avlägsna föremål gör att de flesta världar ska kunna hanteras på en vanlig PC med Internetanslutning via modem.



Figur 4 – Ez3d utvecklingsmiljö för VRML.

För att skapa sin egen VRML-tillämpning behövs troligen något slags hjälpmedel då det är mycket tidsödande att skapa icke-triviala VRML-världar genom att skriva in textbeskrivningen av VRML-världen för hand i en ordbehandlare. Det finns både utvecklingsmiljöer där man "ritar" sin värld i tre dimensioner och miljöer där man genom att ange VRML-kommandon och parametrar skapar sin värld på ett programmeringsliknande sätt. För att sedan använda eller betrakta skapade världar används en VRML-läsare. Denna läsare sammankopplas med WWW-bläddraren och startas automatiskt av WWW-bläddraren när en VRML-värld påträffas.

1.3 VRML-historik

1994 kom den första versionen av VRML. Det var Mark Pesce, Gavin Bell och Anthony Parisi som tog fram den första prototypen till ett språk för att beskriva 3D-rymd ihop med WWW-länkar som fungerade med det befintliga HTTP-protokollet. Denna prototyp fick då namnet Virtual Reality Markup Language men strax därefter byttes ordet Markup ut mot Modeling då detta bättre beskriver vad VRML är. Silicon Graphics erbjöd VRML-skaparna att använda deras Open Inventor ASCII File Format för att realisera VRML så snabbt som möjligt (Open Inventor är ett filformat som beskriver 3D-scener med vanlig text). Då man ville ha VRML-standarden klar snabbt bestämde man sig för att endast stödja en delmängd av den 3D-funktionalitet som finns i Open Inventor-filformatet samt att lägga till funktionalitet för WWW-länkar samt inlines (möjlighet att inkludera en VRML-värld i en annan, jämför med HTMLs IMG SRC).

Tanken med VRML 1.0 var alltså inte att göra en fulländad implementation av VR för WWW, utan främst att få en grundläggande och fungerande VR-realiseringsmiljö som det gick att bygga vidare på. Det var mycket som saknades, framförallt möjligheten att interagera med objekten i VRML-världarna på annat sätt än att förflytta sig och klicka på länkar. Bättre realism kräver exempelvis funktionalitet för att öppna en dörr och kliva in i nästa rum, istället för att klicka på dörren och bli "teleporterad" till nästa rum.

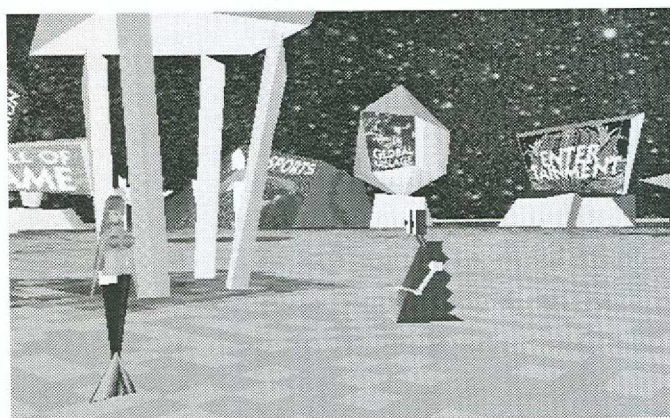
Version 1.1 av VRML presenterades i slutet av 1995 och var tänkt att utvidga 1.0 med ett par väsentliga men ganska enkla egenskaper, bl a ljud, flerspråkig text och förbättrad geometri, men

man bestämde sig för att inte satsa på 1.1. Man väntade hellre ett tag till på den mera komplett utbyggda standarden 2.0 som redan hade påbörjats sommaren 1995. Men flera Internetaktörer ville inte vänta utan gjorde egna tillägg till VRML 1.0 för att passa deras behov. Tilläggen som gjordes var inte koordinerade och resultatet blev att en mängd olika varianter av VRML uppstod med kompatibilitetsproblem sinsemellan.

Inför version 2.0 upprättade VRML Architecture Group (VAG)¹ en mycket kort kravspecifikation på vad de ville ha i den nya VRML standarden. I stora drag kan man säga att VRML 2.0 handlar om interaktion. De huvudsakliga punkterna i kravspecifikationen är att VRML 2.0 måste vara snabb, skalbar, bakåtkompatibel, det måste vara möjligt att skapa avancerade utvecklingsverktyg till 2.0, möjligheter att inkludera en värld i en annan, det måste gå att bygga ut VRML med tillägg som kan återanvändas av andra, kunna samarbeta med andra filtyper/språk som t ex Java samt vara förberett för fleranvändarstöd. Sedan gick VAG ut med denna kravspecifikation och begärde in förslag från olika Internetaktörer. Sammanlagt lämnades det in sex förslag på ny standard. Alla dessa förslag skiljer sig åt i realiserandet och efter en viss tids debatt och ett öppet val blev Moving Worlds från Silicon Graphics utsett till det förslag som skulle tas upp till behandling för standardisering av ISO.

1.4 Vad är möjligt idag?

De tillämpningar på Internet där VRML används idag är oftast inte så häpnadsväckande. Eftersom VRML 2.0 kommer att bli slutgiltigt klart först i augusti '96 baseras dagens tillämpningar på VRML 1.0 med tillägg. Verktøygen för att läsa VRML-världar hänger med bra i utvecklingen av VRML, nya versioner med ny funktionalitet släpps kontinuerligt. De verktyg som finns för att skapa VRML-världar är dock inte lika långt gångna i stöd för ny funktionalitet. Att skapa VRML 1.0 världar med olika tillägg för fleranvändarstöd och beteenden är möjligt med hjälp av verktyg idag. Stöd för VRML 2.0 är utlovat av ett antal utvecklingsmiljöer.

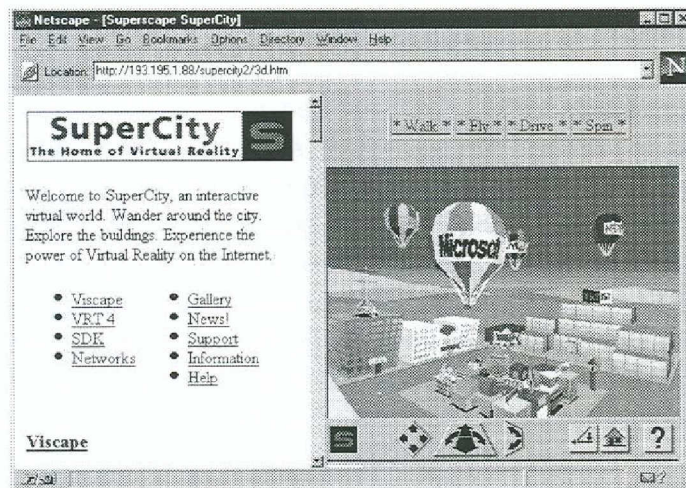


Figur 5 – Chatvärldar, VRMLs motsvarighet till heta linjen.

Figuren ovan visar en bild från PointWorld som är en sk chatvärld där flera deltagare möts och pratar med varandra. Idag sker kommunikationen via text, men stöd för ljudkommunikation är

¹ VRML Architecture Group och dess arbete med att försöka nå enighet i en gemensam VRML 2.0 standard m.h.a. ett öppet val m m beskrivs på <http://kether.vrml.org>.

på gång. I PointWorld kan man diskutera privat en och en, i grupper, eller med alla närvarande. Man kan utbyta ett slags visitkort och man kan klicka på någon av de länkar som syns i form av skyltar i bakgrunden.



Figur 6 – SuperCity.

Figuren på SuperCity av SuperScape ovan visar hur man kan använda sk frames för att dela upp fönstret i olika delar som kan samarbeta. I VRML-världen svävar animerade ballonger som man kan klicka på för att komma till SuperCitys sponsorer och i byggnaderna finns SuperScapes olika avdelningar m m. Man kan även välja att få guidade turer genom vissa delar av världen.

Allt detta är realiserat idag i VRML 1.0 med tillägg. VRML 2.0 standardiserar det som det förut funnits mängder av olika tillägg för och möjliggör en hög grad av interaktivitet. Med fleranvändartillägg till VRML 2.0 kan användaren röra sig fritt i världen och kommunicera och interagera både med objekten och andra användare i världen. Vad man kan åstadkomma med detta, så fort rätt utvecklingsverktyg finns tillgängliga, sätter bara fantasin gräns för.

2. VRML – Språk och funktioner

VRML är som tidigare nämnts ett språk för att beskriva VR-världar som är skapat och optimerat för Internet. Det finns ett flertal olika VRML baserade språk. VRML 1.0 och 2.0 är de rena VRML-versionerna som är standardiserade. Utöver dessa finns ett antal utbyggnader med tillägg för ny funktionalitet.

2.1 VRML 1.0

VRML 1.0 är ett mycket enkelt och minimalt språk för att beskriva en VR-värld. Tanken med VRML 1.0 var att ha en grund att basera kommande språk för VR över Internet/WWW på. Filformatet för att beskriva VRML världar är en vanlig textfil, och eftersom VRML 1.0 är så pass begränsat är det fullt möjligt att skapa de flesta enkla världar "för hand" utan några hjälpverktyg. Man anger rotationer och förflyttningar samt utplacering av objekt i ett 3D-koordinatsystem. Objekten kan vara t ex texter eller geometriska objekt som kon eller kub. Objekten kan ha egenskaper som t ex material, textur eller detaljnivå. Fasta ljuskällor av olika typer kan placeras ut och flera objekt kan grupperas ihop till ett. De enda möjligheterna till interaktion med världen som erbjuds är att gå runt och betrakta världen från olika synvinklar samt klicka på WWW-länkar till andra världar/dokument. Förutom detta är världen helt statisk och begränsningarna ganska stora. Man kan inte interagera med andra användare och världen är helt orörlig och tyst.

Då VRML 2.0 bygger på VRML 1.0 är det ganska viktigt att få en förståelse för hur VRML 1.0 fungerar. För att ge en inblick i VRML 1.0 skapar vi en enkel demonstrationsvärld med ett litet hus och ett par träd.

VRML-koden är hierarkiskt arrangerad och grupperad i en struktur som kallas för scen-
grafen. Alla "kommandon" i en VRML-fil kallas noder. Kommentarer anges med #.
Samtliga noder har standardvärden som används om inget annat anges. Det koordinatsystem
som används kallas för högerhandskoordinatsystemet och innebär att X- och Y-axlarna pekar
som vanligt och att Z-axeln pekar ut genom skärmen. När geometriska objekt placeras ut anges
alltid koordinaterna för objektets centrum. Denna demonstrationsvärld finns även på
<http://www.sisu.se/projects/vrml/exemp11.wrl>.

```
#VRML V1.0 ascii

Separator {

  DirectionalLight { direction 0 0 -1 } # Riktat ljus längs Z-axeln

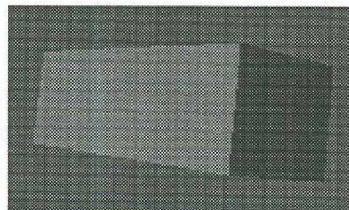
  WWWAnchor { # WWW-länk
    name "http://www.sisu.se"
    description "SISUs hemsida"
```

Varje VRML 1.0 fil måste börja med texten #VRML V1.0 ascii så att WWW-bläddraren och VRML-läsaren sedan känner igen textfilen som en VRML 1.0 fil och visualiserar världen därefter. Separator är en nod som delar upp scen-
grafen i en hierarkisk struktur. En

VRML 1.0 fil får endast innehålla *en* rot-nod så vi måste ange Separator runt hela scenrafen. Det finns olika ljusstyper i VRML, den vi använder och anger i första noden är riktat ljus och riktningen är inåt i skärmen längs Z-axeln. Nästa nod är WWWAnchor som används för att bygga in WWW-länkar i världar liknande HTMLs .

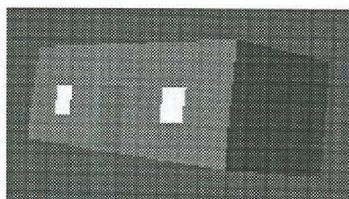
```
Separator { # Här kommer WWW-länk objektet
  Material { diffuseColor 1 0 0 } # Röd
  Cube { width 15
        height 5
        depth 6 }
```

För att gruppera flera objekt till ett används Separator för att åstadkomma rätt hierarki. Alla translationer (förflyttningar), färgval etc som görs inom Separators klamrar påverkar inte de noder som kommer efter klamrarna. Härnäst skapar vi en kub som får färgen röd.



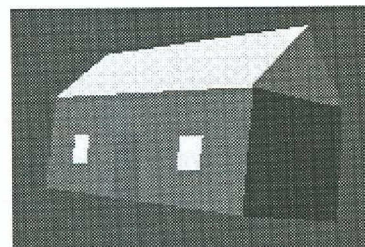
```
Coordinate3 { # Placera ut punkter
  point [ -1 -2.5 3.1, 1 -2.5 3.1, 1 .5 3.1, -1 .5 3.1,
        -3.5 -1 3.1, -5 -1 3.1, -5 .5 3.1, -3.5 .5 3.1,
        3.5 -1 3.1, 5 -1 3.1, 5 .5 3.1, 3.5 .5 3.1 ]
}
Material { ambientColor 0 0 0
          diffuseColor 0 0 0
          emissiveColor[.9 .9 .9, .3 .3 .3]} # Gråa fönster och dörrar
MaterialBinding { value PER_PART }
IndexedFaceSet { coordIndex[ 0, 1, 2, 3, -1, # Bind ihop punkterna
                          4, 5, 6, 7, -1, # till vektorplan
                          8, 9, 10, 11, -1 ]
              materialIndex [1,0,0] }
```

Vi placerar ut ett antal osynliga punkter på kuben genom att ange punkternas XYZ-koordinater och använder IndexedFaceSet för att skapa vektorplan av dessa punkter. Dessa vektorplan blir fönster och dörr på huset.



```
Transform { rotation 1 0 0 0.7854 # Translaterar och roterar i en
          translation 0 2.5 0 } # transformation
Material { diffuseColor [ 1 1 1 ] } # Ljusgrått tak
Cube { width 14.9 # Kuben som blir tak på huset
      height 4.3
      depth 4.3 }
} # Separator
} # WWWAnchor
```

För att demonstrera rotationer skapar vi en ny kub som roteras 90 grader runt X-axeln för att skapa ett tak till huset istället för att använda vektorplan till taket. Tak-kuben görs aningen kortare så att väggarna döljer den del av kuben som inte ska synas.



```

Translation { translation 11 .5 0 }
DEF tree Separator { # Definiera egna typen 'tree'
  Material { diffuseColor .1 .1 0 } # Brun
  Cylinder { height 6 } # Trädstammen
  Translation { translation 0 6 0 }
  Material { diffuseColor 0 1 0 } # Grön
  Sphere { radius 4 } # Lövverket
} # DEF

```

Sedan skapar vi en ny egendefinierad nodtyp som vi kallar "tree" med DEF. Tree består av en brun cylinder med en grön sfär. Notera att samtidigt som vi definierar denna nodtyp placeras ett objekt av nodtypen ut i världen.

```

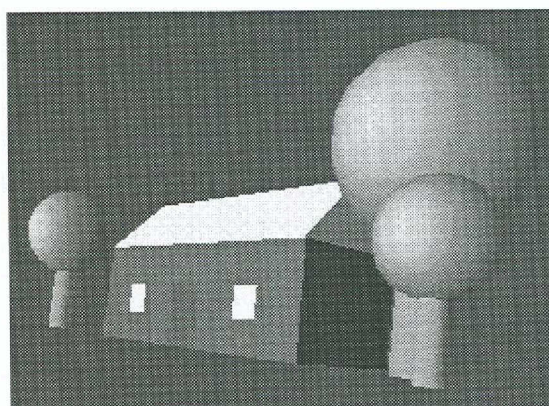
Transform { translation 3 -1.5 4 # Translatera och förminska till 50%
           scaleFactor .5 .5 .5 } # i en transformation
USE tree # Placera ut ett träd

Transform { scaleFactor1.2 1.2 1.2 # Skala upp trädet 20% jämfört med den
           translation -46 1 2 } # nedskalning på 50% vi gjorde sist
USE tree

} # Separator

```

Vi återanvänder tree med USE och skalar om trädet så vi får sammanlagt tre träd i olika storlekar runt huset. Notera att vi inte har Separator runt de enskilda träden vilket innebär att translationernas effekter kvarstår mellan de olika träden och man får translatera från den platsen där senaste trädet placerades.



Figur 7 – Slutgiltig visualisering av exempelvärlden.

2.2 VRML 2.0

Moving Worlds VRML 2.0 har utvecklats av Silicon Graphics i samarbete med Sony m fl. Syntaxen i VRML 2.0 är baserad på VRML 1.0 så övergången från 1.0 till 2.0 är till stor del att lära sig de nya nodtyperna. Det diskuterades livligt om ett helt nytt språk för animeringar etc skulle skapas, men till sist bestämdes att tills vidare använda Java och Javascript då de för det första är färdiga språk och för det andra finns det redan många Javaverktyg och böcker. Dessutom är Java väl förankrat i WWW-sammanhang. Tills vidare är Java och CGIer enda möjligheten att komma åt världen "utanför" WWW-bläddraren men ett API för kommunikation med omvärlden kommer inom kort.

Nyckelpunkter i VRML 2.0 jämfört med VRML 1.0

- Nya nodtyper för mera realistiska landskapseffekter, som t ex dimma, möjlighet att lägga till objekt som t ex berg och moln i horisonten, oregelbundna ytor som passar för t ex marken i en värld.
- Animeringar och rörelser sker via nya scriptnoder. Objekt kan även ha en viss intelligens, t ex en fågel som dyker efter maskar så fort den får syn på en.
- Det finns ett antal olika animeringsmöjligheter genom att interpolera. Man kan interpolera färgbyten på objekt, låta objekt interpoleras mjukt från en form till en annan och man kan interpolera transformationer som t ex skalningar och translationer. Dessa interpoleringsfunktioner erbjuder mycket bra möjligheter till animeringar och rörelser i världen. Alla steg i en rörelse eller animering behöver inte anges utan det är möjligt att skapa ett antal grova steg och sedan låta objekt interpoleras mellan dessa. Förutom interpoleringsfunktionaliteten kan man även skapa guidade turer genom världen som användaren får följa.
- Ljus och ljud kan positioneras och förflyttas.
- Det finns sensornoder som aktiverar saker när man går på den eller klickar på den.
- Förutom sensornoder finns det timers som kan utlösa händelser.
- Objekt kan förflyttas genom att dra dem med musen.
- Kollisionsdetektering och möjlighet att följa terrängens kontur, t ex följa en trappa uppåt.
- Prototyper är ett sätt att skapa en slags mallar för objekt. En prototyp kan vara en grupp noder som man sedan skapar instanser av, för variation kan olika fält i prototypen varieras.
- Saknar stöd för flera samtida användare, men VRML 2.0 är förberett för det.

Hur VRML 2.0 fungerar åskådliggörs liksom för VRML 1.0 bäst med ett exempel. Vi skapar en boll som studsar på två kuber när man klickar på den (se Figur 8). Exempelvärlden finns även på <http://www.sisu.se/projects/vrml/exempel2.wrl>. Det vi tittar närmare på i exemplet är framförallt möjligheten till att få objekt med liv i, möjligheten till att skapa prototyper samt en liten demonstration av interaktion med objekt. Prototyper fungerar ungefär som DEF som demonstrerades tidigare i VRML 1.0-exemplet. Skillnaden är att en prototyp även kan innehålla parametrar.

```
#VRML V2.0 utf8
DirectionalLight { direction -0.3 -.5 -1 }
Viewpoint { position 20 -10 45 }
```

Vi börjar liksom i VRML 1.0 med att specificera vilken version av VRML som används. Därefter anger vi en riktad ljuskälla som går snett inåt i skärmen och anger varifrån användaren skall betrakta världen.

```
DEF JumpingBall Transform { # Kalla den hoppande bollen JumpingBall
  children [
    Shape { appearance Appearance { material Material { diffuseColor 0 1 1 }}
          geometry Sphere {radius 5 } } # Bollen
```

Det första vi skapar är bollen. Transform i VRML 2.0 fungerar lite olika mot Transform i 1.0. I 2.0 är Transform en grupperingsnod som grupperar ihop transformationer. Children liknar Separator i VRML 1.0 och delar upp scenografen i en hierarkisk struktur. Geometriska objekt som kub och sfär anges i 2.0 med Shape noden där även material m m anges.

```
DEF Click TouchSensor { } # Sensor som reagerar på musklick
DEF Timer TimeSensor { cycleInterval 10 } # Tidssensor
DEF jump PositionInterpolator { # Interpolerar bollen mellan positionerna
  keys [ 0, .1, .15, .2, .25, .3, .35, .4, .45, .5,
        .55, .6, .65, .7, .75, .8, .85, .9, 1]
  values [0 0 0, 0 -20 0, 4 -10 0, 8 -6 0, 12 -10 0,
         16 -24 0, 20 -14 0, 24 -10 0, 28 -14 0, 32 -28 0,
         28 -14 0, 24 -10 0, 20 -14 0, 16 -24 0, 12 -10 0,
         8 -6 0, 4 -10 0, 0 -20 0, 0 0 0]]
}
```

För att känna av när man klickar på bollen definieras en klicksensor med TouchSensor. Vi vill att bollens rörelse ska pågå i 10 sekunder så en tidsensor för detta definieras sedan. För att låta bollen studsas mjukt behövs en interpolernod. Det finns ett antal olika och den vi använder, PositionInterpolator, är för att interpolera positionen på objektet. PositionInterpolator tar som parametrar den tid som varje steg ska få ta (ex första steget är mellan 0 och 0.1 vilket innebär att första steget tar 10% av tiden som i detta fall blir en sekund) och vilka XYZ-koordinater varje steg ska ha. De XYZ-koordinater som är angivna beskriver en tänkt bana bollen skall studsas ner och sedan upp samma väg.

```
PROTO RedBox [ field SFVec3f boxSize 16 8 4 ]
{
  Transform {
    children [
      Shape { geometry Box { size IS boxSize } # Använd parametern boxSize
            appearance Appearance {
              material Material { diffuseColor 1 0 0 } }
    ]
  }
}
```

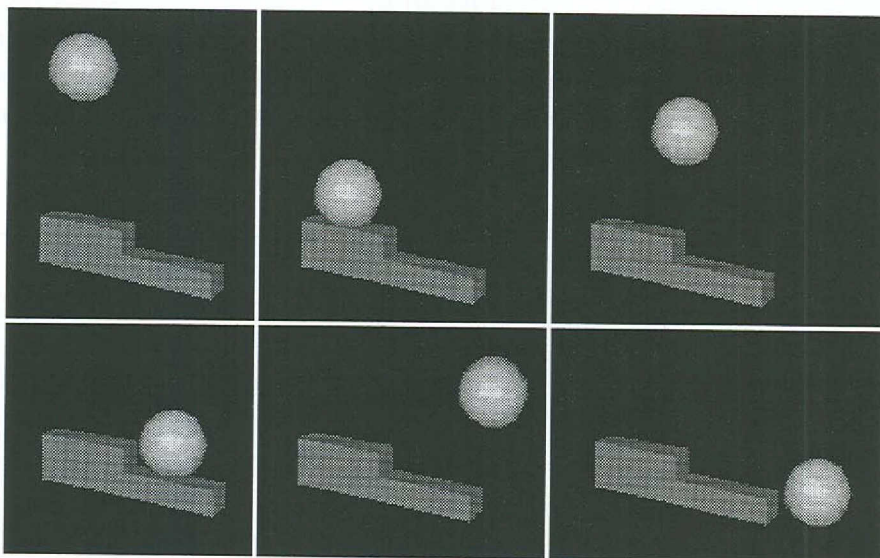
Härnäst definieras två klossar som bollen ska studsas på. För detta används prototyper. Vi definierar på nästan samma sätt som med `DEF` en kub vid namn `RedBox`. Skillnaden mot `DEF` är att vi anger inparametrar med standardvärden och använder dem inom prototypen med `IS`. `RedBox` tar som inparametrar storleken på boxen och har XYZ-värden givna ifall `RedBox` används utan parametrar.

```
Transform {
  translation 0 -28.5 0
  children [
    RedBox { } # Första boxen får standardvärdena
    Transform {
      translation 16 -2 0
      children [
        RedBox { boxSize 16 4 4 } # Andra boxens storlek anges explicit
      ]
    }
  ]
}
```

Prototypen `RedBox` används sedan för att skapa de två röda kuberna.

```
ROUTE Click.touchTime TO Timer.startTime
ROUTE Timer.fraction TO jump.set_fraction
ROUTE jump.value_changed TO JumpingBall.translation
```

Det enda som återstår nu är att få liv i bollen. Genom att vidarebefordra de signaler som kommer från de olika sensorerna är detta mycket enkelt. Klicksensorn `Click` aktiveras när man klickar på den och då vidarebefordras tiden man klickade på den med `ROUTE` till tid-sensorn `Timer`. `Timer` genererar sedan signaler hela tiden den är aktiv (i detta fall 10 sekunder) och dessa tider vidarebefordras till positionsinterpolatorn `jump`. Värdet som interpolatorn genererar efter varje ny tid den tar emot, vidarebefordras i sin tur till en translationsfunktion som translaterar bollen enligt detta värde och bollen börjar studsas.



Figur 8 – Bollen som studsar på kuberna. Längst upp till vänster i ursprungspositionen och sedan studsar den nedåt till bilden längst ned till höger. Efter det studsar den tillbaka samma väg uppåt.

I detta demonstrationsexempel på ca 30 rader definierar vi alltså upp två kuber, en boll, interaktivitet och rörelse! Och vi har inte ens berört scripter som är det mest kraftfulla medlet att få liv i VRML 2.0. På liknande sätt som rörelser interpoleras kan även mjuka färgskiften och rotationer interpoleras. Har man t ex modellerat ett statiskt rum kan man med VRML 2.0 lägga till funktionalitet så att dörren öppnas när man närmar sig på endast ett par rader.

2.3 På gång inom VRML

VRML 2.0 är fruset för ny funktionalitet och togs upp för standardiseringsgodkännande av ISO (International Standards Organization) 5:e augusti -96. För att hinna slutföra VRML 2.0 innan ISO skulle ta upp förslaget för behandling valde man medvetet att skjuta på en del önskvärd funktionalitet till framtiden. I september eller oktober -96 kommer en VRML 2.1 att presenteras. Det som framförallt är aktuellt i 2.1 är:

- **Externt API.** Ett gränssnitt för kommunikation mellan VRML-världen och "omvärlden" är önskvärt. Ett förslag² till API finns redan och det förslaget verkar ha tillräckligt med stöd för att inkluderas i VRML 2.1.
- **Binärt filformat.** För att få kompaktare filer och därmed kortare överföringstider är ett binärt filformat planerat. Apple som redan har ett väl fungerande binärt filformat till QuickDraw3D har erbjudit sig att tillhandahålla en VRML-anpassad variant av detta. Apple verkar dock ha låg prioritet på arbetet med binärt filformat och presenterade i början av augusti ett utkast på sitt förslag³. Granskas förslaget finner man snart att det är långt ifrån klart och att vissa noder och viss funktionalitet saknas. Från början var det tänkt att just detta förslag skulle bli binärt filformat för VRML 2.0 och att Apple skulle presentera ett *färdigt* förslag runt första augusti. Men då arbetet hos Apple har dragit ut på tiden har nu ett alternativ seglat upp. IBM lanserade 1:a augusti ett eget förslag⁴ till binärt filformat. IBMs förslag baserar sig direkt på ASCII-filformatet. Varje "token" i ASCII-filformatet har ersatts med ett motsvarande i binär form. En komprimeringsalgoritm ingår även i förslaget, vilket saknas i Apples förslag.
- **Scriptspråk.** Detta är en punkt som har skapat mycket hård debatt. Till slut blev Java och JavaScript inkluderade i appendix till VRML 2.0 standarden där det beskrevs hur Java och JavaScript används som scriptspråk *om* något av dessa används. Appendix för andra språk kan läggas till. Det som återstår att bestämma är huruvida man ska satsa på något existerande språk som Java och JavaScript eller om man vill ha ett eget skraddarsytt scriptspråk. Bestämmer man sig för Java eller något annat existerande språk kommer detta att skrivas in i 2.1 standarden. Faller valet på ett helt nytt eget scriptspråk dröjer det antagligen längre innan det inkluderas i standarden.

² http://reality.sgi.com/employees/cmarrin_engr/vrml/externalAPI.html

³ http://quickdraw3d.apple.com/VRML20/vrml_2.0_binary_draft1.html

⁴ <http://www.austin.ibm.com/vrml/binary>

Förutom de ovan nämnda punkterna är VRML 2.0 ett relativt komplett språk för en VR-miljö över Internet. Nya nodtyper kan skapas med prototyper och saknas någon funktionalitet kan nästan allting göras med scripts tillsammans med prototyper. Allting som ligger längre fram i tiden än 2.1 och punkterna ovan är enbart spekulationer. Det finns än så länge inga som helst planer på att inkludera något annat än fleranvändarstöd i en framtida VRML 3.0. Efter att 2.1 är klart diskuteras om VRML ska läggas på is ett tag för att avvakta och se vad som saknas och vad som behöver förbättras när väl VRML 2.x börjar användas. Det är först när verktygen klarar 2.x som VRML kommer att börja användas och erfarenhet fås. Om man ändå ska spekulera lite i vad som saknas ligger följande punkter närmast till hands:

- **Kryptering.** Från konstnärer har framförts önskemål att kunna skydda sina verk. Som det ser ut idag kan vem som helst spara källkoden till objekt för att sedan modifiera och använda dem som man vill. Därför har frågan om kryptering framförts. Hur detta skulle realiseras har inte diskuterats alls, men något krypteringssystem baserat på öppna nycklar kan vara genomförbart.
- **Fleranvändarstöd.** Denna punkt har skjutits ordentligt på framtiden. VRML 2.0 är förberett för inkludering av fleranvändarstöd och utan att ändra på någonting kan tillägg som medger delade fleranvändarvärldar göras. Än så länge har arbetet med att skissa på hur detta fleranvändarstöd skulle se ut inte påbörjats. Anledningen till att inte något av de fleranvändartillägg som finns idag inkluderas i standarden är dels att inget av dessa är skalbara till tusentals användare dels att tilläggen oftast är skraddarsydda för en specifik fleranvändarmiljö. Det kan visa sig att det är svårt att hitta en generell lösning som fungerar bra med alla fleranvändarmiljöer, kan hända är det bättre med ett antal olika tillägg som alla är skraddarsydda till specifika lösningar. Även om man hittar en generell lösning är det inte helt säkert att det inkluderas i standarden, det kan komma att läggas i ett separat dokument. Det mest troliga är dock att någon av de utbyggnader för fleranvändarstöd för VRML 2.0 som kommer att finnas så småningom kommer att användas.
- **Spatial partitionering.** För att kunna hantera riktigt stora världar måste världarna kunna delas upp i mindre delar. Tänker man sig att man har en riktigt stor fleranvändarvärld med kanske 1000-tals samtida användare uppstår nämligen två problem som gör att skalning till oändligt stora världar ej är möjlig. För det första blir världen väldigt stor. Att rendera och uppdatera världen blir en alltför stor börda. Genom att dela upp världen i mindre delar löses detta problem då det räcker med att visualisera de delar av världen som är synliga. Det andra problemet är att kommunikationen mellan alla samtida användare blir mycket omfattande. Om varje användares handlingar skall skickas till varje annan användare av världen ger det snabbt en begränsning på antalet användare. Med spatial partitionering räcker det med att uppdatera de användare som befinner sig i närheten av den användare som t ex förflyttar sig.
- **Objektgenskaper.** Det finns en del egenskaper som skulle vara bra att kunna specificera hos objekt. I mycket realistiska världar eller simuleringar kan det vara nödvändigt med egenskaper som massa, tyngdpunkt, objektets temperatur, objektets friktionsmotstånd, m m.
-

- **Världgenskaper.** Liksom vissa egenskaper kan tillföra realism till objekt finns det egenskaper som det kan vara fördel att införa i världar. T ex vind, regn, temperatur, lufttryck, gravitation m m.
- **Förhandstitt.** Det kan vara önskvärt att kunna förhandstitta på världarna innan de laddas i sin helhet. En stor värld kan ta lång tid att ladda och för att låta användaren se direkt om det var den världen han var ute efter skulle en liten stillbild tagen från världen i något lämpligt perspektiv kunna inkluderas i världen. Denna lilla bild visas sedan direkt när användaren börjar ladda in världen och användaren kan då (oftast) se om det var denna värld han var ute efter *innan* hela världen har laddats.

2.4 Vidare läsning

Då utvecklingen inom VRML-området sker snabbt är det svårt att publicera böcker med färskt material om VRML. Bästa informationskällan är olika sidor på WWW. Ett urval länkar till specifikationer, intressanta världar, VRML-kurser, länksamlingar och annat intressant om VRML finns samlade på <http://www.sisu.se/projects/vrml/links.html>.

3. Verktyg

För att kunna använda VRML behövs olika verktyg. Det som behövs för att kunna betrakta och navigera befintliga VRML-världar är en VRML-läsare. Denna läsare sammankopplas sedan med WWW-bläddraren och aktiveras så fort en VRML-värld påträffas. För att skapa sina egna VRML-världar finns det ett antal olika hjälpmedel. Det finns dels textbaserade utvecklingsmiljöer där världen skapas på ett programmeringsliknande sätt, dels grafiskt baserade utvecklingsmiljöer där världen "ritas" i 3D. Enskilda objekt eller en grund till en VRML-värld kan även skapas i de flesta befintliga 3D-utvecklingsmiljöer då det i de flesta fall är möjligt att antingen direkt spara filer i VRML-formatet eller att spara filer i något format som går att konvertera vidare till VRML.

3.1 VRML-läsare

Det finns idag ett stort antal VRML-läsare som nästan samtliga är fritt distribuerbara. Ett genomgående drag är att det stora flertalet stödjer VRML 1.0 fullt ut och att stöd för VRML 2.0 är utlovat. Tyvärr får man nog säga att ett annat kännetecken för de flesta läsare är att de är ganska svårnavigerade. Jämför man med navigeringen i t ex 3D-spelet Doom är navigeringen oftast betydligt svårare. Ett problem med VRML 1.0 var att de som utvecklade VRML-läsarna ofta inte nöjde sig med den funktionalitet som finns i VRML 1.0 utan gjorde egna tillägg till standarden. Problemet som uppstod var att det inte fanns någon samordning mellan dessa tillägg utan alla gjorde egna tillägg för sina lösningar.

VRML-läsarna har olika preferenser huruvida läsaren skall placeras inuti WWW-bläddraren eller i ett fristående fönster. Båda alternativen är lika vanliga och det går inte att säga att det ena eller andra alternativet är bättre. Används VRML-läsaren i ett separat fönster finns det plats för fler knappar, egna rullgardinsmenyer, m m men nackdelen är att det inte går att ha VRML-världen direkt bredvid texten i ett WWW-dokument samt att frames inte är möjligt (frames är när WWW-sidan delas upp i ett antal fristående rutor som kan samarbeta). Alla WWW-bläddrare klarar inte heller att använda en VRML-läsare inuti sitt eget fönster.

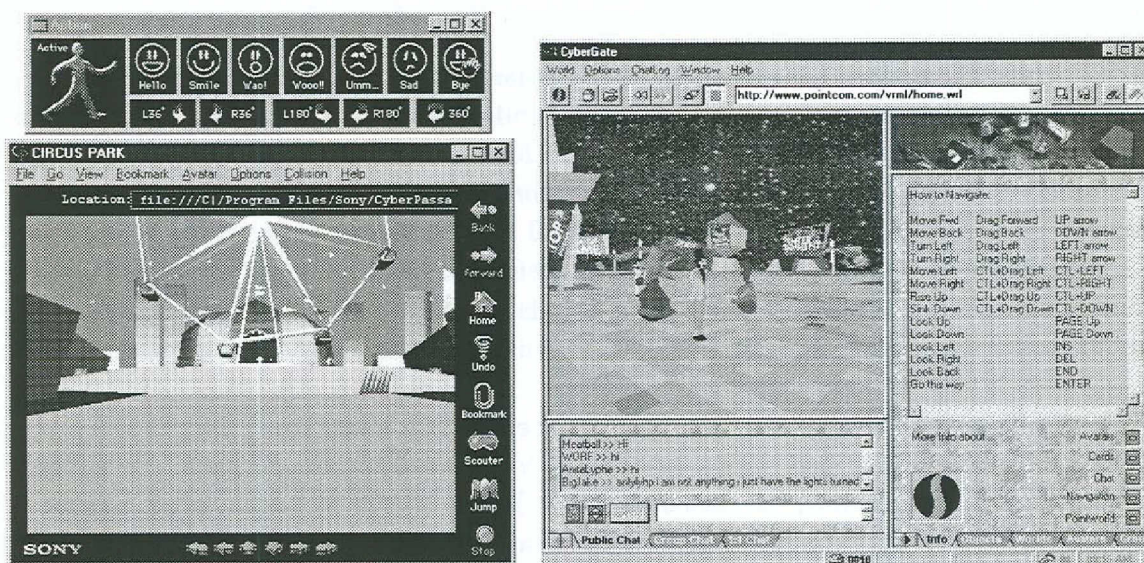
De två mest betydande tilläggen till VRML 1.0 standarden är tilläggen för fleranvändarvärldar och beteenden. Fleranvändarvärldar är ett betydande tillägg även till VRML 2.0 då 2.0 ännu inte har något standardiserat fleranvändarstöd. Världar där flera personer befinner sig samtidigt kan realiseras på två olika sätt. Antingen kan en central server sköta all uppdatering av klienterna eller så kan de enskilda klienterna kommunicera direkt med varandra med hjälp av t ex multicasting.

Det tillvägagångssätt som samtliga VRML-läsare utom Oz Virtual valt är att ha en central server som sköter detta. Antagligen beror övervikten för alternativet med en central server på att man ofta vill tillhandahålla VRML-läsaren gratis men kunna ta betalt för servermjukvaran. I en fleranvändarvärld representeras alla användare av ett objekt som kallas för **avatar**, ett begrepp hämtat ur cyberspace-litteraturen. Med beteenden menas att det finns något språk med vars hjälp det går att beskriva hur objekt i världen ska bete sig. Det kan gälla t ex rörelse, animation, ljud eller att objekt ska reagera på vissa händelser.

I VRML 2.0 används tills vidare Java och JavaScript för beteenden. En skillnad i hur fleranvändarstöd och beteenden realiseras är att fleranvändarstöd nästan uteslutande baseras på kommunikation direkt mellan VRML-läsaren och server för världen (förutom för Oz). Beteenden realiseras genom tillägg till VRML-standarden som tolkas av VRML-läsaren på ungefär samma sätt som läsaren tolkar övriga VRML-filer.

3.1.1 Läsare som klarar både fleranvändarvärldar och beteenden

Ett antal VRML-läsare stödjer både fleranvändarvärldar och beteenden. **CyberPassage** från Sony var först ut med stöd för beteenden och använder något Sony kallar E-VRML för detta. Nyligen lanserade Sony sin Society Server Cyber Passage som är deras server för att hantera fleranvändarvärldar med kommunikation via text. En kul detalj i CyberPassage är att det är möjligt att låta sin avatar göra diverse ansiktsuttryck, låta avataren sova, m m. Navigering med CyberPassage är enkel med hjälp av knappar och man kan följa terrängens kontur m m.



Figur 9 – CyberPassage till vänster och CyberGate till höger.

Black Suns **CyberGate** är en VRML-läsare som är helt inriktad på fleranvändarvärldar. Med hjälp av CyberHub servermjukvara kan man sätta upp fleranvändarvärldar där kommunikationen sker via text och användarna erbjuds möjlighet att antingen välja sin avatar från ett urval av avатарer eller skapa sin egen avatar (max 48 kb stor).

Övrig intressant funktionalitet i CyberGate VRML-läsaren är bl a möjlighet att klicka på ett objekt och bli snabbtransporterad dit, privata konversationer, översiktsvy i fågelperspektiv, möjlighet att få lista på alla avатарer i världen, m m. Om användaren vill kan ett ordentligt användargränssnitt med hjälptexter m m presenteras. En nyttigt detalj är också att CyberGate lämnar bra felmeddelanden vilket gör att den kan vara bra att ha vid felsökning i VRML 1.0 filer.

Black Sun lanserar under sensommaren -96 en vidareutveckling av CyberGate som heter CyberLife. Den är tänkt att användas "ovanpå" Netscapes Live3D VRML-läsare. CyberLife lägger till ett antal ikoner till Live3D-gränssnittet och bygger ut Live3D med fleranvändarstöd via CyberHub. Det både CyberPassage och CyberGate (samt troligtvis CyberLife) saknar vad gäller fleranvändarstöd är framförallt möjlighet att dela objekt, t ex finns ingen möjlighet för en användare att ge en annan användare ett dokument i VRML-världen.

AlphaWorld är en mycket välbesökt och populär fleranvändarvärld för att prata med varandra i virtuella rymden. AlphaWorld baseras på VRML+ som är VRML 1.0 tillägg för fleranvändarvärldar och beteenden. Under sensommaren eller hösten -96 kommer Worlds Inc att presentera efterföljaren till AlphaWorld under namnet **Gamma**. Gamma är en utbyggd VRML 2.0 med ordentligt fleranvändarstöd i form av möjlighet att ha upp till 500 avatarer i varje rum, oändlig skalbarhet av världarnas storlek, delade tillstånd (t ex dela filer eller objekt och samarbeta med varandra), säker autentisering och inloggning, loggning av vad användare gör i världen, möjlighet att leta efter andra användare av världen i en databas, användare kan modifiera världen, möjlighet att ta betalt, världskaparen kan göra egna animerade avatarer åt användarna, etc.

WorldServer-3D används för att hantera fleranvändarvärldarna och världar med 50 samtida användare ska gå att köra på en vanlig PC med Pentium 100 MHz processor. Det skall också vara enkelt att lägga till ytterligare funktionalitet i form av moduler från andra leverantörer, WorldServer-3D kommer att levereras med stöd för elektronisk handel och röst. Gamma kommer att vara realiserat så att vilken VRML 2.0 kapabel läsare som helst ska kunna hantera Gammavärldar fullt ut. All extra funktionalitet sköts av Javakod som laddas ner med världen. Worlds Inc kommer även att lansera Active Worlds som ska vara motsvarigheten till Gamma fast anpassad till Internet Explorer och dess ActiveX kontroller och Visual Basic scripter.

V-Realm Browser av Integrated Data Systems är en VRML-läsare som antingen kan användas fristående eller inuti WWW-bläddraren. V-Realm 3D Media Server är den server som används för V-Realms fleranvändarvärldar. V-Realm stödjer de flesta utbyggnader av VRML 1.0 samt en del extra funktionalitet för fleranvändarstöd. Användare kan dela objekt och filer med varandra i världen och de kan kommunicera med antingen text eller ljud.

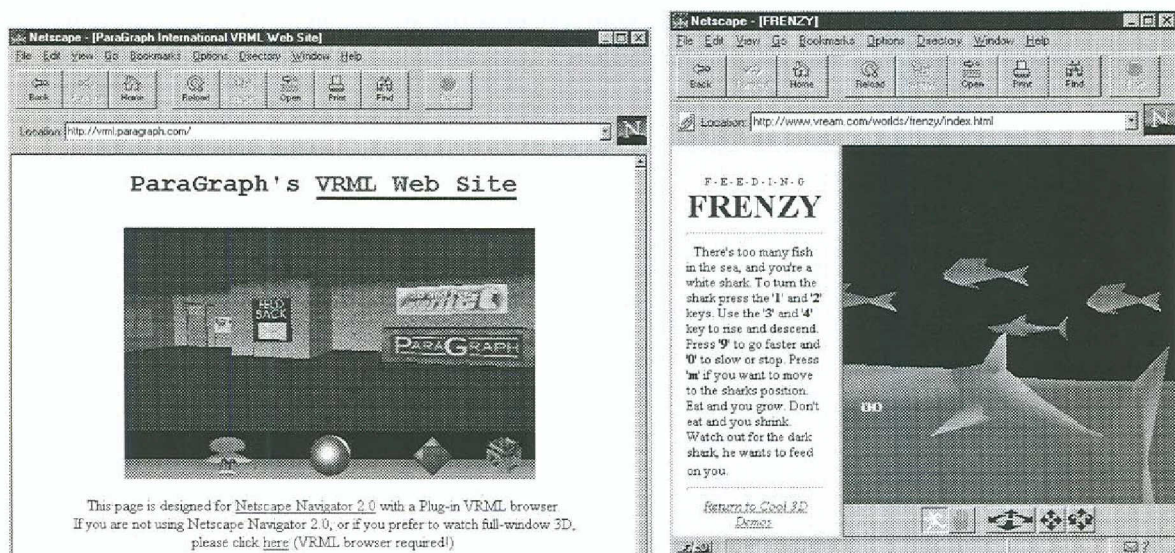


Figur 10 – V-Realm Browser till vänster och Oz Virtual till höger.

Oz Virtual VRML-läsare har skräddarsytt i olika varianter. Förutom den "vanliga" VRML-läsaren finns det **Play3D** som är en läsare för barn och **Oz Virtual For Conferencing** som är en variant för VR-konferenser. Konferensomgivning kan väljas från ett antal olika konferensmallar och sedan skickas automatiskt e-post till alla inbjudna deltagare med lösenord för att kunna delta i konferensen. Mötesrummen har ett antal olika konferensfunktioner inbyggda, t ex röstning, fildelning, automatisk generering av mötesprotokoll i WWW-format, privat eller publik kommunikation med text eller röst samt fullt stöd för kommunikationer med rörliga bilder i realtid. I **Oz Virtual** kan avataren väljas från ett urval eller skapas med **Oz Avatar Editor**. **Oz** har valt att inte ha en server för att hantera fleranvändarvärldarna, utan kommunikationen sker mellan klienterna. Valet att inte ha en server innebär också att det inte endast är speciella världar som stöds av en viss server som får fleranvändarstöd. Två eller flera användare som har **Oz Virtual** kan tillsammans besöka vilken VRML-värld som helst på Internet och ha fleranvändarfunktionalitet i den världen utan något stöd från den dator som världen finns på. En ytterligare variant, **Oz Virtual for Games**, är planerad.

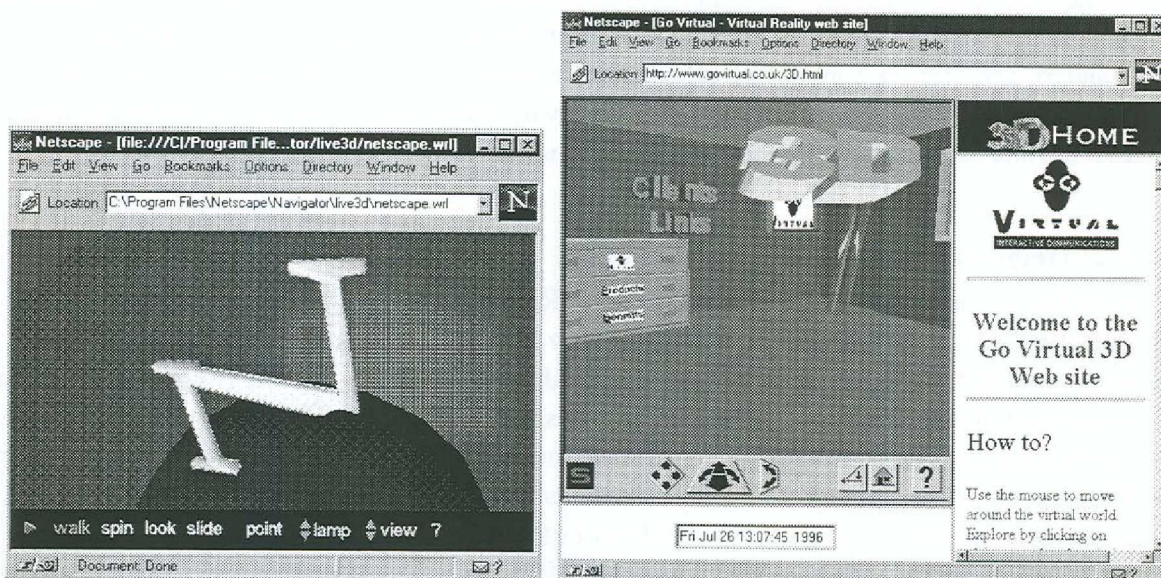
3.1.2 Övriga intressanta VRML-läsare m m

De två läsare som ligger längst fram i stöd för VRML 2.0 är föga förvånande **Cosmo Player** och **CyberPassage**. **Cosmo Player** är utvecklat av **Silicon Graphics** som var den drivande kraften bakom VRML 2.0 förslaget och som hjälpte till att skapa VRML 1.0 genom att bidra med en bantad och något modifierad version av sitt **OpenInventor** filformat. Förutom **Silicon Graphics** var det **Sony** som gjorde den största insatsen i utvecklandet av VRML 2.0 och **Sonys** VRML-läsare **CyberPassage** som tidigare nämnts är även den långt framme i stöd för VRML 2.0. Ingen av dessa två stödjer VRML 2.0 fullt ut ännu men ny 2.0 funktionalitet läggs till kontinuerligt. Förutom dessa två annonserade **WIRL** av **VREAM** med VRML 2.0-stöd, under **Siggraph** mässan i augusti -96. **WIRL** har tidigare baserats på **VREAMScript** som var ett mycket omfattande scriptspråk för beteenden m m, men har nu anpassats till VRML 2.0 standarden.



Figur 11 – CosmoPlayer till vänster och WIRL till höger.

Då alla läsare tolkar färger m m aningen olika kan det vara fördel att kontrollera hur ens världar ser ut i de populäraste VRML-läsarna. Den mest använda läsaren är **Live3D** från Netscape. Live3D stödjer de flesta tilläggen förutom fleranvändarvärldar och levereras tillsammans med Netscape Navigator 3.0. Eftersom Netscape har varit en av de drivande krafterna när det gäller att få Moving Worlds till 2.0 standard vill de givetvis att deras egen VRML-läsare ska stödja 2.0 och detta jobbar Netscape nu för fullt med. Microsofts **VRML Add-In** är en ganska blygsam VRML-läsare för Internet Explorer som stödjer VRML 1.0 och ett par tillägg men torde ändå vara relativt välanvänd.



Figur 12 – Live3D till vänster och den mycket lättnavigerade Viscap till höger.

SuperVRML är en utbyggnad av VRML 1.0 för beteenden, animationer och 3D-ljud. Många av de mer avancerade och omfattande VRML-världar som finns på Internet har skapats med SuperVRML tilläggen och det är **Viscape** som används för att läsa dem. Viscap används inuti Netscape och är den mest lättnavigerade läsaren.

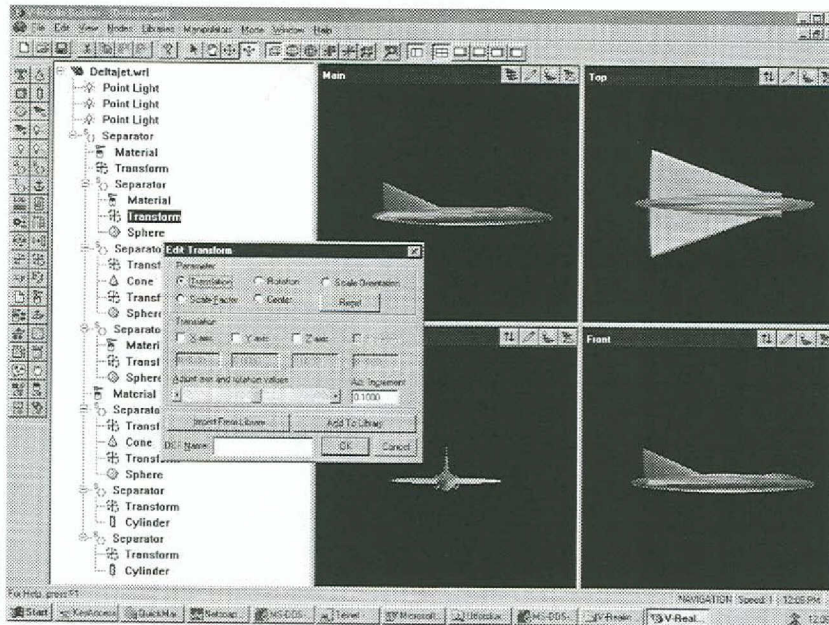
Produkt	Kommentar
AlphaWorld v.87 av Worlds Inc http://www.worlds.net/alphaworld	Populär fleranvändar chat-värld som använder VRML+.
Cosmo Player v1.0 beta 1a Silicon Graphics http://webspacesgi.com/cosmoplayer	VRML 2.0 läsare.
CyberGate v1.0 beta 3 av Black Sun http://www.blacksun.com/products/cybergate.html	Läsare för fleranvändarvärldar.
CyberLife beta av Black Sun http://www.blacksun.com/products.html#cyberlife	Vidareutvecklingen av CyberGate som ska användas ovanpå Live3D för fleranvändarstöd.
CyberPassage v2 beta 2 av Sony http://vs.sony.co.jp/VS-E/vstop.html	VRML-läsare som klarar VRML 2.0, E-VRML och Sonys fleranvändarvärldar.
Gamma av Worlds Inc http://www.worlds.net/tooltech/products.html	Efterföljaren till VRML+ och AlphaWorld. VRML-tillägg för fleranvändarstöd.
Live3D beta 1 av Netscape http://www.netscape.com/comprod/products/navigator/live3d/index.html	Netscapes VRML-läsare.
Oz Virtual beta av Oz http://www.oz-inc.com/products/ozvirtual/index.html	Fleranvändarläsare som finns i olika skräddarsydda varianter förutom den vanliga läsaren.
Pueblo v2.01 av Chaco http://www.chaco.com/pueblo	VRML MUD klient.
Topper beta 2 av Kinetix http://www.ktx.com	VRML-läsare med stöd för Kinetix beteendespråk VRBL.
V-Realm Browser v1 av Integr. Data Systems http://www.ids-net.com	Läsare med fleranvändarstöd.
Viscape v1.06 av Superscape http://www.superscape.com	Läsare för SuperVRML världar.
VR Scout v av Chaco http://www.chaco.com/vrscout	Finns både som plugin och fristående läsare.
VRML Add-In v2.0 av Microsoft http://www.microsoft.com/ie/addons/vrml.htm	Microsofts VRML-läsare.
VRweb v1.2 av IICM, NCSA och University of Minesota http://hyperg.iicm.tu-graz.ac.at/Cvrweb	VRML 1.0 läsare som distribueras med full källkod och kan modifieras för egna syften.
WebView v1.0 beta 1 av SDSC http://www.sdsc.edu/EnablingTech/Visualization/vrml/webview.html	1.0 läsare med full källkod.
Whurlplug v1.0d6 av Apple http://quickdraw3d.apple.com	3DMF och VRML 1.0 läsare för Macintosh Power PC. Använder QuickDraw 3D för rendering.
WIRL v2.0 av VREAM http://www.vream.com/3wirl.html	VRML 2.0 läsare med stöd för beteendespråket VREAMScript.

3.2 Utvecklingsmiljöer för VRML

Att göra sin egen VRML-värld kan vara ganska mödosamt. Att göra det "för hand" utan verktyg är fullt möjligt men både svårt och arbetskrävande. Det som gör det svårt är framförallt att man rör sig i tre dimensioner. Att beskriva 3D-rotationer m m utan hjälp av verktyg är mycket omständligt. Ett av målen med VRML 2.0 var att standarden skulle göra det så enkelt som möjligt för verktygsutvecklare. I och med detta skall det inte behövas några som helst expertkunskaper om hur VRML 2.0 fungerar för att skapa VRML 2.0 världar. De flesta befintliga utvecklingsmiljöer stödjer ännu bara VRML 1.0, fler och fler deklarerar dock att stöd för 2.0 är på gång eller redan klart.

Det finns två olika kategorier av verktyg som man kan ta till sin hjälp för att skapa VRML-världar. Den första är program som är direkt skapade enbart för VRML. Av dessa finns det endast ett fåtal och de är oftast inte särskilt kraftfulla. Den andra kategorin är program som är skapade för andra ändamål än VRML, men som har möjlighet att exportera sina objekt till VRML-filformatet. Det finns numera också ett antal olika konverteringsprogram som konverterar scener från diverse 3D-filformat till VRML. Nackdelen med de verktyg som ej är skapade direkt för VRML är att vissa saker ej går att uttrycka i en vanlig 3D-utvecklingsmiljö, t ex WWW-länkar, samt att de ofta är *för* kraftfulla med en mängd funktionalitet som inte går att överföra till VRML. Antagligen kommer de flesta större 3D-utvecklingsmiljöer att stödja VRML till viss mån i framtiden. Men de som är mest intressanta är de verktyg som är inriktade mot VRML.

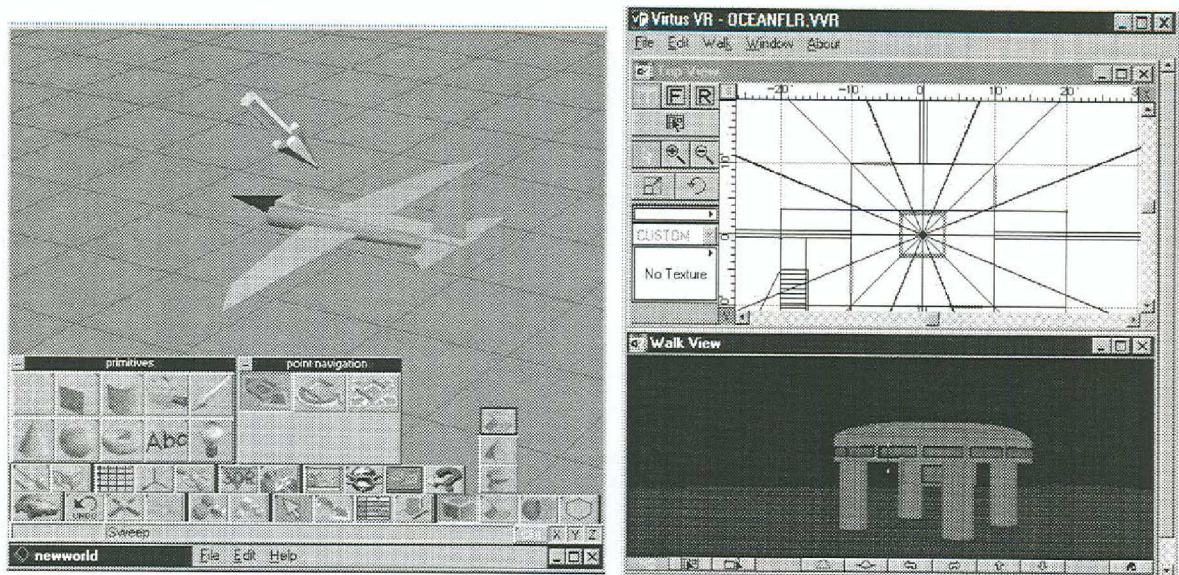
Det finns idag en handfull olika renodlade VRML utvecklingsmiljöer. Valet av miljö är inte enbart en smaksak, utan också beroende av kunskapsnivån vad gäller VRML. I och med att VRML är ett filformat/språk som det är fullt möjligt att konstruera världar för hand med, finns det verktyg som byggt vidare på den principen. **V-Realm Builder** från Integrated Data Systems är ett mycket bra exempel på detta. Principen är här att man har en hierarkisk struktur med de olika noderna som världen består av, kallad scenograf. För att lägga till eller modifiera objekt får scenografens struktur eller nodernas parametrar modifieras. Resultatet visas hela tiden i en visualisering av världen. Fördelen med detta tillvägagångssätt är att man får exakt kontroll över allting i världen som skapas. Nackdelen är att man förlorar en del av de avancerade funktioner som en mera grafisk baserad miljö kan erbjuda och att man förutsätts veta hur alla VRMLs nodtyper fungerar. Version som stödjer VRML 2.0 är utlovad.



Figur 13 – 3D-konstruktion med V-Realm Builder.

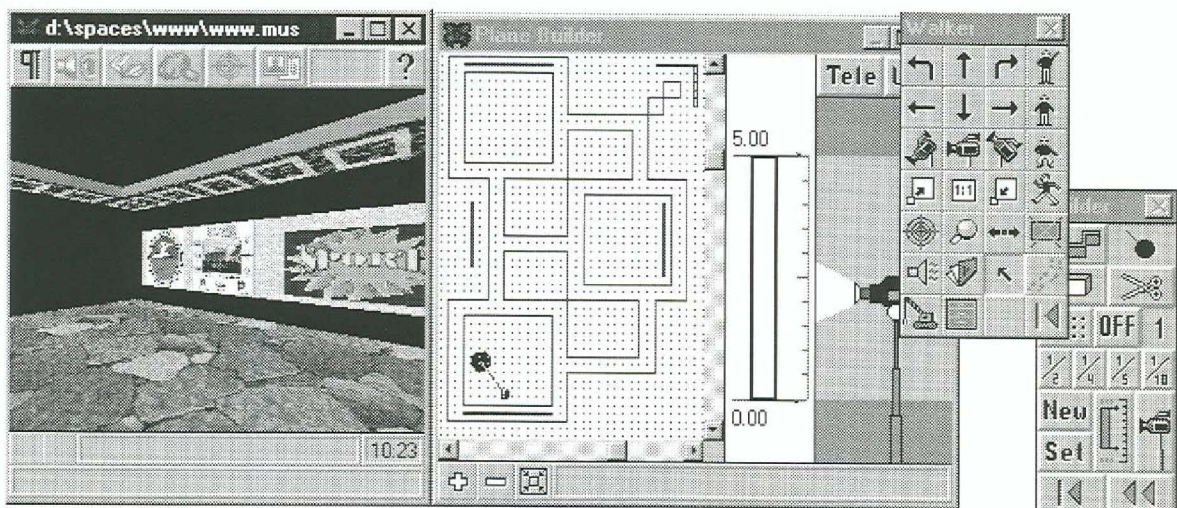
Föredrar man en utvecklingsmiljö som är helt grafiskt baserad och inte förutsätter några VRML-kunskaper finns det ett antal att välja mellan. Caligaris **Pioneer** stödjer all 1.0 funktionalitet och har förutom de vanliga funktionerna, som att placera ut och rotera/flytta/skala objekt, en hel del extra funktionalitet för att mycket enkelt utföra avancerade objektmanipulationer och dessutom klarar Pioneer av 3D-ljud. Enda nackdelen är att allt styrs med musen och att allt modelleras grafiskt i 3D. Om man har ett objekt och vill skala om det exakt längs någon axel kan det vara svårt med musen. Att placera ett objekt med precision bredvid ett annat är också ganska svårt i en 3D-rymd. Det finns heller ingen rullgardins- eller text-menju för editeringskommandon, utan enbart en rad med ikoner.

Det medföljer ett antal mycket bra hjälpexempel som bygger objekt medan man ser på för att lära sig hur och vad man kan göra. Hjälpfunktionen inuti Pioneer är också mycket bra. Man klickar på hjälp-knappen och sedan på den knapp man vill ha hjälp med. Pioneer startar då en demonstration där det visas hur objekt manipuleras med de funktioner som finns med den knappen samtidigt som förklarande text visas. Helhetsomdömet är att detta är en mycket bra VRML-editor så fort man lärt sig att hantera den trots att det kan vara svårt att manipulera med objekten i 3D.



Figur 14 – Pioneer till vänster och Virtus Walkthrough Pro till höger.

Walkthrough Pro från Virtus har något mindre funktionalitet när det gäller komplexa objektmanipulationer men är i gengäld aningen lättare att använda för att skapa vissa enklare objekt som t ex rum och byggnader då objektmanipulationerna inte sker i 3D. I Walkthrough finns det två olika typer av fönster, dels en 3D-representation av världen och dels en eller flera 2D-representationer av världen sett längs någon axel. 2D-representationen är en parallellprojicering och består enbart av polygoner vilket gör att skapandet av objekt liknar det tillvägagångssätt arkitekter använder för att rita 3D-figurer på papper. Omdömet blir att det är lättare att skapa världar i Walkthrough än i Pioneer, men viss funktionalitet saknas.

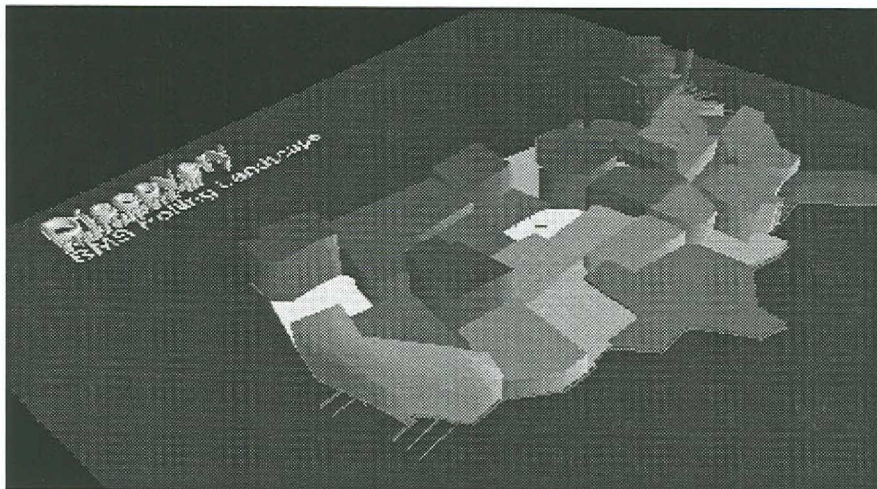


Figur 15 – Virtual Home Space Builder.

ParaGraphs **Virtual Home Space Builder** (VHSB) är en annan helt grafisk baserad VRML utvecklingsmiljö som nyligen släpptes i en VRML 2.0 version. Med VHSB är det mycket enkelt att skapa en viss typ av världar. Tillvägagångssättet i VHSB är att man hela tiden bygger med boxar. För att göra t ex ett rum ritar man först en stor box, klipper sedan ut en box mitt i och till sist klipper man ut boxar för fönster och dörrar. Liksom i Walkthrough ritar man i 2D och har en visualisering av objektet i 3D. Att grovt modellera t ex sitt eget kontorsrum tar inte mer än ett par minuter och inläringstiden är minimal. Omdömet blir att VHSB är mycket bra och väldigt lättanvänd men beroende på vilka världar som ska skapas kan den vara alldeles för begränsad.

Utvecklingsmiljöer som ännu inte finns tillgängliga men verkar intressanta är ParaGraphs **Internet3D Space Builder** (ISB) och WIRLs **VRCreator 2.0**. ISB är en utvecklingsmiljö för VRML 2.0 som enligt ParaGraph ej har några begränsningar i världskapandet liknande de VHSB har. Enligt ParaGraph är ISB ett alternativ till VHSB för den som vill skapa avancerade VRML-världar på ett lika enkelt sätt som med VHSB. Stämmer detta påstående är ISB ett mycket intressant alternativ för VRML-utveckling. VRCreator är en VRML 2.0 utvecklingsmiljö som är grafiskt baserad och har avancerad funktionalitet liknande Pioneer för manipulation av objekt. Både VRCreator 2.0 och Internet3D Space Builder kommer att finnas tillgängliga under hösten -96. Under senhösten -96 kommer ParaGraph även att släppa ett antal verktyg för VRML 2.0 utveckling. Internet3D Avatar Builder för att skapa avatarrer, Internet3D Behaviour Builder för att lägga till beteenden till objekt samt Internet3D Multi-User Server som är en server för att hantera fleranvändarvärldar.

Beroende på syftet med VRML-världen kanske en statiskt skapad värld inte är tillräcklig. Ibland vill man ha en dynamiskt skapad värld som genereras när världen efterfrågas. Vill man undvika CGI-program (Common Gateway Interface; program skrivna i "valfritt" programmeringsspråk för att generera WWW-dokument) finns det idag olika verktyg till ens hjälp. **Distributed Discovery** av Visible Decisions är en objektorienterad utvecklingsmiljö för att framförallt visualisera databaser tredimensionellt. Discovery genererar en VRML-värld bestående av tredimensionella tabeller och grafer dynamiskt. Den genererade världen kan även manipuleras interaktivt med t ex en 3D-"slider" för att förändra variabelvärden.



Figur 16 – Distributed Discovery genererad scen av ett tänkt valresultat i USA.

Ett antal databashanterare och fristående program erbjuder möjligheten att skapa HTML sidor dynamiskt med hjälp av mallar. Denna mall ser ut som en vanlig HTML-sida men kompletterad med HTML-liknande kommandon för SQL databasanrop, variabler och repetitioner. Mallen fylls sedan i med resultatet från databasanropet och slutresultatet blir en dynamiskt skapad HTML-sida som reflekterar informationen i databasen. En produkt som erbjuder exempel på hur mallar kan användas för att dynamiskt generera VRML-världar är **Cold Fusion**. Men även andra produkter som stödjer mallbaserad skapning av HTML-sidor bör gå att använda för VRML-utveckling utan förändringar.

Dimension X har skapat ett mycket intressant toolkit för VRML 2.0 vid namn **Liquid Reality**. Det är skrivet helt i Java och består av 250 Javaklasser. Enligt företaget bakom Liquid Reality ska det i princip vara möjligt skapa exakt den VRML 2.0 baserade applikation man vill med Liquid Reality. Ett företag som använder VRML för något ändamål välkomnar säkerligen möjligheten att få styra exakt hur kontroller och navigering fungerar. Det enda som krävs från klientsidan är att detta toolkit är installerat. Själva Javakoden, som t ex kan vara en skraddarsydd VRML-läsare, laddas sedan ner med VRML-världen. Stödjer i nuvarande versionen ej VRML 2.0 fullt ut, men det är på gång inom kort.

Silicon Graphics har två produkter för att skapa VRML 2.0 världar. **Cosmo Create 3D** är en utvecklingsmiljö för VRML 2.0 som tyvärr bara finns tillgänglig för Silicon Graphics egna datorer. **Cosmo 3D** är ett toolkit för utveckling av VRML 2.0 applikationer som påminner om Liquid Reality. Cosmo 3D har ett API för Java med klasser och metoder som hanterar all VRML 2.0 semantik. Cosmo 3D kan användas för att läsa, skriva och visualisera VRML 2.0 filer och är plattformsoberoende i och med att utveckling med Cosmo 3D sker i Java.

Utöver dessa renodlade VRML-utvecklingsmiljöer och verktyg finns det redan nu ett antal olika konverteringsprogram för att konvertera 3D-objekt från diverse populära filformat till VRML. Värt att nämna är DoomToVrml2 som konverterar WAD-världar (används av Doom m fl) till VRML 2.0. Det finns ett flertal fritt distribuerbara bra verktyg för att skapa och editera WAD-världar. Dessutom kan man räkna med att de flesta större 3D-utvecklingsmiljöer kommer att lägga till funktionalitet för att exportera objekten till VRML-format. Nackdelen med dessa utvecklingsmiljöer är dock att de inte har möjlighet att uttrycka VRML-specifika saker som t ex WWW-länkar, men de kan vara mycket lämpade till att skapa enskilda objekt som sedan kan importeras i någon VRML-editor. Även om man har en 3D-utvecklingsmiljö som inte går att konvertera/exportera till VRML-formatet kan man antagligen överföra den till något filformat som sedan antingen går att konvertera vidare till VRML eller importera i en VRML-editor.

Produkt	Kommentar
3D Studio MAX av Kinetix http://www.ktx.com/products/01max/	Variant av 3D Studio som är inriktad på VRML-utveckling.
Cold Fusion v1.5 av Allaire http://www.allaire.com	Mallbaserad dynamisk skapning av HTML och VRML-världar.
Cosmo produkterna av Silicon Graphics http://www.sgi.com/Products/cosmo	Cosmo Create 3D för att skapa VRML-världar. Cosmo 3D är ett Java toolkit.
CyberKit av Black Sun http://ww3.blacksun.com/products/cyberkit.html	VRML-utvecklingsmiljö. Släpps antagligen under hösten -96.
CyberPassage Conductor v1 beta 3b av Sony http://vs.sony.co.jp/VS-E/vstop.html	Verktyg för att lägga till E-VRML beteenden och ljud till VRML-världar.
Distributed Discovery av Visible Decisions http://www.vdi.com/Products/disco.vrml.html	Objektorienterad dynamisk skapning av VRML-världar.
DoomToVrml2 av Silicon Graphics http://vrml.sgi.com/tools/doomtovrml2/index.html	Konverterar WAD-världar till VRML.
Ez3d v2.0 av Radiance Software http://www.webcom.com/~radiance	Blandning av grafisk och textbaserad VRML-utveckling.
G Web av Virtual Presence http://www.demon.co.uk/presence	VRML-utveckling baserad på scen grafen.
HyperWire beta av Kinetix http://www.ktx.com	Javautveckling för VRML-världar.
Internet3D Space Builder av ParaGraph http://www.paragraph.com	VRML 2.0 utvecklingsmiljö.
Liquid Reality av Dimension X http://www.dimensionx.com/products/lr/index.html	Javabaserad utveckling av VRML 2.0 program.
Pioneer v av Caligari http://www.caligari.com	3D-grafikbaserad VRML-utveckling.
Spinner v1.0 av 3D-Web http://www.3dweb.com	Skapar VRML-världar genom att importera 3D-objekt samt scener och lägga till WWW-länkar och beteenden.
V-Realm Builder 1.0 av Integr. Data Systems http://www.ids-net.com/ids/builder/builder.html	VRML-utveckling baserad på scen grafen.
Virtual Home Space Builder v2.0 ParaGraph http://www.paragraph.com/vhsb	Mycket lättanvänd men något begränsad VRML 2.0 utvecklingsmiljö.
Virtus Walkthrough Pro v2.5.1 av Virtus http://www.virtus.com/vwtpro.html	Grafiskt baserad VRML-utveckling.
VRCreator v2.0 av VREAM http://www.vream.com/3vrc.html	VRML-utvecklingsmiljö. Ej släppt ännu.
VRT v4 av SuperScape http://www.superscape.com/ProductsVRT.html	SuperVRML-utveckling med stöd för beteenden m m.
WorldToolKit Direct av Sense8 http://www.sense8.com	Avancerad VRML utvecklingsmiljö. Släpps under slutet av -96.
http://ftp.sunet.se/pub/pc/games/idgames/docs/editing/dmedit10.zip	Översiktsbeskrivning av WAD-editorer.

3.3 Alternativ till VRML

VRML är inte det enda alternativet för den som vill realisera VR över Internet och WWW. Ett antal befintliga VR-system försöker nu anpassa sig till Internet samtidigt som det idag finns ett par alternativ till VRML som har utvecklats enligt samma koncept. Studien av dessa icke VRML-baserade VR-system är ej komplett och informationen bygger enbart på information hämtad från Internet.

Dive är ett VR-system utvecklat av SICS i Kista som är mycket populärt och spritt, framförallt inom VR-forskning. Målet med Dive är att åstadkomma en så fullständig och omfattande VR-miljö som möjligt. Man försöker att undvika begränsningar och eftersträvar att likna den verkliga världen så långt det går samt dessutom lägga till den extra funktionalitet som bara är möjlig i en VR-värld. Objekt i Dive-världar kan ges ett beteende via Dive/Tcl script. Användaren ser världen från sin Dive-avatars ögon och det är möjligt att koppla till head-mounted display (HMD). Användarna har även möjlighet att använda datahandskar för att gripa och hantera objekt i världen. Dive går att använda över Internet, men då kommunikationen är mycket omfattande är det inte lämpligt i praktiken idag. Dive är dessutom mycket beräkningskrävande. D v s både kraftfulla datorer och snabba kommunikationskanaler behövs. Från en Dive värld kan dokument hämtas över HTTP och FTP via ett WWW-gränssnitt. För närvarande stödjer Dive VRML 1.0, men stöd för 2.0 är planerat liksom en integrerad WWW-bläddrare.

Apple var tidigt ute med sin QuickTime teknik. QuickTime lät användaren bygga "nästan" 3D miljöer genom att t ex komponera ihop ett antal fotografier. QuickTime fick mycket uppmärksamhet i pressen då den kom. Sommaren -95 släppte Apple **QuickDraw 3D**, ett icke-VRML-baserat VR-paket för *konstgjorda* 3D världar. QuickDraw 3D är ett API för att skapa, och visualisera 3D-objekt i realtid. Om man ska jämföra VRML och QuickDraw3D och bortser från funktionaliteten kan man säga att VRML är ett språk för att beskriva en 3D-omgivning och QuickDraw3D är ett språk, en 3D-omgivning och en läsare. För närvarande finns QuickDraw 3D endast till Macintosh, men ska komma till Windows (95/NT) sommaren -96 enligt Apple. Det filformat som QD3D använder kallas 3D Metafile Format (3DMF) och en modifierad variant av detta är ett av två förslag på binärt filformat till VRML 2.0. Apple, som hade ett eget förslag till VRML 2.0 baserat på QuickDraw3D teknik, var ovilliga att stödja Moving Worlds VRML 2.0 i sina produkter till en början. Men under våren -96 deklarerade Apple att de kommer att stödja VRML 2.0.

ActiveX Animation av Microsoft är ännu ej lanserad. ActiveX Animation baserar sig på Microsofts förslag till VRML 2.0 standard vid namn ActiveVRML. Microsoft vägrade acceptera att Moving Worlds förslaget blev utsett till VRML 2.0 standard och deklarerade tidigt att de ej tänkte stödja Moving Worlds VRML 2.0 utan skulle satsa på sitt eget ActiveVRML. I augusti -96 stod det dock klart att Microsoft kommer att stödja VRML 2.0 och att de kommer att omarbete ActiveVRML och lansera det senare mot slutet av -96 under namnet ActiveX Animation.

ActiveVRML baserade sig på funktionell programmering och tidsberoenden. Alla objekts beteenden beskrevs med tidsberoenden och ville man byta beteende på ett objekt fick man byta tidsberoende, t ex kunde ett objekt ha ett tidsberoende före och ett efter användaren

klickar på det. Beträktade man specifikationerna för ActiveVRML och VRML 2.0 verkade de ganska olika. Men skillnaden i funktionalitet, d v s vilka världar som kunde skapas, var marginell. Det finns ännu ingen dokumentation om ActiveX Animation, men det kommer att vara en vidareutveckling och ombearbetning av ActiveVRML som är skräddarsydd för interaktiva 2D- och 3D-animationer.

Spline (Scalable Platforms for Internet Environments) är ett experimentellt VR-system av Mitsubishi. Inriktningen på Spline är att tillhandahålla fleranvändarmiljöer för t ex utbildning och spel som ska vara skalbara till ett mycket stort antal användare. Spline har liksom VRML valt att reducera bandbreddsbehovet till ett minimum för att kunna köras över Internet. Med Splineteknik är det möjligt att ha röstkommunikation och användare av världen kan förändra den interaktivt. Mitsubishi har använt sin Splineteknik för att realisera en omfattande fleranvändarvärld vid namn Diamond Park. Förflyttning i Diamond Park sker huvudsakligen genom att använda en cykel för att komma närmare en helkroppsupplevelse. Vill deltagarna inte cykla kan de ta en av de bussar som trafikerar området. Ett antal datorstyrda robotar vandrar omkring och hjälper deltagarna till rätta i världen.

Play3D, **Web-Active** och **WebXpresso** är VR-system som alla påminner mycket om VRML. 3D-världen beskrivs i form av geometriska objekt. Denna beskrivning överförs sedan till användarens dator som i sin tur visualiserar den aktuella världen åt användaren. Play3D är en Netscape plugin vars funktionalitet inkluderar animeringar, beteenden, WWW-länkar, möjlighet att byta färg, form, ytegenskaper eller transparens i realtid. WebXpresso har enkla beteenden, animeringar samt möjligheten att uppdatera världarna från klientsidan genom manipulation av objekten. Web-Active har liksom VRML 1.0 endast statiska objekt och WWW-länkar.

VR System	Kommentar
ActiveX Animation av Microsoft http://www.microsoft.com	Vidareutveckling av ActiveVRML för interaktiva 2D- och 3D-animationer.
Dive v3.0.13 av SICS http://www.sics.se/dce/dive/dive.html	Av SICS i Kista. Spridd VR-miljö som nu anpassar sig till distribuerad VR över Internet med stöd för HTML och VRML.
QuickDraw3D v1.06 av Apple http://quickdraw3d.apple.com	VR-miljö som skiljer sig åt från VRML framförallt genom att den är ett språk, en 3D-omgivning och en renderer medan VRML bara är ett språk för att beskriva 3D-världar.
Play3D v1.42 av Heads Off http://HeadsOff.com	Liknande VRML. Animeringar, beteenden, WWW-länkar m m.
Spline av Mitsubishi E. R. L. http://www.merl.com/spline/Welcome.html	Experimentellt VR-system inriktat på distribuerade fleranvändarvärldar.
Web-Active v1.0.1b6 av Plastic Thought http://www.3d-active.com	Skapa 3D-objekt och rotera samt betrakta dem över Internet/WWW.
WebXpresso v1.1 av DataViews http://www.dvcorp.com/webxpresso/	Liknande VRML. Kan editera världarna från klientsidan. Har animeringar, beteenden och WWW-länkar.

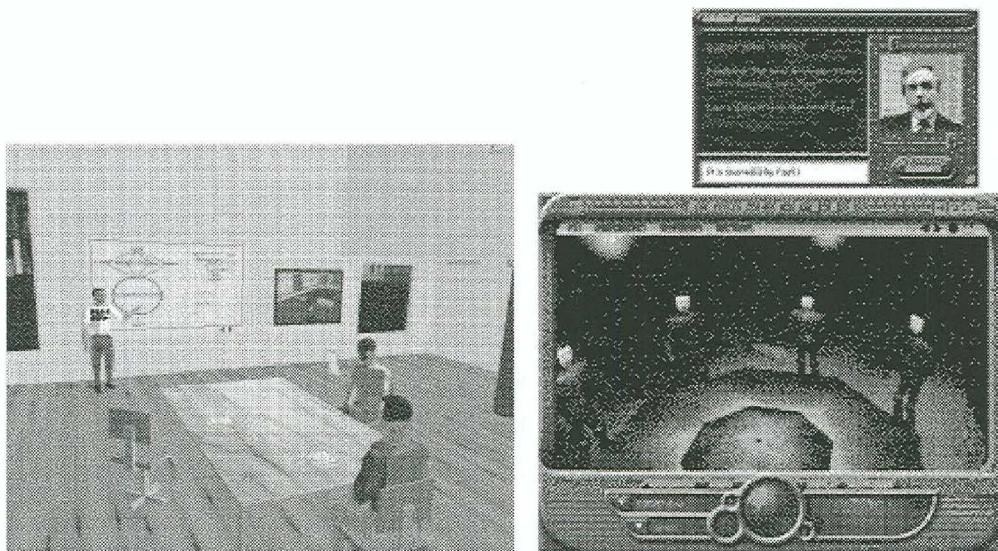
4. Applikationsområden

Att någonting är "inne", "hett", "jättebra", o s v spelar egentligen ingen större roll om det inte finns någon användning för det. Frågan är då: Finns det några egentliga användningsområden för VRML och kommer VRML att användas?

Eftersom VRML endast används i mycket begränsande omfattning idag går det inte att göra någon fullständig lista över VRMLs användningsområden. VRML kan användas till i princip allting. Några exempel på användningsområden där VRML med fördel kan användas är gruppssamarbete, visualisering, försäljning, underhållning och informationstjänster. Dessa beskrivs nedan.

4.1 Gruppssamarbete

Ett av de områden som ofta brukar nämnas när man diskuterar vad man kan göra med VR är möjligheten att ha konferenser i virtuella mötesrum. Man vill kunna ha en konferens mellan deltagare som sitter fjärran från varandra i sina egna kontor. De ska kunna diskutera, presentera saker för varandra och kanske gå fram till en annan deltagare och ge denne ett dokument. VRML med vissa utbyggnader och rätt verktyg är mycket lämpat för detta. Det finns redan idag VRML-produkter för detta ändamål. Se framförallt Oz Virtual som har ett skraddarsytt verktyg för just denna typ av konferenser. Gamma-plattformen av Worlds Inc verkar också ha all den funktionalitet som krävs. Det som behövs för att realisera detta är en fleranvändarvärld med ganska omfattande fleranvändarstöd. Man bör ha funktionalitet för att kontrollera vilka som deltar i konferensen, möjlighet att dela objekt, deltagarna måste kunna kommunicera med varandra (gärna med levande ljud och bild) m m.



Figur 17 – DIVE konferens till vänster och Oz Virtual for Conferencing till höger.

Andra applikationer inom gruppssamarbete kan vara att t ex dela ett gemensamt virtuellt arbetsrum med dokument m m. Detta arbetsrum skulle kunna byggas ut så att användarna t ex ser vilka som befinner sig i rummet och på ett ungefär vad de gör. Vid distansarbete kan det lätt bli så att man känner sig lite ensam och ett gemensamt arbetsrum för distansarbetare skulle kanske ta bort lite av den känslan. Vill användarna diskutera något med någon annan i rummet kan en privat eller publik diskussion via text eller röst startas. Det behöver inte heller vara så att detta rum måste användas för att utföra saker, det kan t ex användas huvudsakligen för att åskådliggöra vad distansarbetarna gör för tillfället. Är t ex skärmläckaren på hemma hos distansarbetaren kan en signal skickas till rummet som låter den avataren sätta sig i fikahörnet o s v.

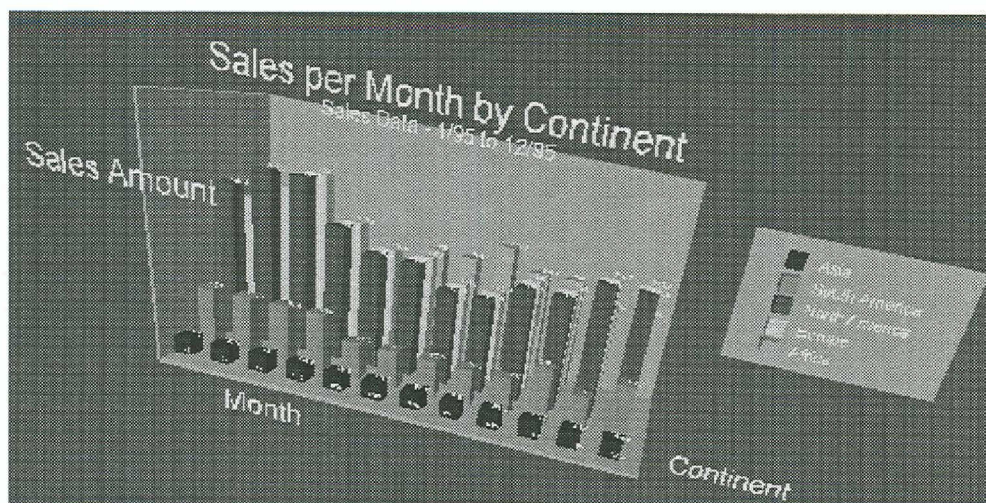
4.2 Visualisering

Antag att en svensk byggfirma för nyckelfärdiga hus har ett antal olika typhus som kunderna kan välja mellan. Firman skulle kunna generera sina hus i VRML-kod. Antagligen kan de göra det direkt från sina CAD-program i framtiden. Den intresserade husspekulanten kan sedan hemifrån betrakta de olika typhusen från alla håll och kanter. Husfabrikanten har antagligen också ett antal varianter av varje typhus med diverse tillval, t ex veranda eller balkong. Dessa tillval skulle kunden sedan kunna jämföra mot standardutförandet. Kunden kan "testa" vilken färg huset gör sig bäst i o s v. Sedan kan kunden gå direkt in till husfabrikanten och säga exakt vad han vill ha – vilket sparar tid och pengar för fabrikanten och kunden blir nöjdare!



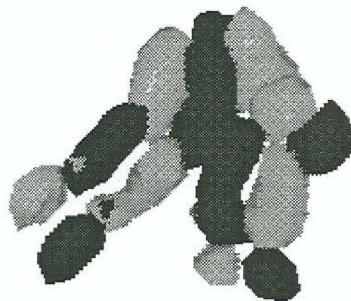
Figur 18 - Utbyggnaden av San Franciscos flygplats modellerades med olika designalternativ med VR-teknik innan den byggdes.

Det är ofta önskvärt att kunna presentera statistik och liknande på Internet. Viss information modelleras bäst i två dimensioner, medan annan information modelleras bäst i tre. Tabeller över information kan ofta kombineras ihop och åskådliggöras i ett enda betydligt mer åskådliggörande diagram i 3D. Dessutom kan man tänka sig att manipulera med tabellerna interaktivt för att, utan att gå genom sitt kalkylprogram, direkt få svar på frågor som t ex "Om vi ökar priset på den produkten med 10%, hur ser det ut då?".



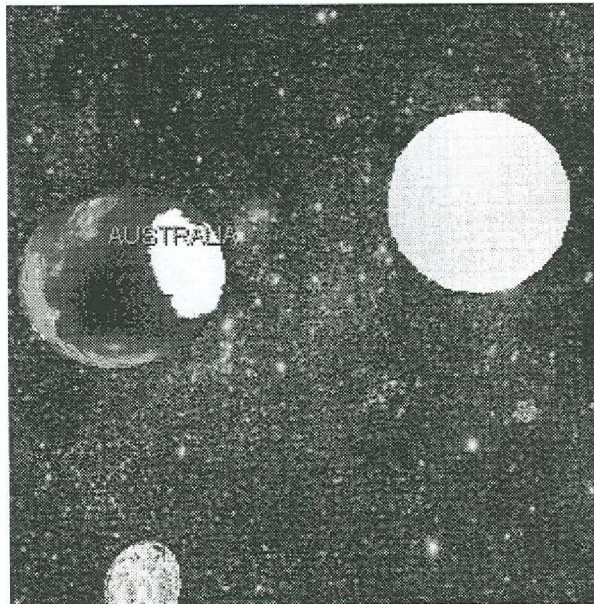
Figur 19 – Cold Fusion genererad 3D-statistik.

Alla kemiska formler och molekyler är något som med fördel skulle kunna realiseras i 3D. Redan idag finns vissa kemiska föreningar modellerade med VRML. Kemielever skulle säkerligen dra fördel av att kunna se och manipulera med kemiska föreningar m m i 3D. Fördelen med VRML är dessutom att det räcker med att *en* person gör en VRML-modell och lägger ut denna på Internet. Sedan kan intresserade koppla upp sig mot den sidan och betrakta modellen.



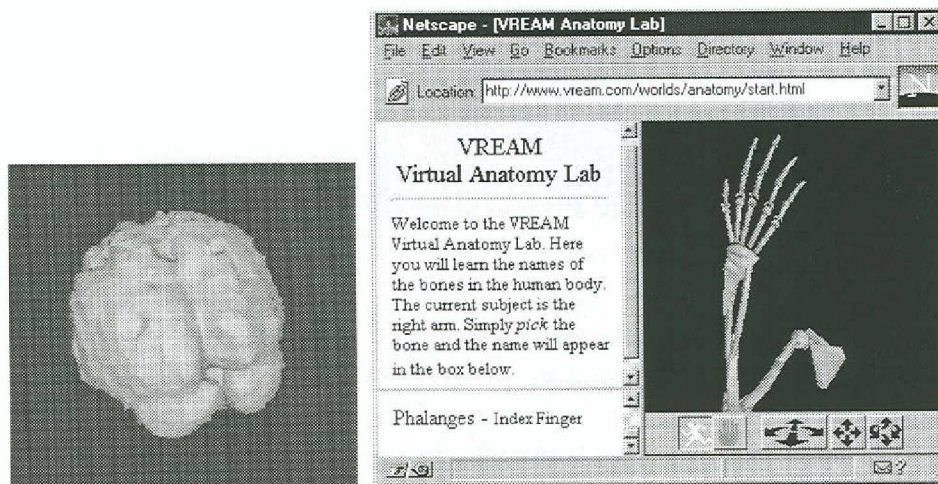
Figur 20 – Kemisk förening i VRML.

Redan idag finns bl a vårt solsystem som VRML-världar på Internet. Det mesta inom astronomin skulle kunna modelleras med VRML och släppas för allmän kännedom på Internet. Med VRML 2.0 behöver man t ex inte heller begränsa sig till en fast orörlig bild av ett solsystem, utan alla planeter kan röra sig i sina respektive banor runt solen och betraktaren skulle t ex kunna klicka på de olika planeterna för mera information. Man behöver inte heller nöja sig med planeter, månar kan cirkulera runt planeterna samtidigt som planeterna cirkulerar runt solen o s v.



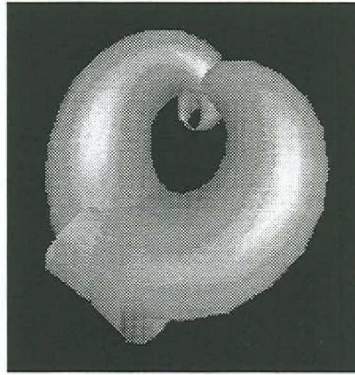
Figur 21 – En del av vårt solsystem i VRML. Här pekar man just nu på Australien med muspekaren.

Människans kropp skulle kunna modelleras som ett VRML-objekt. Beträktaren skulle sedan kunna manipulera kroppen på en mängd olika sätt. Användaren skulle t ex kunna välja ett visst objekt från kroppen och manipulera vidare med det. Eller om den avbild som ofta finns i dagens läkarböcker används. Det är ett antal genomskinliga plastsidor med kroppsdelar på och när man bläddrar kommer man successivt längre in i kroppen. I VRML kan det först finnas en bild på en människa och sedan "skalas" lager efter lager av. Först tittar man på huden för att sedan titta på blodkärlen och musklerna o s v.



Figur 22 – VRML-model av en människas hjärna till vänster och en "anatomiläktion" till höger.

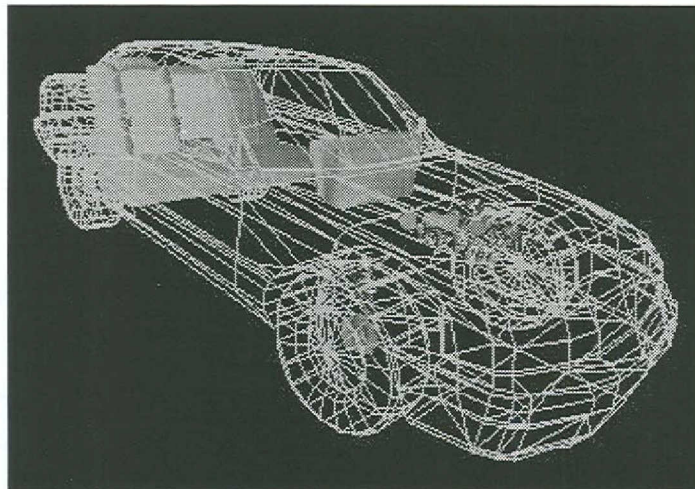
Många matematiska formler och samband presenteras bäst i 3D. Det finns redan nu ett antal matematiskt relaterade VRML-världar, t ex fraktala 3D-landskap. Vid presentation av matematiska uppsatser på Internet kan VRML användas för att bättre åskådliggöra de olika matematiska sambanden och resonemangen. Kanske kan ett litet script animera åskådliggörandet för att ännu bättre visa sambanden mellan olika variabler och vad som händer när värden förändras.



Figur 23 – Matematisk modell.

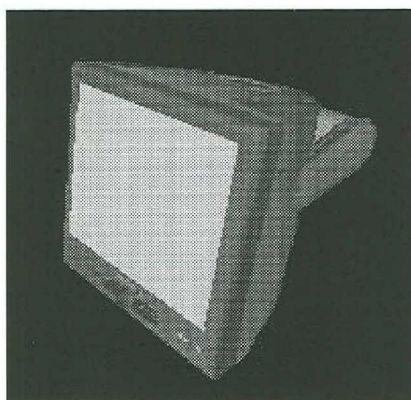
4.3 Försäljning

På samma sätt som en byggfirma kan lägga ut hus på Internet kan ett bilföretag som t ex Volvo låta alla sina bilmodeller med alla fakta och i alla möjliga färgvarianter finnas tillgängliga för allmänheten på Internet. Kunden kan redan innan han kommer till bilaffären välja färg m m.



Figur 24 – Toyota Lexus har avbildat sina modeller i VRML. Här Lexus LS400.

Framförallt i USA är postorder över Internet en stor marknad. Anledningen till att det inte slagit igenom fullt ut i Sverige beror antagligen på att kreditkort används i betydligt mindre utsträckning här. Antag att ett av dessa postorderföretag skulle VRML-modellera sig själva som en vanlig butik. Kunden går in i butiken med sin avatar och får möjlighet att bli teleporterad direkt till den avdelning som denne är intresserad av. Där finns det, liksom i verkligheten, hyllor med produkter. Runt omkring går andra kunder och kanske försäljare, också i form av avatarer. Eventuellt kommer en försäljare fram och frågar om han kan hjälpa till med något. Han kanske visar hur butikens söksystem fungerar. Du har möjlighet att plocka fram saker från hyllan och se diverse information om dem, kanske finns det en länk till en kundservice VRML-värld hos fabrikanten. Detta tillvägagångssätt kombinerar det bästa med att handla i verkligheten (kunna gå i affären och titta, få hjälp av personalen, etc) med det bästa från de vanliga Internet postorderföretagen (sökning, hoppa snabbt mellan avdelningar och produkter, etc).

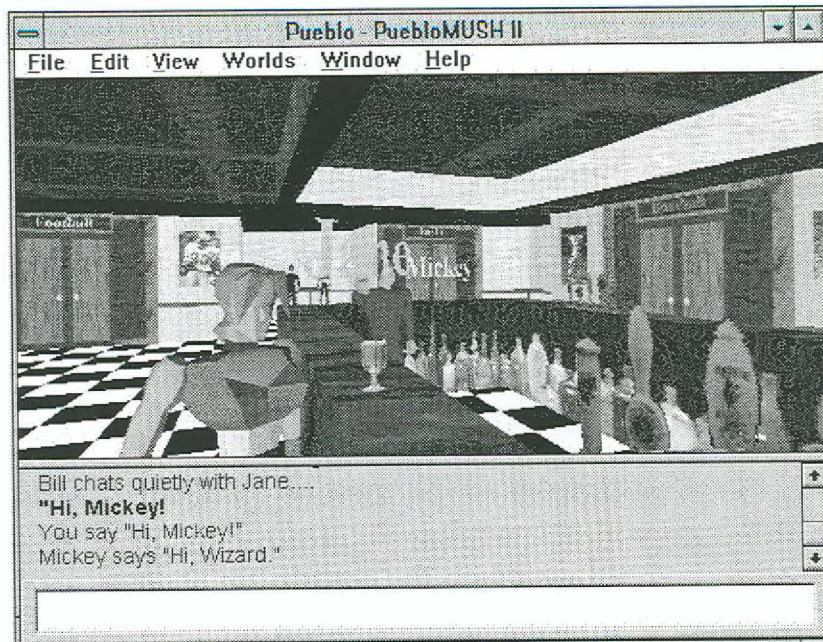


Figur 25 – Grundig har avbildat ett antal av sina produkter i VRML.

Vissa exempelprodukter har redan nämnts. Men troligtvis skulle de flesta produkter som tas fram och presenteras på Internet kunna modelleras helt eller delvis med VRML. Beträktaren blir garanterat inte mindre intresserad av en 3D-bild av en produkt än en motsvarande 2D-bild.

4.4 Underhållning

Spel kommer antagligen stå för en stor del av VRML-applikationerna. Med VRML och dess utbyggnader blir det i teorin möjligt att realisera spel som t ex Doom över Internet. Det finns redan en del VRML-spel och inom kort kommer det säkerligen en hel del fler. Pueblo är en VRML-läsare som är skraddarsydd för MUD-spel. MUD, som står för Multi User Dungeons, har tidigare varit textbaserade fleranvändar rollspel där spelaren befinner sig i en fantasivärld och spelar med/mot alla andra deltagare. MUD är ett gissel på många datautbildningar då många datastudenter börjar spela MUD och blir helt bitna av det. Syftet med Pueblo är att ge MUD-spelen ljud, 2D- och 3D-grafik, avatrar och bättre chat-möjligheter. Mark Pesce, som var med och skapade den första VRML versionen, har sagt att han är imponerad av Pueblo och att det var det här han hade i bakhuvudet när han startade med VRML. Med VRML kan plattformsoberoende spel utvecklas och sedan distribueras utan kostnad för tillverkaren, något som borde vara av intresse för alla spelproducenter.

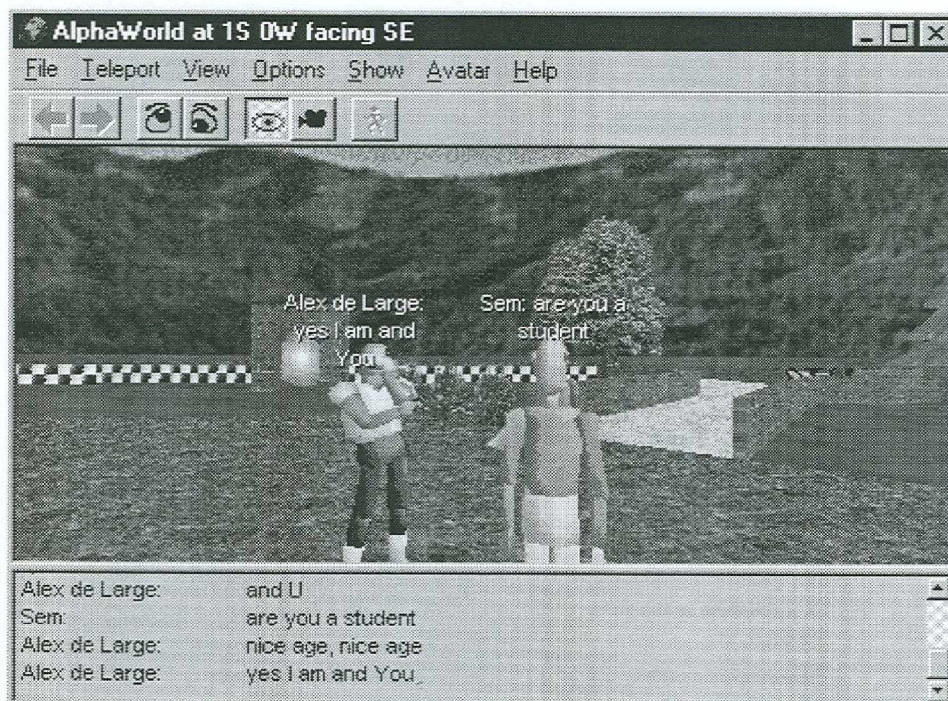


Figur 26 – Pueblo MUD med VRML-gränssnitt.

Chatvärldar är Internets motsvarighet till heta linjen. När det gäller VR-chatvärldarna väljer användaren en avatar och träder in i en 3D-värld som det finns andra avatarer i. Deltagarna småpratar och umgås. För många deltagare blir detta ett ställe där de träffar många nya bekanta och tillbringar mycket tid. AlphaWorld är en mycket populär chatvärld, där man försökt efterlikna den verkliga världen så mycket som möjligt. Med VRML+ tilläggen kan användare välja en av ett par olika avatarer att bli representerad av i världen, himlen och markens utsträckning mot horisonten syns, animeringsfunktionalitet finns, m m. Kommunikationen mellan de olika deltagarna i AlphaWorld sker via textkommunikation (typ pratbubblor).

Förutom att umgås och prata med andra avatarer kan deltagarna även bygga sin egen bostad. Det gäller bara att lyckas hitta en obebyggd plats, vilket oftast betyder att man måste teleportera sig en bra bit utanför centrum eftersom världen är ganska ordentligt bebyggd. En mediebevakad händelse tilldrog sig i AlphaWorld våren -96. Två personer som hade träffats i just AlphaWorld gifte sig i denna chatvärld⁵. De hade ritat en egen virtuell brud och brudgum med en egen virtuell bröllopsklänning etc. Worlds Inc's VD var präst och alla övriga avatarer var vittnen. Detta illustrerar ganska bra hur pass allvarligt deltagarna i chatvärldar tar på den virtuella verkligheten som dessa världar erbjuder. Förutom AlphaWorld finns det ett antal chatvärldar. Med nya VRML 2.0 och dess utbyggnader kommer säkerligen detta område att få ökad popularitet. AlphaWorld världen kommer att överföras till Gamma plattformen så småningom och då kallas GammaWorld.

⁵ <http://www.wmc.com/hotnews/worldweb.html>



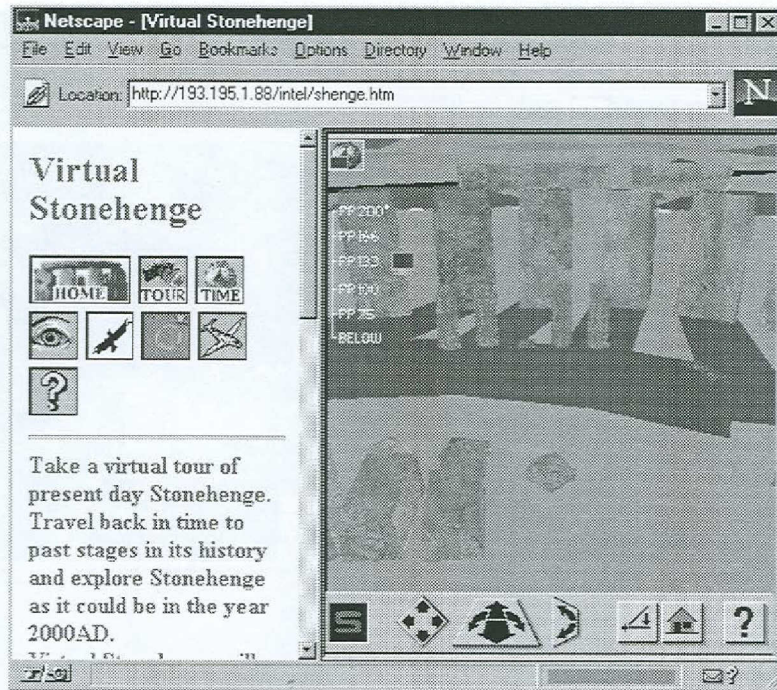
Figur 27 – Bild från AlphaWorld där två avатарer står och småpratar med varandra.

4.5 Informationstjänster

Det finns redan ett par VRML-gallerier med konstverk på Internet. T ex kan ett museum skapas och tavlor placeras ut. Andra tredimensionella objekt som skulpturer kan också avbildas. Varför inte återskapa ett helt galleri eller museum med hela dess innehåll?

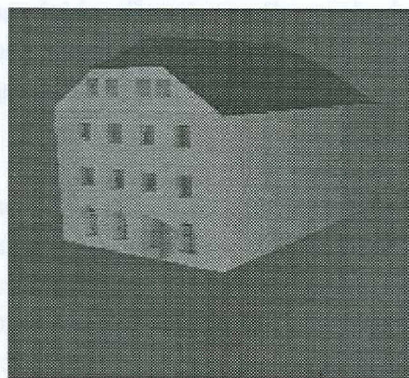
Informationskiosker som finns vid vissa muséer och evenemang skulle med fördel kunna realiseras över Internet med VRML.

Mayaruiner finns idag som en VRML-värld. Men man behöver inte begränsa sig till enbart mayaruiner. Det mesta inom historia skulle kunna modelleras med VRML. T ex Egyptens pyramider och Sfinxen med kringinformation skulle säkert vara intressanta objekt för allmänheten. Begränsningarna på hur långt det går att komma i att avbilda den gångna tidens stora byggnadsverk m m bestäms enbart av fantasin och hur mycket möda världskaparen är beredd att lägga ner på avbildningen.



Figur 28 – Stonehenge avbildat i VRML.

Med VRML kan företag som komplement till den vanliga hemsidesinformationen avbilda ett helt kontor. Varje anställd kan få konstruera sitt eget kontor. Ett script skulle kunna kontrollera ifall den anställde är påloggad på det lokala nätverket och i så fall t ex låta lampan vara tänd på den anställdes kontor. På dörren till varje kontor skulle det kunna finnas information som anger vem som har rummet, rumsnummer, telefonnummer m m.



Figur 29 – IICM:s byggnad med alla kontorsrum.

Många företag erbjuder idag sina kunder hjälp per telefon. Men ofta räcker kanske inte ord till för att förklara. Man vill kanske göra en enkel teckning eller visa ett dokument. Med hjälp av VRML, dess utbyggnader och vissa verktyg skulle det gå att ha en help-line över Internet. Företagen kan ha en VRML-värld med information om produkterna och låta ett antal avatarer som representerar hjälppersonalen befinna sig i världen. När en användare sedan får problem kan de följas åt till de HTML/VRML-sidor som beskriver lösningen på problemet. Med hjälp av t ex Gamma kan man kanske även ha röstkommunikation mellan personerna och låta hjälppersonalen överlämna dokument och filer åt kunderna. Detta skulle för multinationella företag också innebära att det räcker med *en* help-line. Företagen behöver inte längre sätta upp en help-line i varje land de är representerade i.

4.6 Vanliga sidor

Naturligtvis måste VRML inte användas till någonting "nyttigt". Den största användningen av VRML kommer antagligen inte att vara "nyttig användning". VRML kommer antagligen att användas flitigt för att göra bättre och mer attraktiva hemsidor. Både privata och kommersiella hemsidor kan med fördel göras mer effektfulla med VRML. Idag använder många diverse Java-applets för att få sina hemsidor bättre och roligare. I framtiden kommer säkerligen VRML att vara ett spritt komplement till Java-applets.



Figur 30 – En SISU logo i VRML.

4.7 Kommer VRML att användas?

Det som talar för VRML är en hel del. För det första så kan användargränssnitten bli mer lättförståeliga. Metaforer från verkliga livet är lättare att komma ihåg och använda för att de är enklare att förstå. Ett bra konstruerat gränssnitt med VRML ger användaren en realistisk bild att lagra i minnet istället för ett antal kommandon eller liknande. Detta kan jämföras med t ex dataspel som kan användas intuitivt av de flesta utan att behöva några omfattande manualer eller kurser.

För det andra finns det vissa saker som i princip inte går att realisera på ett vettigt sett utan 3D. Hur skulle t ex applikationen med virtuella konferenser realiseras utan 3D? Andra saker modelleras bättre i 3D än i 2D, t ex skulle applikationsexemplet med hustillverkaren säkerligen kunna realiseras i 2D men resultatet blir betydligt bättre i 3D. Vidare så är man inte alltid enbart ute efter ren information. Det är ingen slump att det i t ex dagstidningar finns bilder som åskådliggör texten. 3D-bilder är att gå ännu ett steg i förklarande bilder till text. Sedan finns det också nöjessidan.

Det finns ett antal "mötesplatser" på Internet. I Sverige finns t ex Passagen. Dessa lever på att användarna finner sidorna tilltalande, både informations- och utseendemässigt. 3D-bilder gör att dessa sidor blir mera utseendemässigt attraktiva. En liten chatvärlds-hörna i t ex Passagen skulle troligtvis också locka ännu fler besökare.

Det finns huvudsakligen två punkter som talar emot VRML. För det första blir det komplext att skapa bra, intuitiva världar. En HTML sida kan vem som helst skapa mycket snabbt, en VRML sida kräver mera. Ett av målen med VRML 2.0 är att det inte ska behövas några expertkunskaper om VRML för att skapa världar. VRML ska vara gjort så att programvaruföretagen kan konstruera lättanvända verktyg för detta. Faran är om verktygen inte skulle bli tillräckligt lättanvända. Då kommer arbetsinsatsen för att skapa VRML-sidor vara så stor att det inte är mödan värt att använda VRML. För det andra så kan det visa sig att användarna inte tycker om VRML-gränssnittet. Att navigera i en 3D-värld kan vara mera intuitivt och lättförståeligt än ett HTML-gränssnitt, men det tar i många fall också längre tid att använda.

Detta var enbart en översiktlig skiss av exempel på applikationer där VRML kan användas samt fördelar och nackdelar. Men som med det mesta är det svårt att sia om framtiden. Idag används VRML endast marginellt men förhoppningarna är att fördelarna med VRML-världar ska vara så stora att VRML kommer att användas i stor omfattning. Frågan är inte *om* VRML kommer att användas, utan i vilken omfattning det kommer att användas.