

## Sammanfattning

Detta projekt bygger vidare på ett system, Webscoop, som genomfördes som D3-projekt föregående läsår. Användaren presenterar några intressanta länkar för Webscoop och kommer därefter att få liknande länkar på sin personliga hemsida. Användaren kan även ge feedback på dessa för att Webscoop ska få mer information om användaren, och därmed kunna göra bättre sökningar.

Årets projekt har lagt till en 3D-värld och försökt förbättra analysen av sidor så att användaren med större sannolikhet får rätt information. Själva 3D-världen är skriven i Virtual Reality Modelling Language (VRML) och ihopkopplingen med gamla Webscoop har skett med CGI och Python. För att förbättra analysen har projektet jobbat med eXtended Markup Language (XML).

Rapporten behandlar gamla Webscoop, 3D-layouten och XML-delen. Den innehåller även information om några andra områden som studerades innan projektet bestämde sig för 3D-världen och XML.

## Förord

Projektkurs D är en obligatorisk del av Civilingenjörsutbildningen i Datateknik på Chalmers Tekniska Högskola. Kursen går under utbildningens tredje år och dess huvudsakliga mål är att stärka teknologens tilltro till sin förmåga att lösa ingenjörproblem.

Projektmedlemmarna har varit Cecilia Arvidsson, Yavuz Danis, Keyvan Seyedi Honarvar, Håkan Johansson och Johan Vettefors. Projektet har drivits på avdelningen Datavetenskap under höstterminen 2000 och vårterminen 2001. Handledarna för projektet har varit Philippos Tsigas och Håkan Sundell.

# 1 Sammanfattning av projektets kronologi

I början av september gav handledarna en genomgång av vad projektet skulle handla om. Vi skulle bygga vidare på ett projekt, kallat Webscoop, som genomfördes förra läsåret. Av handledarna fick vi följande förslag på förbättringar som kunde göras:

- ⑩ Personlig layout.
- ⑩ Koppling till användarens kalender.
- ⑩ Användning av användarens bakgrund för att Webscoop skall hitta sidor som är mer intressanta för användaren.
- ⑩ Ta hänsyn till veckodag, så att man får olika information beroende på vilken dag det är.
- ⑩ Rubrik till WAP - om användaren vill ha mer information om det hon läser på WAP-browsersn kan hon ange detta, det vill säga feedback kan ges precis som i "vanliga" Webscoop.
- ⑩ Använda XML för att förbättra analysen av nya sidor.

På ett efterföljande möte i projektgruppen fördelade vi projektets olika roller mellan oss och bestämde att vi skulle rotera roller vid jul, något vi inte gjorde. Efter att ha funderat lite över de olika förslagen på förbättringar kom vi fram till att vi skulle fokusera på följande 3 punkter:

- ⑩ Layouten.
- ⑩ Rubrik till WAP.
- ⑩ Kalenderkopplingen.

Vi bestämde oss för att nya delar i projektet inte fick påbörjas förrän tidigare delar blivit klara och testats. Efter att ha börjat skissa på layouten och jobbat en del med den insåg vi i mitten av november att det här området inte var det vi ville fokusera på. Framför allt hade vi inte kommit på någon idé till layout som vi tyckte var tillräckligt intressant att bygga på. Vi bestämde oss därför för att ändra projektets inriktning. Då flera av projektets medlemmar var intresserade av artificiell intelligens beslöt vi att försöka förbättra systemet i detta avseende. Handledarna gav oss artiklar om detta för att vi skulle skaffa oss en uppfattning om vad vi ville göra, och som gav oss en del nya idéer. En idé föddes om att göra en 3D-layout som avspeglade AI:n. Denna idé förverkligades inte, utan istället utvecklades en 3D-värld vars utseende inte var kopplat till Webscoops beteende. Efter att några av projektets medlemmar hade tittat på olika tekniker för artificiell intelligens under jullovet kom vi fram till att vi inte skulle förbättra systemets intelligens då det verkade vara för svårt. Inte heller XML verkade vara någonting att satsa på då det inte verkade finnas några sidor som använde XML. Vi beslöt att satsa på Wap och 3D-layouten. Efter möte med handledarna beslöt vi att fortsätta med XML, medan WAP-delen av projektet lades på hyllan. För vidare information om 3D-layouten och XML-delen se avsnitt 3-4 och 5.

## **2 Webscoop förra året**

Projektet "Personalized Electronic Newssystem" är ett D3-projekt som har löpt över två år. Förra året resulterade det i ett system som fick namnet Webscoop, och som årets projektgrupp har haft till uppgift att vidareutveckla. Denna del av rapporten är en kort beskrivning av förra årets projekt och är till för att ge en överblick över detta system.

Webscoop är en personlig nyhetsportal och innehåller länkar som användaren är intresserad av. Dessa länkar hittar Webscoop automatiskt och det enda användaren behöver göra är att visa Webscoop vilka sidor hon tycker är bra och att ge programmet feedback på de sidor systemet funnit.

Webscoop kan delas upp i fyra huvudkomponenter, proxyservern, analysdel, crawler samt en del som används för att visa resultatet för användaren.

### **21 Proxyserver**

När användaren surfar via Webscoop, registrerar nya sidor eller följer länkar som rekommenderats av Webscoop går all denna trafik via proxyservern. En följd av detta blir att proxyservern har som uppgift att se till att denna information når de komponenter som behöver den. Proxyservern används på begäran av användaren (även om användaren kanske inte alltid är medveten om att hon kommit med en sådan begäran).

### **22 Crawler**

Crawlerns uppgift är att gå igenom länkar (med utgångspunkt från länkar som getts av proxyservern) och ser till att länkarna (och text från dem) finns tillgängliga för analysdelen i gemensamt minne. Crawlern körs kontinuerligt, oberoende av andra komponenter.

### **23 Analysdel**

Både länkar som registrerats till proxyservern av användaren och som funnits av crawlern analyseras här. Analys av de länkar användaren registrerar ger som följd att användarens profil uppdateras. Användarprofilen sparas i databas i form av ett antal poängsatta ordlistor, där varje användare har egna sådana ordlistor. Analys av länkar från crawlern sker med utgångspunkt från den profil varje användare har, och används för att avgöra om ett dokument är tillräckligt intressant för att visas för användaren ifråga. De sidor som bedöms som mest intressanta sparas i databas för att senare kunna visas för användaren. Analysverktyget körs kontinuerligt, oberoende av andra komponenter.

### **24 Publisher**

Denna del bygger upp den hemsida, inklusive layout, som användaren ser. Detta gör den med hjälp av den information som sparats i databas i analyssteget. Publishern möjliggör även för användaren att ge feedback på systemets val av sidor. Denna feedback skickas, via databasen, till analysdelen så att ordlistorna uppdateras. Publishern används på begäran av användaren, då användaren skapas, loggar in, uppdaterar sina inställningar och då användaren vill ta bort sitt konto.

## 3VRML-layout

### 31Inledning

Idag är så gott som alla sidor på internet i två dimensioner och projektet ville därför pröva ett nytt och spännande koncept, 3D-hemsidor. Därför gjordes en 3D-sida i Virtual Reality Modelling Language (VRML) och förhoppningarna är att 3D-sidan underlättar för användaren att skilja mellan tre olika intressen. VRML-sidan ska även ge användaren och projektmedlemmarna en tankeställare av fördelar och nackdelar med 3D-sidor på nätet.

Kommunikation med databasen och med analyzern sker med hjälp av CGI och Python. I det här stadiet av projektet gjordes enkla tester av detta, där resultatet matas ut till en enkel textfil, för att se att idéerna var riktiga.

### 32Vad är VRML?

Virtual Reality Modeling Language (VRML) är en standard för 3D-sidor på internet. VRML har funnits sedan 1995 och projektet har använt den senaste versionen, VRML97. Att programmera i VRML har stora likheter med att bygga HTML-sidor. Man skriver hur man vill ha det och resultatet syns direkt på skärmen. Men, till skillnad från HTML, krävs en plugin till läsaren och i projektet har vi huvudsakligen använt Cosmo Player<sup>1</sup>. Fördelarna med Cosmo Player är att den är gratis, enkel att installera, välkänd samt beprövad. Den enda nackdelen är att den inte fungerar i Unixmiljö, men detta löste sig eftersom projektmedlemmarna hade tillgång till Windowsmaskiner.

VRML97 släpptes 1997 och det kommer att krävas minst en ny version innan VRML blir ett bra alternativ till HTML. VRML saknar mycket av den funktionallitet som finns i HTML och VRML-grafiken är dålig. Projektmedlemmarna provade även en annan plugin för VRML, Cortona<sup>2</sup>, för att förvissa sig om att problemet med grafiken härstammade från VRML och inte från läsaren. Web3D Consortium, före detta VRML Consortium, är medvetna om problemen och jobbar med en uppföljare till VRML97, men den är ännu inte klar för marknaden.

---

<sup>1</sup>Cosmo Software, <http://www.cai.com/cosmo/home.htm>

<sup>2</sup>Parallel Graphics, <http://www.parallelgraphics.com/>

Bild som visar Cosmo Player utan att en VRML-värld har laddats in



### 33VRML-värlidens uppbyggnad

Världen består av fyra stycken sammankopplade rum. Användaren börjar i det största rummet och går därifrån vidare till de tre mindre rummen. På väggarna i dessa rum finns de länkar som av Webscoop bedömts vara mest intressanta för användaren. Varje rum innehåller nio stycken olika länkar från det speciella intresseområde som hör till rummet. Under varje länk finns fyra feedbacklänkar, bra, dåligt och ett bra alternativ för vart och ett av de andra rummen. Dessa två sista alternativen är till för att tala om för agenten att man vill ha mer information om detta, men i ett annat rum.

Kodavsnitt från VRML-mallen som visar hur en feedbacklänk ser ut:

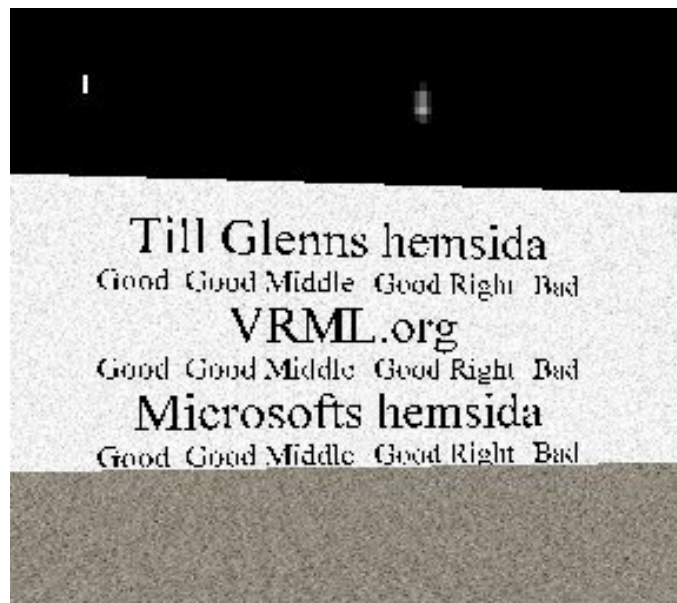
```
#feedback 11
#good feedback
  Anchor {
    url
    "http://www.mdstud.chalmers.se/~dproj4/cgi-
bin/vrmlfeedback.py?Feedback=Good"
    description "Good"
    parameter ["target=new"]
    children [
DEF gfeed11 TouchSensor {}
,Transform {
  translation -3.5 3.25 -7.47
  children [
DEF Trans Shape {
  appearance Appearance
  {
    material Material
    {transparency 1.0
  }
  geometry Box {size 1.2 0.4 0.01}
}
]
```

```

    },
    Transform {
      translation -3.5 3.25 -7.40
      children [
        Shape {
          appearance Appearance {
            material DEF mtrl11g Material {diffuseColor 0 0 0}
          }
          geometry DEF Goodtext Text {
            string "Good"
            fontStyle FontStyle {size 0.6
              justify ["MIDDLE","MIDDLE"]}
          }
        }
      ]
    }
  }
}
}
}

```

Bild som visar feedbacklänkarna



### 34Problem och lösning

Projektet provade först några olika 3D-CAD-program för att förenkla VRML-implementeringen. Men det slutade ändå med att layouten kodades direkt i VRML då det inte fanns något bra 3D-CAD-program som var gratis. När rumsimplementeringen väl kom igång gick det problemfritt tills de gamla länkarna i världen skulle bytas ut mot nya. VRML är mycket begränsat men erbjuder en möjlighet att köra JavaScript eller Javaprogram. Dessa program kan skrivas direkt i VRML-filen och köras därifrån. Därför testades först JavaScript och sedan Java, men inget av försöken gav något bra resultat. Slutligen lyckades problemet med länkarna lösas med hjälp av Python. Pythonprogrammet tar in, rad för rad, en mall i form av en VRML-värld som saknar användarens personliga länkar och letar efter specifika ord på raden. Om den hittar ett sökord tar den motsvarande rad (till exempel en länk eller en sidtitel) ur användarens databas och editerar ihop de två olika raderna. Finns inget sökord som

matchar lämnas originalraden oförändrad. Därefter skrivs raden ut på standardoutput och en ny rad läses in. Fördelen med denna lösning är att den till viss del liknar den som används för att generera Webscoops nyhetssida i HTML, med undantaget att det finns färdiga funktioner för att generera HTML-kod och som kan användas i stället för att ändra rad för rad i mallen.

Kodavsnitt ur choospageshorter.py som ersätter länkar ur mallen med användarens länkar ur databas.

```
defaultlink='http://www.mdstud.chalmers.se/~dproj-4/cgi-  
bin/choosepageshorter.py'  
    # look for the place for the link in the VRML file  
    while row[-(len(defaultlink)+1):] != defaultlink + '\":  
        print row  
        row = vrmltemplate.readline()  
  
    # replace from linkLists if link exists. Links are shown via  
    proxy, as on the HTML-page  
    if (i) < len(linkLists[n]):  
        row = row[:-(len(defaultlink+1))]  
        row = row + proxyaddress + '?url=' + linkLists[n][i] +  
'&action=view' + "\""  
        print row  
        row = vrmltemplate.readline()
```

Nästa stora problem uppstod när det skulle läggas möjlighet att ge feedback till länkarna. Efter att ha försökt med Java och rådfrågat en VRML-mailinglista utan att få några konstruktiva svar verkade problemet vara olösligt, men efter tips från projektets handledare lyckades vi komma runt det. Lösningen går ut på att feedbacklänkarna är HTML-länkar till ett CGI-skript skrivet i python. Nackdelen med denna lösning är att ett HTML-fönster öppnas. Dessutom måste argument till skriptet hårdkodas i VRML-filen, eftersom VRML till skillnad från HTML inte stöder forms för att skicka argumenten. Med tanke på svårigheterna att komma fram till en lösning är detta ändå acceptabelt. Fördelar här är liksom vid genereringen av sidan att Webscoop redan tidigare använder den här metoden för att ge feedback. I övrigt har feedbacklänkarna modifierats så att de ändrar färg om de använts, vilket inte är det normala i VRML.

### 35 Test av layouten

Därefter gjordes ett tänka högt protokoll för att utvärdera 3D-världen. (Se Appendix A). Testpersonerna gav en hel del konstruktiva förslag på saker som kunde förbättras. Mestadels handlade det om väldigt enkla saker som lätt kunde förbättras, men som kan vara svåra att se när man är alltför inne i projektet. De designmässiga slutsatserna av testet var att ingångarna till rummen var för små, världen såg tråkig ut och att ordningen på länkarna i rummen inte var avgörande hur man klickade vidare. Det som förvånade projektdeltagarna mest var det sistnämnda, att testpersonerna tittade på länkarna och utifrån detta bestämde sig för vad som verkade intressant. De som designade VRML-världen var fokuserade på att testpersonerna skulle välja det som analyseraren rangordnade, men det är naturligtvis normalt att välja en länk som man finner intressant och inte läsa dem i ordning. Detta innebar att tanken med numrerade länkar slopades och i stället sattes de bara in i rummen i den ordning som verkade lämpligast. Övriga förändringar som gjordes var att dörröppningarna breddades, en stjärnhimmel lades till som bakgrund för att öka användarens känsla av att gå runt i en

riktig värld och golv och väggar fick passande texturer för att användaren ska få mer verklighetskänsla. Ytterligare utsmyckning hade varit önskvärt, men uteslöts för att sidan inte skulle bli för långsam att ladda ner.

Bild som visar användarens utgångspunkt i VRML-världen



### 36 Test av programmens korrekthet

Om ett VRML-program är felimplementerat kommer det upp ett felmeddelande i Cosmo Players display. Detta felmeddelande berättar vilken rad som felet uppstod på och det är då enkelt att ändra.

Testningen av att få användarens personliga länkar överförda till VRML-världen, det vill säga att generera användarens personliga sida, gjordes genom att ett testprogram tog in en VRML-fil och en textfil och skrev resultatet till en VRML-fil. När detta testprogram fungerade var det en smal sak att implementera det i Webscoop. En smärre förändring gjordes i den verkliga pythonversionen så att antalet jämförelser som utfördes på varje rad kunde minskas. Detta tack vare att VRML-filens uppbyggnad är känd, och därför går det att utesluta jämförelser som inte behövs vid ett visst tillfälle.

Testningen av skriptet för att skriva feedback gjordes på liknande sätt som testet av att generera användarens personliga 3D-värld. VRML-världens feedbacklänkar till testskriptet fick värdet av feedbacken som argument. Testskriptet skrev sedan dessa argument till en fil, för att visa att feedbacken kunde ges på detta sätt.

### 37 Slutsatser

Efter att ha stött på många problem när vi försökte använda VRML till något mer än att bara vara en 3D-värld har följande slutsats dragits:

I dagsläget är VRML alltför begränsat för att vara riktigt användbart och endast genom ytterligare utveckling kan det bli konkurrenskraftigt.

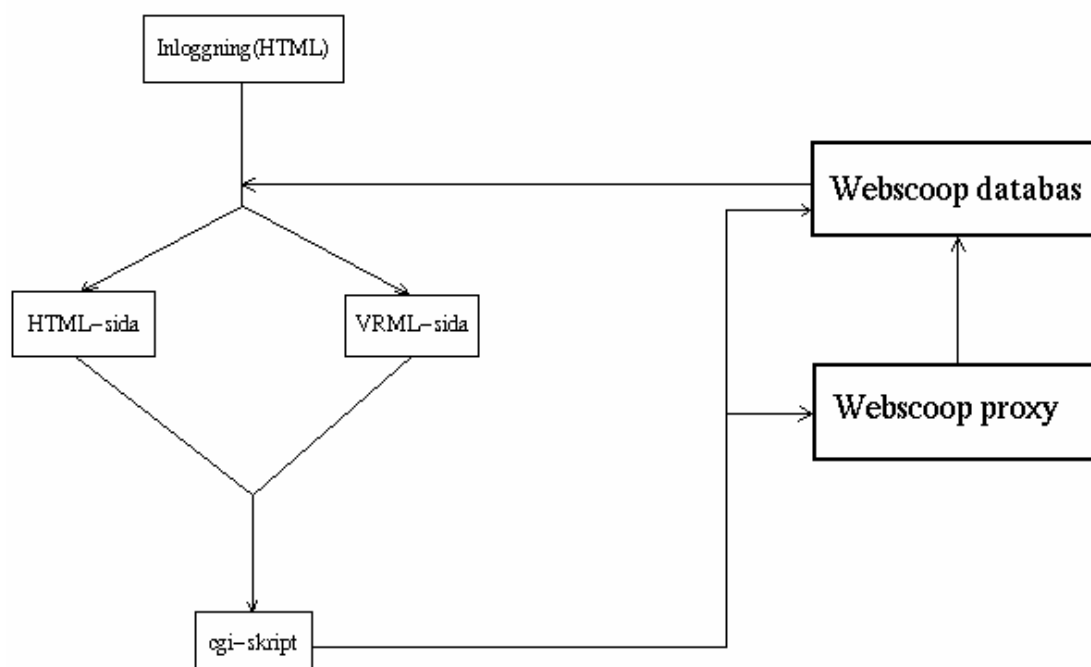
## 4 Anpassningar i Webscoop till VRML-layout

### 4.1 Introduktion och problembeskrivning

För att kunna använda den nya 3D-världen krävdes en modifikation av Webscoop. De flesta ändringar visade sig vara förhållandevis små och det enda egentliga problemet låg i att det ursprungliga Webscoop var anpassat för att varje användare bara hade ett intresseområde, medan 3D-världen har tre olika rum avsedda att innehålla information om olika ämnen.

En stor del av arbetet låg här i att studera Webscoop på djupet för att kunna utreda vilka delar som låg närmast till hands att ändra på för att kunna använda den nya layouten.

Bild som visar hur VRML-världen passats in i Webscoop



### 4.2 Lösning och implementation

Problemet med olika intressen löstes genom att för varje ny användare skapa ytterligare två, som enbart används internt av systemet och som den riktiga användaren alltså inte ser. Var och en av dessa tre användare kan ha varsitt intresseområde, och dokument kan analyseras med olika resultat för de olika rummen. På samma sätt kan feedback ges, och nya dokument registreras för de olika rummen genom att Webscoop omdirigerar dessa förfrågningar till rätt användare. På så sätt kan simuleras att varje användare har tre skilda intressen utan att behöva göra alltför stora förändringar i programmets inre struktur. Vi har alltså enbart ändrat på de cgi-skript som används vid generering av sidan och vid återkopplingar från sidan. Borttagning av de två extra användarna utförs samtidigt som den "riktiga" användaren tar bort sig.

En funktion som lagts till i samband med detta är att det finns möjlighet för användaren att namnge rummen. Detta används inte till någon analys, utan används enbart som en titel på rummet. Av denna anledning behöver ett extra fält i tabellen USERS läggas till, för att spara rumsnamnen. Inställningar samt information om användaren kommer enbart att sparas för huvudanvändaren, eftersom dessa data är överflödiga för de två extra användarna.

Det enda användaren ser av detta är de ytterligare funktioner förändringen ger tillgång till. De extra användarna har fått standardiserade namn, uppbyggda av en siffra och det riktiga användarnamnet. Av den här anledningen togs möjligheten bort att ha ett användarnamn som inleds med en siffra.

Kod ur WebScoop.py där en kontroll att användarnamnet inte inleds med en siffra

```
# Check if the first letter is a digit, if so displayError
    if username[0] in string.digits:
        displayError("Invalid user name", "Please choose a user name
which does not begin with a digit.", "/webscoop/newuser.html",
"Create New User")
```

Kodavsnitt ur newuser.py där "mittenrummsanvändaren" skapas

```
# Middle room "user"
# Insert into USERS
queryStr = "insert into USERS values ('" + userid + "','" + username
+ "','"
queryStr = queryStr + password1 + "','" + roomname + "'"
db.query(queryStr, "User name already taken!",
"","/webscoop/newuser.html","create new user")
```

När användaren har möjlighet att ha tre olika intressen, är det viktigt att HTML-sidan har en liknande uppdelning, så att programmets funktionalitet inte skiljer sig alltför mycket åt mellan de olika layouterna. Detta har lösts på enklast möjliga sätt, genom att helt enkelt dela av sidan i tredjedelar med tillhörande ämnesrubriker.

Väsentliga skillnader i visningen av de av Webscoop framvaskade sidorna är att den dåliga upplösningen i VRML-grafiken gör det svårt att visa en liten bit i början av texten, som görs i HTML-varianten. I övriga delar, som inte gäller visningen av länkarna till dessa sidor, går i princip inget att göra i VRML-layouten. Inställningar och registrering av sidor till proxyn måste göras i HTML, eftersom begränsningar i språket gör det omöjligt i VRML. Även om det vore möjligt bedömdes det som svårare för användaren att sköta den här typen av inmatningar i VRML-världen. Länkar till dessa HTML-sidor finns dock i VRML-världen, eftersom funktionerna är nödvändiga för Webscoop.

Eftersom de olika rummen behandlas som skilda användare då dokumentet analyseras, tas ingen hänsyn till resultatet av analysen i övriga rum. Om användarens intressen skiljer sig för lite från varandra kan det då hända att Webscoop vill visa samma sida i flera rum. Dessa överflödiga sidor sorteras bort då länkarna ska visas, det vill säga när Webscoop-sidan/världen genereras. Sidan visas i det rum där den fått högst poäng. Detta kan leda till att ett rum snabbare får slut på sidor att visa. Inträffar detta, meddelas användaren - oavsett orsaken till att det saknas länkar - att denne nu

måste registrera nya sidor till proxyservern.

### **43Fördelar och nackdelar med denna lösning**

Den stora nackdelen med denna lösning är att belastningen på analysdelen i programmet tredubblas. Eftersom analyseringen av dokument är den stora flaskhalsen i det existerande programmet är detta mindre bra. Å andra sidan är det svårt att tänka sig någon lösning där varje användare har olika intressen som inte ökar arbetsbördan vid analyseringen.

Genereringen av användarens sida blir också långsammare eftersom alla länkar måste jämföras med varandra innan de kan visas.

Fördelen med lösningen är att den är lätt att passa in i det existerande programmets uppbyggnad. Genom att i stort sett behålla den inre strukturen blir risken mindre för följdfel av förändringarna. Detta var en nödvändighet eftersom möjligheterna till testkörning var mycket små.

### **44Tester och kontroller av skripten för sammankoppling**

Vid tester av skripten uppstod en del problem. Webscoop hade fortfarande inte fåttts att köra, och följaktligen kunde inget av skripten testas i sin helhet, eftersom de alla behöver både läsa och skriva information i Webscoops databas. Ytterligare problem uppstod eftersom vi inte lyckats installera tilläggsmoduler i Python, som använts förra året. Dessa tilläggsmoduler behövs vid databasåtkomst och snabbgenerering av HTML-sidor. Av dessa anledningar har bara mycket små delar av skripten kunnat testas. Dessa har dock konstaterats uppföra sig korrekt. I övrigt har koden studerats och ibland jämförts med skript från tidigare år för att så långt som möjligt rätta de fel som förekommit.

## 5XML

### 51Inledning

XML, eXtended Markup Language, har vissa likheter med HTML, men är mer välstrukturerat, och denna egenskap hos språket kan utnyttjas av vårt projekt i analysdelen av Webscoop. Den stora fördelen med att använda XML är därför att analysen av sidor förbättras så att man med större sannolikhet får rätt information.

### 52Vad är XML?

XML är ett förslag till metaspråk som erbjuder en möjlighet att införa struktur i informationen på Internet, samt att man kan skapa generiska dokument som kan bearbetas av olika applikationer.

Ett exempel på ett XML-dokument som hanterar information om cd skivor i en skivhandel:

```
<?xml version="1.0"?>
  <CATALOG>
    .
    .
    .
    <CD>
      <TITLE>Empire Burlesque</TITLE>
      <ARTIST>Bob Dylan</ARTIST>
      <COUNTRY>USA</COUNTRY>
      <COMPANY>Columbia</COMPANY>
      <PRICE>10.90</PRICE>
      <YEAR>1985</YEAR>
    </CD>
    .
    .
  </CATALOG>
```

För att publicera ett XML-dokument används XSL, som är en mall för hur ett XML-dokument ska formateras och publiceras.

Standarden för XSL består av två huvuddelar, en som används för att konvertera ett XML-dokument till ett format som kan visas, till exempel HTML, och en del som handlar om hur sidan ska se ut till exempel vad gäller typsnitt och teckenstorlek.

Exempel på ett XSL-dokument för motsvarande XML-dokument:

```
<?xml version="1.0"?>
  <xsl:stylesheet xmlns:xsl="http://www.w3.org/...">
  <xsl:template match="/">
    <html>
    <body>
      <table border="2" bgcolor="yellow">
        <tr>
          <th>Title</th>
          <th>Artist</th>
```

```

</tr>
<xsl:for-each select="CATALOG/CD">
<tr>
  <td><xsl:value-of select="TITLE"/></td>
  <td><xsl:value-of select="ARTIST"/></td>
</tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

## 53DTD

Document Type Definition anger de element som är tillåtna i ett XML-dokument, definierar taggar och deras betydelse. DTD-dokumentet kan antingen vara ett separat dokument eller inbyggt i XML-dokumentet. Strukturen för ett XML-dokument bygger på överensstämmelse med DTD-dokumentet.

Example på DTD :

```

<!DOCTYPE NEWSPAPER [
  <!ELEMENT NEWSPAPER (ARTICLE+)>
  <!ELEMENT ARTICLE (HEADLINE, BYLINE, LEAD, BODY,
NOTES)>
  <!ELEMENT HEADLINE (#PCDATA)>
  <!ELEMENT BYLINE (#PCDATA)>
  <!ELEMENT LEAD (#PCDATA)>
  <!ELEMENT BODY (#PCDATA)>
  <!ELEMENT NOTES (#PCDATA)>

  <!ATTLIST ARTICLE AUTHOR CDATA #REQUIRED>
  <!ATTLIST ARTICLE EDITOR CDATA #IMPLIED>
  <!ATTLIST ARTICLE DATE CDATA #IMPLIED>
  <!ATTLIST ARTICLE EDITION CDATA #IMPLIED>

  <!ENTITY NEWSPAPER "Vervet Logic Times">
  <!ENTITY PUBLISHER "Vervet Logic Press">
  <!ENTITY COPYRIGHT "Copyright 1998 Vervet Logic Press">
]>

```

## 54XML-parser

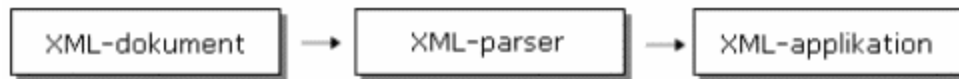
XML-parser är en programvara som bearbetar XML-syntaxen för att en applikation lättare ska kunna använda sig av XML-dokumentet. Parsern bygger upp ett abstrakt syntaxträd för att underlätta sökningen efter givna sökord.

Det finns två typer av XMLparser:

- ⑩ Icke validerande

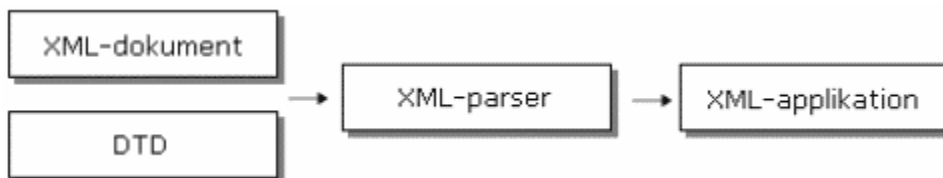
## ⑩ Validerande

En icke validerande parser kontrollerar enbart om XML-dokumentet är välutformat, vilket innebär att det kontrollerar att dokumentets syntax är korrekt. Detta sträcker sig inte längre än en kontroll av att varje starttag har motsvarande sluttagg, till exempel: `<person><name>Per Nilsson</name></person>`



Figur 1

En validerande parser analyserar kontrollerar också att dokumentet är välutformat. Utöver detta analyserar om dokumentets grammatik överensstämmer med den som givits i motsvarande DTD-dokument.



Figur 2

## 55NewsML

NewsML är en delmängd av XML som är speciellt definierat för nyhetsmedia. Det är de generiska egenskaperna som gör NewsML effektivt, att publicera en gång, till en låg kostnad och oberoende av plattform och applikation. Olika nyhetspublicerare kan med hjälp av NewsML publicera sina nyheter på flera olika sätt, i form av bilder, video, text eller audio filer. NewsML ska göra det lättare när allt fler nyhetstjänster publicerar sitt material på nätet.

IPTC ( International Press Telecommunications Council ) är en organisation som arbetar med en standardiseringen av NewsML och försöker få nyhetsmedier att använda sig av NewsML. IPTC anser att NewsML i dagsläget är tillräckligt utvecklat för att kunna användas som avsett.

## 56Applikation

Egentligen skulle analysdelen i Webscoop ha utökats för att även bearbeta XML- och NewsML-dokument, men eftersom Webscoop inte har kunnat köras implementerades en fristående applikation. Applikationen använder ett utvecklingsbibliotek (Libwww) för webbtillämpningar.

Applikationen letar igenom en HTML-sida efter länkar till XML-dokument. I XML-dokumentet söks sedan efter vissa taggar för att avgöra om dokumentet innehåller nyheter eller ej. De taggar som söks efter har bestämts med avseende på NewsML:s DTD version 1.0, som har arbetats fram av IPTC:

1. DateAndTime - Specificerar datumet för nyheten
2. HeadLine - Specificerar rubriken för nyheten
3. DateContent- Innehåller nyhetsinformation
4. NewsItem - Refererar till http URL eller ett NewsML URN

Applikationen söker först efter en "DateAndTime"-tagg för att undvika gamla nyheter. Om datumet visar att nyheten är aktuell, letar applikationen vidare efter en "HeadLine"-tagg i dokumentet. Och om den taggen i sin tur specificerar en rubrik som kan vara intressant för en användare så letar applikationen vidare efter en "NewsItem"-tagg som innehåller en länk till ett annat XML-dokument som innehåller nyhetstexten.

I det refererade XML- dokumentet finns i sin tur en DataContent-tag. Här finns nyhetsinformationen man sökte efter.

Eftersom ett obearbetat XML-dokument inte är till för publicering, måste man först bearbeta den för att publicera den i ett önskat format. Man kan erhålla informationen från NewsML-dokumentet och visa den direkt för användaren eller hänvisa dem till URL:en som innehåller länkar till NewsML-dokument.

XML-komponenten ska implementeras färdigt och kopplas till andra komponenter.

## **57 Testning**

Av olika anledningar som kommer att tas upp senare, misslyckades projektteamet med att få igång Webscoop. Som en följd av detta har komponenter som implementerats inte kunnat testas tillsammans med Webscoop. Däremot har applikationerna testats enskilt för att kontrollera att de fungerar som avsett.

## **58 Framtida arbete**

För att applikationen ska anpassas till Webscoop krävs en del ändringar. Utvecklingsbiblioteket måste raderas och applikationen kan använda crawlern för webbtillämpningar. Själva applikationen kan kombineras med analysdelen.

### **Implementation**

NewsML-komponenten bygger på att crawlern, som hämtar dokument från www, ska ändras så att den även klarar av att hämta XML-dokument och vidarebefordra dem till analysverktyget. Vid analys ska det i sin tur finnas metoder för att bedöma dokumenttypen.

Man skulle också kunna tänka sig att välja att utnyttja minnesdelad meddelandeskickning (så som sker mellan crawlern och analyser-komponenten). Det innebär att NewsML-komponenten ska implementeras så att även den ligger och väntar på XML-dokument i den minnesdelade arean.

Newsml(1.1)

Den NewsML-komponent som implementerats i projektet kommer att erhålla

dokument med hjälp av logger, som har sparat giltiga dokument i minnet. NewsML-komponentens uppgift är att parse dokument och spara resultatet i databasen för att sedan kunna visas för användaren.

## 59Ändringar i Webscoop

### Crawlern

Funktionen "validURI" ska ändras så att den även accepterar filer med suffix .XML och .xml, så att även XML-länkar returneras av crawlern.

### Analyzer

Utöka analyzern med en metod som kan bearbeta dokumentet vidare enligt förklaring ovan.

### DBInterface

Lagra respektive hämta NewsML-dokument från databasen. Databasen kan lämnas oförändrad, och resulterande information från applikationen ska lagras respektive hämtas från databasen. De fält där länken ska sparas används i så fall för att spara länken till ett färdiggenererat dokument som går att titta på, och fält för text och titel får sitt innehåll från XML-filen.

## **60 Områden som studerats utan att ha införlivats i projektet**

### **61 Nearest Neighbour-klassificering**

Nearest Neighbour-klassificering var ett av de områden som studerades i hopp om att förbättra träffsäkerheten vid analysering av nya dokument. Då den här metoden är tillämpbar ger den bra resultat. Idén förkastades dock, eftersom metoden kräver stora mängder indata för att fungera tillfredsställande. Användaren skulle i så fall tvingas registrera många sidor innan metoden blir användbar, vilket går emot idén bakom Webscoop, att användaren inte i princip bara ska behöva registrera en sida för att komma igång.

### **62 Hur datorsystem kan användas för att upptäcka kunskap.**

Idén är att datorsystemet skall upptäcka regelbundenheter i data och på så sätt få ny kunskap. Datorsystemet kan kanske upptäcka vissa gemensamma drag hos de som inte betalar tillbaka sina lån i en bank.

Man skulle kunna förvänta sig att datorsystemet upptäcker att de som har låg inkomst har svårare att betala tillbaka sina lån. Utvecklingen har ännu inte kommit så långt att det finns system som klarar av stora mängder av riktig data såvida den inte har förenklats av människor innan, men detta är antagligen något som kommer i framtiden. Ett ytterligare steg som man kan se som en utveckling av idén är att datorsystemet förutom att upptäcka regelbundenheter skall kunna föreslå lämpliga åtgärder baserat på vad man vill uppnå. För att återvända till exemplet ovan så skall datorsystemet när det upptäcker vissa gemensamma drag hos de som inte betalar tillbaka sina lån i en bank kunna ge förslag på hur man skall komma till rätta med problemet.

Tillämpningen för vårt projekt skulle ha varit att vårt system skulle ha kunnat leta efter gemensamma drag hos en användares surfvanor och till exempel sätta högre betyg på en webbplats om användaren surfar dit mycket. Vårt system skulle även kunnat jämföra olika användares surfvanor med varandra och dra slutsatser som till exempel att om en användare surfar till en viss webbplats så tycker hon antagligen att vissa andra sidor också är bra, och systemet bör i så fall visa även dessa.

### **63 Personalisering av hemsidor**

Ett förslag till förändring av Webscoop var att ge användaren en möjlighet att skapa en mer personlig lay-out. Tanken med detta var att användaren skulle kunna placera de länkar som Webscoop lägger upp i en ordning som passar personen. Exempel på andra features är ändra färg på bakgrund och lägga till eller ta bort länkar. Webscoop skulle därför fungera mer som en portal och detta kändes inte rätt.

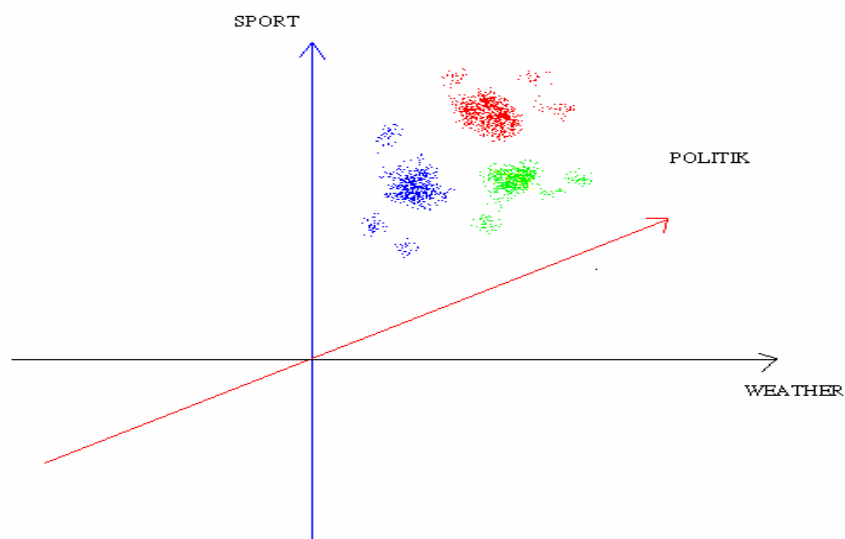
Projektmedlemmarna bestämde dessutom att man skulle satsa på 3D och dessa två skäl gjorde att denna del lades ned.

### **64 WAP**

Wireless Application Protocol (WAP) är en universell standard som kopplar ihop internet med mobiltelefonen. Användaren skulle därmed mobilt kunna inhämta information som Webscoop har funnit. Projektet släppte denna del på grund av tidsbrist.

## 65Klustring

Klustring används för att effektivisera informationssökning. Man kan använda sig av klustring för att erhålla den information man söker efter. Man kan se Internet som ett stort vektor rum där dokumentet kan repressenteras som en vektor i vektorrummet.



Figur 3

Alla dokument som användaren har besökt, registreras som en vektor i vektorrummet. Efter ett tag kan man urskilja olika kluster av dokument (Figur 1). Dessa kluster kan man se som olika ämnen som kan användas för informationssökning. Varje axel motsvarar en ämneskategori som måste vara definierad från början. De fördefinierade ämneskategorierna söks efter i dokumenten och får då ett bestämt läge i vektorrummet. Läget i rummet beräknas fram av ämneskategoriernas förekomst i dokumentet.

Inom Information Retrieval (IR) området finns det många tillämpningarna som använder sig av metoder som bygger på klustring. Undersökningar på olika klustringsmetoder gjordes för att se om dessa kunde användas för projektet. Men det visade sig att implementeringen av någon av de komplexa metoderna som bygger på klustring, skulle kräva mer tid än vad som fanns att tillgå.

## **7Fel som gjordes under projektets gång**

Resultatet av vårt projekt har blivit något under förväntan. Vi har här utrett vilka problem vi haft som lett till detta.

### **71Fel inom gruppen**

Gruppen hade svårt att ta beslut om vad som skulle göras i projektet. Detta berodde på att det var en viss oenighet inom gruppen och medförde att vi saknade en klar linje att arbeta efter.

Gruppmedlemmarna hade svårt att kommunicera med varandra, detta innebar att det blev svårt för någon medlem att få en överblick av läget.

Initialproblem med projektet medförde att gruppmedlemmarna tappade lite av sin motivation och därför nedprioriterade kursen.

En stor miss gjordes även i planeringen av projektet. Projektmedlemmarna underskattade grovt tidsåtgången för att sätta sig in i Webscoop. I princip ingen tid var avsatt till detta, och därför spräcktes den ursprungliga planeringen.

### **72Fel utanför vår kontroll**

Den arkiverade version vi fick av Webscoop var inte fullständig. Viktiga filer saknades och programmet var på vissa ställen knapphändigt kommenterat. Dessutom hade omgivningen i datorsystemet på Matematiskt Centrum (MC) förändrats efter föregående års projektavslutning, vilket medförde att programmet blev omöjligt att kompilera. Under detta år har vi fått intrycket att Helpdesk varit underbemannat och därför inte haft tid att hjälpa oss med de problem som rör systemet på MC. Dessa problem har sammantaget lett till att vi inte fått igång det gamla programmet. Detta innebar att projektet fick en mycket teoretisk karaktär och vi fick därför ingen känsla för programmet. Detta medförde i sin tur att vi hade svårt att avgöra vilka förbättringar som på ett naturligt sätt kunde passa in i Webscoop.

Andra problem som vi har ställts inför är att VRML saknar viktig funktionalitet som hade behövts för vårt projekt och vi tvingades därför hitta på fusklösningar på enkla problem. Vi har även lagt ned mycket tid på laptopen utan att detta har gett oss någonting. Vi trodde att allting skulle bli bra när laptopen skulle fungera, men det visade sig vara nästan oöverkomligt eftersom programvar kostar pengar (Fråga Yavuz om helpdesk har velat ge oss några licenser och vad som hände egentligen).

## 8Referenser

VRML

Cosmo Software, <http://www.cai.com/cosmo/home.htm>

Parallel Graphics, <http://www.parallelgraphics.com/>

Web 3D Consortium, <http://www.web3d.org>

Andrea Ames, David R. Nadeau, John L. Moreland - The VRML 2.0 Sourcebook -  
2nd Ed (September 1996) John Wiley and Sons

Mark Lutz, David Ascher - Learning Python - (1 Mars, 1999) O'Rilley UK

<http://www.oasis-open.org/cover/newsML19991130.html>

<http://www.w3schools.com/xml>

World Wide Web Consortium, <http://www.w3.org>

International Press Telecommunications Council, <http://www.iptc.org>

Internet Engineering Task Force Working Groups, <http://www.imc.org/ietfwwg.html>

<http://newsshowcase.reuters.com>

## 9Appendix A

### 91Tänka högt protokoll, VRML-layout

Ett tänka högt protokoll genomförs enligt följande; Testpersonen får ett antal uppgifter som hon ska lösa, men medan de utförs ska hon berätta högt varför hon gör det på det sättet och berätta om det uppkommer några tveksamheter. Ett tänka högt protokoll är därför ett bra redskap när man provar något nytt.

Målet med projektets tänka högt protokoll är att få svar på följande frågor:

Är texten tydlig?

Är det enkelt att gå in i rummen?

Vilken ordning ska länkarna på väggarna vara ordnade?

Testpersonerna fick även besvara fem stycken frågor efter att de hade utfört tänka högt protokollet.

### 911Testpersoner

Testpersonerna i undersökningen bestod av 12 stycken högskolestuderande i åldrarna 19-24 år. Lite drygt hälften, 7 stycken, av testpersonerna ansåg sig tillhöra kategorin "erfaren" när det gäller datorvana, medan de övriga endast använde datorn till att surfa, skicka mail samt som ordbehandlare. Kön fördelningen var 50/50.

Innan testpersonerna genomförde tänka högt protokollet fick de en kort genomgång av projektet samt hur Cosmo Player fungerar.

### 912Uppgifter

**Testpersonen ska gå in i det vänstra rummet och klicka på den tredje mest intressanta länk som hon tror att analyseraren har valt ut.**

Två av testpersonerna klagade direkt på att ingången till rummet var alldeles för smal och även några andra hade svårigheter att komma in, men detta skyllde de dock inte på dörren utan på "bristande datorvana".

Endast 5 testpersoner tryckte på "rätt" länk. Det fanns två huvudalternativ; Längst ned på väggen rakt fram eller längst ned på den vänstra väggen (denna var den "rätta"). En av testpersonerna tryckte på den länk som personen tyckte verkade vara den tredje mest intressanta ur personens synvinkel. En annan testdeltagare påpekade att ordningen spelar inte någon roll, eftersom man klickar på de länkar som man tror verkar vara mest intressanta", dvs man följer inte analyserarens råd.

Många av testpersonerna klagade över svårigheterna att se den högra väggen samt att gå ut ur rummet.

**Gå in i mittenrummet och klicka på länk 7.**

Liknande resultat som ovan, ej självklart vilken länk sju är.

**Klicka på huvudlänken för det högra rummet och ge den ”bad” som feedback.**  
Ingen av testpersonerna klickade på länken i det stora rummet och de hade även svårt att bestämma vilken som var huvudlänken.

Vid feedbacken fick vi följande kommentarer; "Bra att det blev en förändring", "Varför kommer det upp ett nytt fönster och varför måste man själv stänga det?", "Rött framhäver, varför inte tona ned en använd feedbacklänk istället?", "Varför blev båda arna röda? Jag har bara tryckt på en.", "Varför stängs inte feedbacken av när man har tryckt en gång?"

### **913Slutsatser av tänka högt protokollet**

Öppningarna in till rummen måste breddas eftersom många av testpersonerna hade problem med att ta sig in och ut. Ordning på hur länkarna sitter spelar ingen roll eftersom testpersonerna väljer de som verkar intressanta, och inte dem som systemet ansåg vara bäst.

### **92Frågor som ställdes efter att testpersonerna har gjort ett tänka högt protokoll**

#### **Vilket var ditt första intryck av Webscoop?**

Här kan man dela upp testpersonerna i två grupper, ”erfarna” och ”vanliga användare”. De erfarna klagade på att det var svårt att se texterna, långsamt samt dåligt genererat. Vanliga användare tyckte det var kul med 3-D, häftigt samt kul att styra sig fram.

#### **Hur kan designen förbättras?**

Kommentarer från testpersonerna var att det var sterilt, ej hightech, förskräckliga färger, svårt att se texten, det behövs vackrare färg på golvet och att det var en tråkig himmel.

#### **Vad var bra med Webscoop?**

Nytt sätt att interagera med internet, bra att analyseraren hittar de länkar som man är intresserad av.

#### **Vad var dåligt med Webscoop?**

Kommentarerna var följande; "Varför är det blandat svenska och engelska?", "det var svårt att styra Cosmo Player". Många kommentarer handlade om designförbättringar och har därför redan noterats i "Hur kan designen förbättras?".

#### **Kan du tänka dig att använda systemet?**

Här kan man återigen dela upp testpersonerna i ”erfarna” och ”vanliga användare”. De erfarna tyckte att det gick för långsamt och ville hellre ha en HTML-sida, medan vardagsanvändarna var överlag mer positiva. Några kommentarer som fälldes av testpersonerna; Bra att ha allting samlat i ett rum. Endast för specifika personer, exempelvis om man är expert i någonting och vill hålla koll vad som händer så är Webscoop ett bra alternativ.

### **Slutsatser av frågor**

De personer som var negativa mot Mycket av den kritik som kom gäller VRML och inte vår idé. Det gäller att det kommer ett par versioner till innan VRML slår.