

Framtagning av lämplig teknologi för konvertering av VBA-applikation till Internet Explorer

Investigation and recommendation of a suitable
technology for converting a VBA-application to
Internet Explorer

Andreas Mohlin

2003

EXAMENSARBETE
Informatik
Nr:



DALARNA
University College

EXAMENSARBETE, C- nivå i Informatik

Program Systemvetenskapligt program, 120p	Reg nr	Omfattning 10p
Namn Andreas Mohlin	Månad/År augusti 2003	Handledare: Joakim Karlsson Examinator: Göran Hultgren
	Företag/Institution GUTAB	Handledare vid företaget Gunnar Tholander
Titel Framtagning av lämplig teknologi för konvertering av VBA-applikation till Internet Explorer		
Nyckelord VBA, ASP, ASP.NET, COM, .NET-objekt, konvertering, Internet Explorer		

Sammanfattning

Företaget GUTAB använder idag en frågeanalysmodul som är utvecklat i Visual Basic for Applications i Microsoft Excel. GUTAB fann det intressant att veta om modulen kunde konverteras till webbläsaren Internet Explorer samt vilken teknologi som var lämplig för detta. GUTAB var även intresserade av att veta vilken teknologi som kunde användas för att skapa grafiska manipulerbara objekt.

En skrivbordsundersökning genomfördes som omfattade teknologierna ASP, ASP.NET, JSP, PHP, ActiveX samt Java applets. Skrivbordsundersökningen utgjordes av två delar där den första delen redogjorde för de två klientbaserade teknologierna ActiveX och Java applets med vilka grafiska manipulerbara objekt kunde realiserats. GUTAB ansåg att ActiveX program var mest lämpliga för detta ändamål.

Baserat på skrivbordsundersökningens andra del bestämde GUTAB att ASP och ASP.NET skulle användas för att testa teknologiernas kapacitet att konvertera programlogik samt grafiska objekt från VBA-applikationen i en webbapplikation då dessa två teknologier ansågs som mest lämpliga för en konvertering.

Testet som genomfördes visade att det var fullt möjligt att konvertera utvald funktionalitet med båda teknologierna. Dock kunde inte grafiska objekt skapas i asp varför befintliga objekt användes för det ändamålet. Många fördelar identifierades med ASP.NET varför GUTAB beslutade att ASP.net var den mest lämpliga teknologin att använda vid en eventuell konvertering.



DALARNA
University College

DEGREE PROJECT in Informatics

Course System and business development	Reg number	Extent 15 ects
Names Andreas Mohlin	Month/Year August 2003	
	Supervisor Joakim Karlsson Examiner: Göran Hultgren	
Company/Department GUTAB	Supervisor at the Company/Department Gunnar Tholander	
Title Investigation and recommendation of a suitable technology for converting a VBA-application to Internet Explorer		
Keywords VBA, ASP, ASP.NET, COM, .NET-objects, conversion, Internet Explorer		

Summary

The company GUTAB is today using a questionanalysismodule developed in Visual Basic for Applications in Microsoft Excel. GUTAB wanted to investigate if the module could be converted to the webbrowser Internet Explorer and which technology that was suitable for that. GUTAB was also interested in which technology that could be used to create graphical manipulative objects.

A desktopinvestigation was carried out that covered the technologies ASP, ASP.NET, JSP, PHP, ActiveX and Java applets. The desktopinvestigation was split into two parts where the first part presented the two client-based technologies ActiveX and Java applets that could be used to create graphical manipulative objects. GUTAB decided that ActiveX-programs was most suitable.

Based on the second part of the desktopinvestigation it was decided that a test of ASP and ASP.NET would be made. The purpose of the test was to verify the possibilities of the technologies to convert sourcecode and graphical objects in a webbapplication. The test that was done showed that it was fully possible to convert the specific functions with both technologies. With asp it was not possible to create graphical objects, therefore existing objects were used to solve that problem.

Many benefits were identified with ASP.NET why GUTAB decided that ASP.NET was most suitable to use in a potential conversion.

Innehållsförteckning

1	INLEDNING	1
1.1	GUTAB	1
1.2	BAKGRUND	1
1.3	SYFTE & MÅL	2
1.4	METODÖVERSIKT	2
1.5	AVGRÄNSNING	3
2	METOD	5
2.1	FRAMTAGNING AV KRITERIER	5
2.2	SKRIVBORDSUNDERSÖKNING	5
2.3	VAL AV KONVERTERINGSTEKNOLOGIER FÖR TEST	6
2.4	LITTERATURSTUDIE	6
2.5	TEST AV KONVERTERINGSTEKNOLOGIER.....	6
2.6	TESTUTVÄRDERING	7
2.7	VAL AV KONVERTERINGSTEKNOLOGI.....	8
3	GUTABS FRÅGEANALYSMODUL.....	9
3.1	MICROSOFT VISUAL BASIC FOR APPLICATIONS	9
3.2	PROGRAMBESKRIVNING	9
3.3	FUNKTIONSBEKRIVNING	10
4	FRAMTAGNING AV LÄMPLIG KONVERTERINGSTEKNOLOGI.....	12
4.1	KRITERIER FÖR VAL AV KONVERTERINGSTEKNOLOGI	12
4.2	SKRIVBORDSUNDERSÖKNING	13
4.2.1	<i>Skrivbordsundersökning, del 1</i>	<i>13</i>
4.2.2	<i>Skrivbordsundersökning, del 2</i>	<i>16</i>
4.3	VAL AV KONVERTERINGSTEKNOLOGIER FÖR TEST	18
4.4	TEORETISK REFERENS RAM	19
4.5	TEST AV KONVERTERINGSTEKNOLOGIER.....	24
4.5.1	<i>Val av funktionalitet.....</i>	<i>24</i>
4.5.2	<i>Användarfall</i>	<i>24</i>
4.5.3	<i>Treskiktsmodell tillämpad på webbapplikationerna</i>	<i>25</i>
4.5.4	<i>Testmiljö</i>	<i>25</i>
4.5.5	<i>APIs</i>	<i>26</i>
4.5.6	<i>Utvecklingsverktyg för testning</i>	<i>26</i>
4.6	TESTUTVÄRDERING	29
4.7	VAL AV KONVERTERINGSTEKNOLOGI.....	33
5	SLUTSATS	35
6	DISKUSSION.....	36
7	REFLEKTION.....	37
8	ORDLISTA.....	39
9	KÄLLFÖRTECKNING	40

1 Inledning

Den här rapporten utgör dokumentationen i kursen ”Examensarbete i informatik”, 10 poäng på C nivå. I detta inledande kapitel beskrivs uppdragets bakgrund, mål, syfte och avgränsning. Kapitlet omfattar även en beskrivning av uppdragsgivaren samt metoder som används vid insamling av information.

1.1 GUTAB

GUTAB bildades 1989 och är ett företag vars verksamhet huvudsakligen utgörs av metoder och verktyg för projektstyrning och affärsutveckling. Företaget arbetar med metodutveckling, försäljning och implementering av programvaror.

Tillsammans med kvalificerade partners erbjuder GUTAB cirka 20 experter inom sitt affärsområde. Dessa utgörs av tio metodkonsulter, sju systemutvecklare och tre tekniker inom programvarusupport som tillsammans besitter kompetens inom projektstyrning, programvara för projektstyrning, systemintegration, riskbedömning och affärsutveckling.

Företaget ger expertstöd inom sina verksamhetsområden genom att leverera tjänster för tillämpning av effektiva metoder som stöds av programvarorna. Företagets kunder utgörs till stor del av stora internationella koncerner, statliga verk och medelstora företag.

Tillsammans med sina partners är företaget verksamma i Sverige, Norge och Danmark och har över 20 års erfarenhet av internationell produktdistribution.

1.2 Bakgrund

GUTAB har utvecklat en frågeanalysmodul¹ i programspråket Visual Basic for Applications som i fortsättningen kommer att benämnas VBA². Frågeanalysmodulen är utvecklad i programmet Microsoft Excel.

Företaget överväger att konvertera frågeanalysmodulen så att den blir exekverbar i Webbläsaren Internet Explorer. De vill därför undersöka vilka möjligheter det finns att konvertera VBA-applikationen samt veta vilken teknologi som är mest lämplig för detta.

Den webbaserade frågeanalysmodulen ska användas i en nätverksbaserad miljö där klienten som använder modulen inte är beroende av någon klientprogramvara som exempelvis Microsoft Excel. Frågeanalysmodulen ska finnas tillgänglig på en server där information kommer att sparas. Vad gäller

¹ GUTABs frågeanalysmodul förklaras i kapitel 3 ”GUTABs frågeanalysmodul”

² VBA förklaras i kapitel 3 ”GUTABs frågeanalysmodul”

teknologins framtidsstatus så bör den i dagsläget betraktas som modern. Den blivande webbaserade frågeanalysmodulen ska kunna spara användarens resultat i exceldokument.

1.3 Syfte & Mål

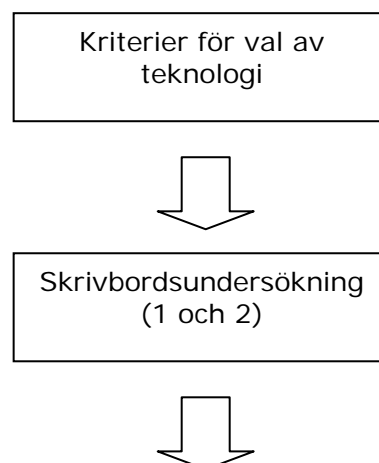
Målet med det här examensarbetet är att utföra en undersökning som besvarar följande tre frågeställningar:

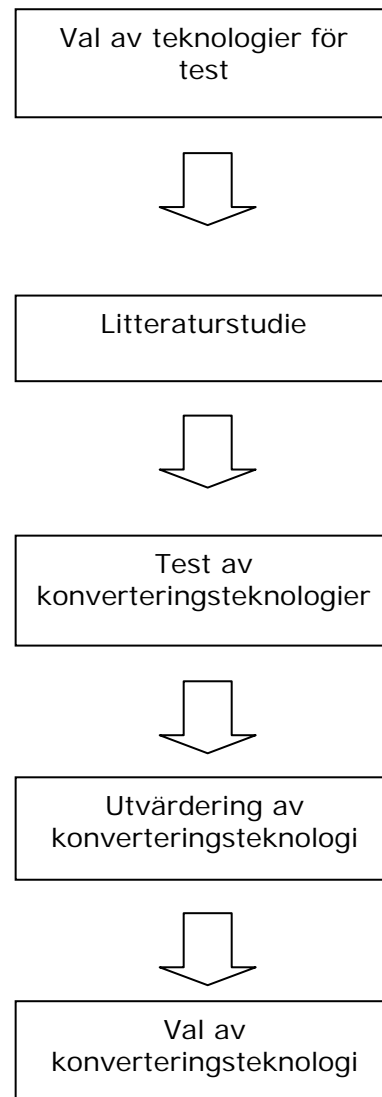
- Vilka två teknologier kan, utifrån GUTABs befintliga frågeanalysmodul, betraktas som lämpliga att använda för att konvertera företagets VBA-applikation till Internet Explorer?
- Vilken av de två lämpliga teknologierna är mest lämplig att använda för att konvertera GUTABs frågeanalysmodul till webbläsaren Internet Explorer?
- Är det möjligt att konvertera frågeanalysmodulen till Internet Explorer utan att befintlig funktionalitet går förlorad?

Arbetet syftar till att genomföra ett test av två potentiella konverteringsteknologier som på en teknisk nivå redogör för vilka problem respektive möjligheter som kan uppstå då en konvertering genomförs med de två valda teknologierna. Arbetet genererar således ett förslag på hur en webbaserad version av GUTABs frågeanalysmodul kan se ut.

1.4 Metodöversikt

För detta arbete har en egen metod skapats. Metoden heter X-method och illustreras i figur 1.1.





Figur 1.1 – Metoden X-method's samtliga metodsteg.

1.5 Avgränsning

En avgränsning i detta examensarbete har gjorts vid valet av konverteringsteknologier som ska ingå i skrivbordsundersöknings första del. Skrivbordsundersökning del 1 behandlar teknologierna Active Server Pages, Active Server Pages.NET, Pearl Hypertext Preprocessor och Java Server Pages. Dessa fyra teknologier har valts då de i dagsläget används i störst utsträckning vid utveckling av interaktiva och dynamiska webbsidor¹.

¹ Active Server Pages kommer i fortsättningen att benämnas ASP, Active Server Pages.NET - ASP.NET, Pearl Hypertext Preprocessor - PHP och Java Server Pages - JSP

I skrivbordsundersökning del 2 behandlas två teknologier vilka inte kommer att testas genom implementering eller konvertering. Dessa teknologiers för- respektive nackdelar kommer endast att identifieras genom en teoretisk referensram.

Val av teknologier som kommer att testas avgränsas till två till antalet vilket ger utrymme för en hanterbar utvärdering.

Konverteringsarbetet kommer endast att baseras på utvald funktionalitet, vilken anses som mest kritisk att konvertera.

Webbapplikationerna konverteras endast med syftet att genomföra en utvärdering av dessa i detta arbete. Konverteringsarbetet genererar därmed inte webbapplikationer innehållande funktionalitet som svarar mot samtliga krav för en blivande webbaserad frågeanalysmodul.

Webbapplikationerna kommer anpassas till att endast vara exekverbara i webbläsaren Internet Explorer.

2 Metod

I följande kapitel beskrivs den metod som har skapats för det här examensarbetet. Metoden är skraddarsydd och innehåller framtagna metodsteg som anses relevanta för undersökningen.

2.1 Framtagning av kriterier

Viktiga kriterier arbetas fram som bör gälla vid konverteringen av den befintliga programmodulen. I arbetet med att ta fram kriterierna bör aspekter som prestanda, datalagring, utvecklingsmiljö för konverteringsteknologi samt förvaltning av systemet tas med.

Samtliga inblandande parter i arbetet skall vara med och arbeta fram kriterierna där uppdragsgivaren bidrar med sina krav på konverteringsarbetet och handledaren på högskolan bidrar med tänkbara kriterier som anses relevanta ur en teknisk synpunkt. Studenten samarbetar med dessa parter och bidrar själv med egna relevanta kriterier. Ett seminarium skall genomföras för att arbeta fram kriterierna.

2.2 Skrivbordsundersökning

En skrivbordsundersökning genomförs som syftar till att samla in information om ett antal tänkbara teknologier som kan vara aktuella att använda för att konvertera GUTABs frågeanalysmodul.

Teknologierna studeras på ett övergripande plan, där varje teknologis fördelar respektive nackdelar identifieras av studenten. Sökningar kan genomföras på Internet eller i litteratur för att samla in fakta om teknologierna. För- respektive nackdelarna identifieras dels utifrån en generell synvinkel, dels med utgångspunkt från GUTABs frågeanalysmodul.

En utredning genomförs som syftar till att redogöra för tänkbara teknologier som kan skapa grafiska manipulerbara objekt. Skrivbordsundersökningen ligger till grund för valet av de två teknologier som ska testas. Samt den teknologi som anses mest lämplig att använda för att skapa grafiska manipulerbara objekt.

2.3 Val av konverteringsteknologier för test

Baserat på skrivbordsundersökningen fattas ett beslut om vilka teknologier som, på en teknisk nivå ska testas. På ett seminarium tillsammans med GUTAB fattas ett beslut med stöd av de fördelar respektive nackdelar som har listats med varje teknologi.

Rimligtvis testas två teknologier eftersom jämförelsen mellan de två teknologierna då blir mer konkret och rättvisande. Dessutom kan jämförelsen då omfattas av en ny och en gammal teknologi.

Valet av teknologi bör inte omfattas av mer än två teknologier då ett större antal än två kan göra utvärderingen av testningen svårhanterlig samt bli mer resurskrävande än vad som ligger inom ramen för detta examensarbete.

2.4 Litteraturstudie

En litteraturstudie genomförs som syftar till att införskaffa relevant information om ämnen som kan tänkas stödja arbetet i dess teoretiska referensram. Data kan samlas in genom exempelvis seminarier eller sökningar på Internet.

2.5 Test av konverteringsteknologier

Befintlig funktionalitet som ska konverteras väljs ut från frågeanalysmodulen i samråd med uppdragsgivaren och beskrivs kort vad den syftar till att prestera. Ett användarfall skapas för att tydliggöra detta. Eventuella parametrar som funktionen behandlar presenteras och förklaras. Val av datalagringsteknik bör göras där hänsyn tas till befintlig datalagringsteknik i frågeanalysmodulen.

Programutvecklaren väljer vilken utvecklingsmiljö som ska användas vid konverteringen. Med hjälp av den valda utvecklingsmiljön skapas sedan en testmiljö där de konverterade funktionerna återfinns skapade med de valda konverteringsteknologierna.

Testmiljön bör innehålla nödvändig information för konverteringen och ska inte bestå av avancerad grafik samt programlogik som inte berör den utvalda funktionaliteten.

Utvald VBA-funktionalitet programmeras med de valda konverteringsteknologierna. Konverteringsteknologins kapacitet att presentera samma resultat som frågeanalysmodulen genererar i Excel testas i Internet Explorer.

Webbapplikationernas skikt kan med fördel illustreras i treskiktsmodellen. Treskiktsmodellen bör tydliggöra valet att datalagringsteknik samt hur affärslogik separeras från webbapplikationens utseende

Krav från uppdragsgivaren gällande struktur av programlogik samt utseende redogörs.

2.6 Testutvärdering

Testet av de valda konverteringsteknologierna utvärderas. Utvärderingen baseras på de kriterier som har satts upp. Utvärderingen baseras på resultatet av det test som genomförs. På en teknisk nivå diskuteras resultatet av testet utifrån nedanstående aspekter;

- *Webbapplikationens storlek*
Webbapplikationernas storlek har betydelse eftersom webbapplikationen kan vara lokaliserad på servrar som GUTABs kunder tillhandahåller.
- *Programutvecklarens subjektiva bedömning*
Programutvecklaren gör en subjektiv bedömning om vilken utvecklingsmiljö som är mest användarvänlig. Bedömningen baseras på programutvecklarens generella intryck av utvecklingsmiljön. Bedömningen bör baseras på hur svårt det är att administrera objekt, skapa utseende etc.
- *Teknologins kapacitet att använda exceldokument som datakälla*
Webbapplikationerna måste kunna använda exceldokument som datakälla. Eventuella komplikationer som uppstår i samband med detta bör redogöras.
- *Teknologins kapacitet att skapa grafiska objekt*
Teknologiernas måste klara av att hantera grafiska objekt. Teknologin bör ha ett bra stöd vid användning av dessa. Arbetsinsatsen för att skapa grafiska objekt är en aspekt som bör diskuteras.
- *Användaraspekter*
Det är viktigt att ett användarvänligt utseende kan skapas med teknologin. Hänsyn tas till det generella intrycket.
- *Kommande funktioners realiserbarhet*
Teknologin kapacitet att skapa grafiska manipulerbara objekt bör undersökas.

2.7 Val av konverteringsteknologi

Ett seminarium genomförs där GUTAB beslutar vilken teknologi som är mest lämplig att använda för att konvertera företagets frågeanalysmodul. Seminariet utgår ifrån den utvärdering som genomfördes i föregående metodsteg. Valet av konverteringsteknologi motiveras med stöd av såväl den teoretiska som tekniska referensramen.

3 GUTABs frågeanalysmodul

I följande kapitel beskrivs programspråket VBA. Vidare redovisas en programbeskrivning samt en funktionsbeskrivning av GUTABs frågeanalysmodul. Kapitlet syftar till att ge en introduktion till den idag befintliga frågeanalysmodulen för att öka förståelsen för konverteringsprocessens helhet.

3.1 Microsoft Visual Basic for Applications¹

VBA är en kraftfull programutvecklingsteknik som används för att snabbt kundanpassa klient programvara och integrera den med existerande data och system. VBA omfattas av ett antal programmeringsverktyg som ingår i Microsoft Visual Basic utvecklingsystem. Microsoft Visual Basic kommer i fortsättningen att benämnas VB.

VBA möjliggör för utvecklare att köpa klientprogramvara² som kan skräddarsys till kundens specifika affärsprocesser istället för att utveckla program från grunden. Utvecklarna kan således spara tid och pengar samtidigt som de uppfyller kundens behov.

VBA tillhandahåller en komplett integrerad utvecklingsmiljö (IDE) som kännetecknas av samma element som finns tillgängliga i VB. VBA innehåller även stöd för Microsofts formulär som används till att skapa användargränssnitt mot användaren.

VBA version 6.3 ingår som en komponent i Microsoft Office XP och är integrerat med Microsoft Access, Microsoft Excel, Microsoft Frontpage, Microsoft Outlook och Microsoft Word. VBA version 5.0 är integrerat med Microsoft Office 97.

3.2 Programbeskrivning

GUTABs frågeanalysmodul är ett analysverktyg som används till att utvärdera olika kunders potential att utvecklas inom sin verksamhet. Genom frågeanalysmodulen kan GUTAB identifiera kundernas starka respektive svaga sidor och samtidigt få en uppfattning om hur kunderna själva uppfattar situationen.

Programmet består av två delar. Den ena delen utgörs av en frågeomgång med frågor som besvaras av GUTABs kunder, intervjupersoner. Den andra delen av

¹Microsoft Corp.

² Med klientprogramvara menas Microsoft Office applikationer såsom exempelvis Excel och Word.

programmet utgörs av en kontrollpanel där administratören har tillgång till ett antal funktioner som används för att analysera kundernas resultat i frågeomgångarna.

Frågeanalysmodulen startar med att en användare loggar in som administratör eller intervjuperson. Vid inloggningen anger användaren sitt fullständiga namn och vilket företag den tillhör.

Om användaren loggar in som intervjuperson startas en frågeomgång. Frågeomgången baseras på en frågemall som är kopplad till det företag som intervjupersonen tillhör. Frågorna är indelade i grupper som urskiljs med en rubrik. Varje företag har en specifik mall kopplad till sig med relevanta frågor för deras verksamhet.

Frågeanalysen går igenom samtliga frågor i frågemallen och låter intervjupersonen besvara frågorna. Efter avslutad frågeomgång sparas det sammanställda resultatet i ett exceldokument som utgörs av frågor och svar.

En användare kan även logga in som administratör och opererar då utifrån programmets kontrollpanel. Administratören utgörs alltid av en representant från GUTAB vars roll är att analysera resultaten av de olika frågeomgångarna. I arbetet med att analysera resultaten kan administratören välja att visa ett diagram över de resultat som intervjupersonen har angett

Från kontrollpanelen kan administratören även hämta sammanställningar från befintliga frågeomgångar och från enskilda personer samt spara resultaten av frågeomgångarna i ett Microsoft Word dokument som blir formaterat för utskrift.

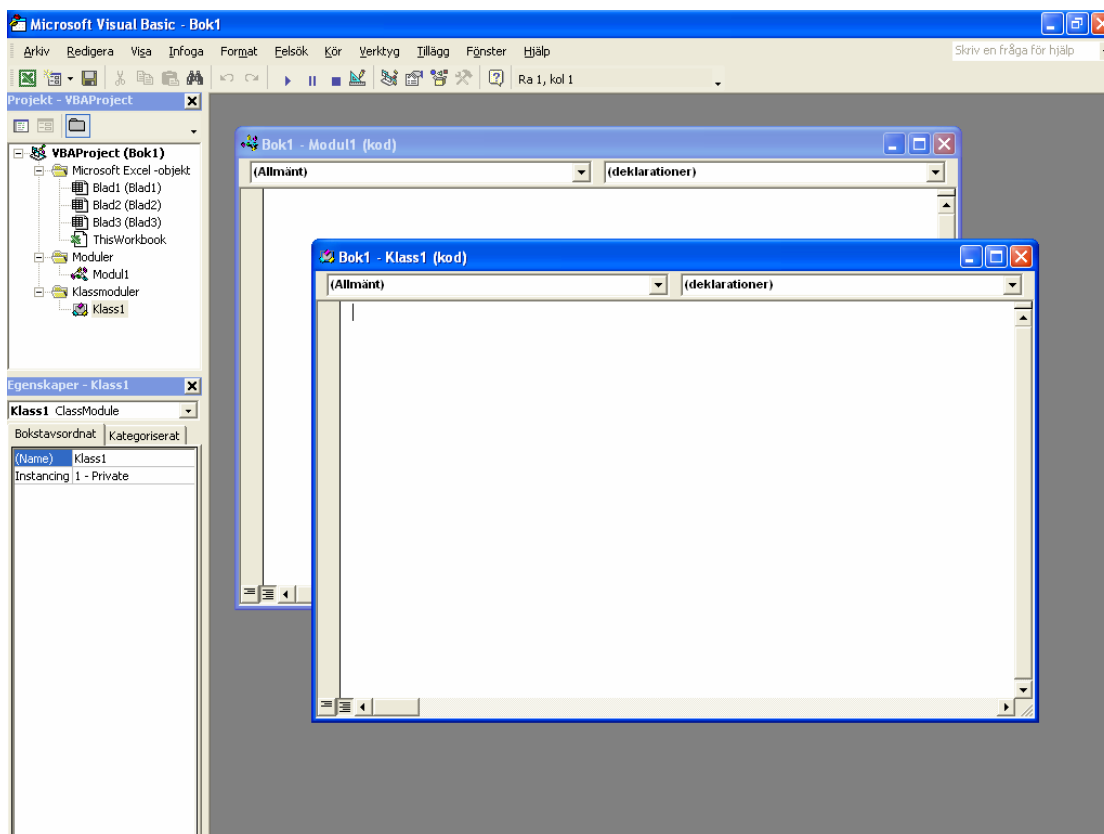
Frågeanalysmodulen innehåller även funktionalitet för att skapa egna frågeomgångar, modifiera innehållet i befintliga frågemallar samt bestämma vilken frågemall ett specifikt företag skall använda.

3.3 Funktionsbeskrivning

Utvecklingsmiljön i Microsoft Excel baseras på moduler och användarformulär. Användarformulären bildar det grafiska användargränssnittet mot användaren och modulerna innehåller applikationens affärslogik.

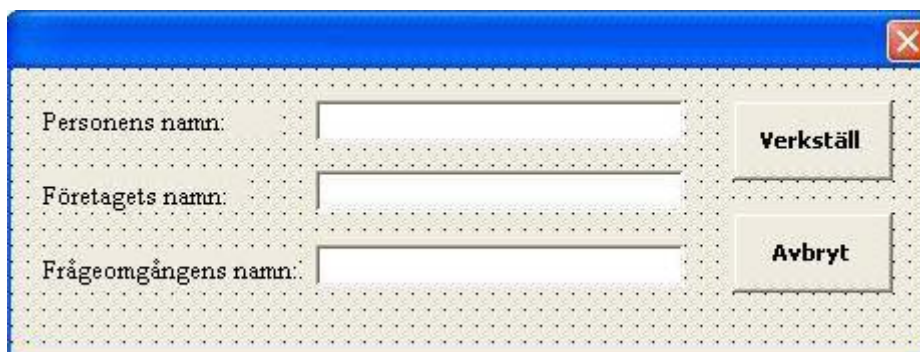
En stor del av affärslogiken i GUTABs frågeanalysmodul finns idag lokaliserad i ett antal användarformulär¹. Utvecklingsmiljön i Microsoft Excel illustreras i figur 1.2:

¹ Vanligen skrivs inte programlogik i användarformulären utan i modulerna.



Figur 1.2 – Utvecklingsmiljö i VBA

Följaktligen illustreras en grafisk komponent i figur 1.2 för att exemplifiera frågeanalysmodulens uppbyggnad med användarformulär. Bakom knappen ”Verkställ” finns funktionen *Private Sub CommandButton3_Click()*. Funktionen hämtar en sammanställning som är gjord på svaren från en enskild person.



Figur 1.3 – Formulär som används till att hämta en sammanställning som är gjord på en persons resultat.

4 Framtagning av lämplig konverteringsteknologi

I följande kapitel väljs en lämplig konverteringsteknologi som kan användas för att konvertera GUTABs frågeanalysmodul till Internet Explorer. Valet utgår från ett antal uppsatta kriterier och examensarbetets teoretiska referensram. Vidare genomförs ett test av två konverteringsteknologier där två prototyper utvecklas. Slutligen utvärderas konverteringsteknologierna baserat på det genomförda testet.

4.1 Kriterier för val av konverteringsteknologi

Lämpliga kriterier som ligger till grund för konverteringen har jag arbetat fram tillsammans med handledaren på Högskolan samt uppdragsgivaren GUTAB. Följande kriterier ligger till grund för val av två konverteringsteknologier som ska testas:

- *Dold programkod*
Programkoden ska döljas eftersom frågeanalysmodulen i framtiden kan vara aktuell för försäljning alternativt finnas tillgänglig på kunders servrar. Genom att kapsla in programkod i COM- respektive .NET-objekt som länkas ihop till så kallade "Dynamic Link Libraries"¹ skyddas programkoden.
- *Strukturerad programkod*
Programkoden i webbapplikationen bör vara strukturerad för att optimera återanvändbarhet och prestanda. En teknologi som möjliggör separering av programkod och utseende bör väljas.
- *Utvecklingsmiljö*
Utvecklingsmiljöns användarvänlighet har betydelse för GUTAB som anser att det är fördelaktigt om utvecklingsmiljön möjliggör enkel så enkel förvaltning och utveckling som möjligt.
- *Kommande funktioners realiserbarhet*
Varje teknologis kapacitet att realisera kommande funktionalitet bör undersökas och redogöras.
- *Teknologins framtidsstatus*
Teknologin ska vara tillräckligt modern i dagsläget och ska förbli det i några år till.
- *Befintlig funktionalitet*
Teknologin ska ha kapacitet att återskapa all funktionalitet som idag finns lokaliserad i GUTABs frågeanalysmodul.

¹ "Dynamic Link Libraries" kommer i fortsättningen att benämnas DLL

- *Prestanda*
Webbapplikationerna ska kunna presentera frågorna i frågeomgången utan märkbart dröjsmål.
- *Webbläsarkapacitet*
Den konverterade frågeanalysmodulen behöver endast fungera i webbläsaren Internet Explorer med en skrämpplösning på 800 x 600 pixlar.

4.2 Skrivbordsundersökning

Skrivbordsundersökningen presenterar de fördelar samt nackdelar som har identifierats med ett antal utvalda teknologier. Skrivbordsundersökningen har delats in i två delar där del 1 omfattas av serverbaserade teknologier och del 2 omfattas av klientbaserade teknologier.

4.2.1 Skrivbordsundersökning, del 1

I skrivbordsundersökning första del har fyra tänkbara teknologier undersökts. Utifrån GUTABs frågeanalysmodul har fördelar och nackdelar identifieras för varje teknologi.

Fördelar med ASP

- ASP används i störst utsträckning tillsammans med Microsoft webbserver Internet Information Server, databasen msSql och operativsystemet Windows.
- Teknologin stöder användandet av affärsobjekt (COM-objekt) som är utvecklade i VB 6.
- Färdiga komponenter finns tillgängliga som är anpassade till ASP vilket gör att programutvecklaren kan skapa grafiska objekt med små resurser.
- Teknologin är modern och kommer att vara i några år till.
- Programlogik på ASP-sidor skrivs i de flesta fall med VBScript vilket härstammar ifrån VB vilket även VBA gör.

Nackdelar med ASP

- Användarformulären som genererar utseendet i GUTABs frågeanalysmodul går inte att återskapa i ASP. Däremot går det att skapa ett liknande utseende med HTML-taggar.

- Teknologin skiljer på klient och server vilket gör det omöjligt att tillämpa den händelsestyrda modellen i ASP applikationer.
- Teknologin kan inte av egen kraft skapa grafiska objekt utan måste använda sig av färdigutvecklade COM-objekt som kan generera diagram.

Fördelar med ASP.NET¹

- ASP.NET baseras på en liknande utvecklingsmiljö som VBA där även en liknande programmeringsmodell tillämpas.
- Att utveckla webbsidor med ASP.NET liknar mer traditionell programutveckling.
- ASP.NET innehåller många nya kraftfulla objekt som har stor kapacitet. Klassen System.Drawing möjliggör för programutvecklaren att skapa grafiska objekt såsom olika typer av diagram².
- ASP.NET ger programutvecklaren mycket gratis i utvecklingsarbetet eftersom den inte skiljer på klient och server. ASP.NET applikationer tillämpar således den händelsestyrda modellen.
- En webbapplikation som är gjord med ASP.NET kan innehålla COM-objekt. Komponenterna inkluderas då i projektet för webbapplikationen och programutvecklaren behöver då inte administrera komponenten själv.
- Teknologin har stöd för att läsa respektive skriva data till respektive från exceldokument
- Programlogik kan skrivas med VB.net som härstammar från VB. Det bör dock tilläggas att en ASP.NET applikation likväl kan utvecklas i andra programspråk³.
- Ett installerat .NET Framework kan tillsammans med operativsystemet betraktas som en del i .NET.l
- ASP.NET är en vidareutveckling av ASP och betraktas idag som framtidens teknologi.
- .NET Framework är kostnadsfritt

¹ Chris Payne, 2002

² Det bör dock tilläggas att det idag finns färdiga .NET-objekt som använder System.drawing

³ Se möjliga andra programspråk på följande webbsida; [http:// www.asp.net](http://www.asp.net)

Nackdelar med ASP.NET

- Priset för Microsofts utvecklingsverktyg VS.net är högt.

Fördelar med PHP¹

- PHP har en hög prestanda eftersom programkod exekveras i PHPs minnesutrymme. och kommunicerar således inte med t.ex. COM-objekt som andra teknologier gör.
- Teknologin är plattformsoberoende.
- PHP stöder användandet av COM-objekt. I dessa inkluderas även COM-objekt som kan generera diagram.
- PHP kan skriva och läsa data till och från Exceldokument
- PHP består av öppen källkod vilket leder till att teknologin ständigt kan utvecklas samt att uppgradering till nyare versioner blir kostnadsfria.

Nackdelar med PHP²

- Teknologin baseras inte på Microsofts plattform med Microsofts programspråk. Detta ger en stor arbetsinsats vid en eventuell konvertering.
- Teknologin används i stor utsträckning tillsammans med webbservern Apache, databasen mySql samt operativsystemet Linux
- PHP, kan inte på egen kraft skapa grafiska objekt.

Fördelar med JSP³

- Teknologin kan köras på olika typer av webbservrar och är plattformsoberoende.
- JSP kan användas tillsammans med Javabeans⁴.
- JSP använder inte scriptspråk vilket anses stabilt i komplexa system.
- JSP kan läsa och skriva data till exceldokument och använda det som datakälla i webbapplikationer.
- Javabeans som genererar diagram finns tillgängliga på Internet.

¹ Harish Kamath, 2002-04-23

² Gbdirect

³ Sun Microsystems

⁴ Javabeans är motsvarigheten till Microsofts COM-objekt

Nackdelar med JSP

- JSP baseras på Javas programspråk som inte påminner om VBA.
- Komponenter baseras på Javabeans eller JSP-taggar. Detta kräver en stor arbetsinsats vid en eventuell konvertering.
- JSP kan inte på egen kraft, skapa grafiska manipulerbara objekt

4.2.2 Skrivbordsundersökning, del 2¹

GUTAB finner det intressant att veta vilken/vilka teknologier som kan hantera grafiska manipulerbara objekt på Internet². GUTABs definition av ett grafiskt manipulerbart objekt är;

- ”Funktionalitet som möjliggör för användaren att manipulera ett grafiskt objekt på Internet genom att klicka och dra i objektet”

Företaget Netronic³ har utvecklat applikationer med teknologierna ActiveX samt Java Applets där ovannämnda funktionalitet finns applicerad. Dessa applikationer är intressanta eftersom de kan vara en del i en vidareutveckling av GUTABs blivande webbapplikation.

Teknologierna fördelar och nackdelar redogörs nedan:

Fördelar med ActiveX

- ActiveX komponenter kan utvecklas med Microsoft VB 5.0, Visual C++ 5.0 samt Visual J++
- Stora tillverkare som säljer ActiveX komponenter stöder Java support.
- ActiveX komponenter som är utvecklade med olika programspråk kan lätt interagera med varandra.
- Utvecklare kan lätt och snabbt skapa unika och interaktiva webbsidor som inte behöver byggas från grunden eftersom det idag finns över 1000 återanvändbara activeX komponenter som är tillgängliga på marknaden
- En activeX komponent kan använda kraftfulla APIs, däribland DirectX, OpenGL and QuickDraw 3D.

¹ Merlin Hughes, March 1997

² Motsvaras av kriteriet ”Kommande funktioners realiserbarhet” i kriterielistan

³ <http://www.netronic.com/english/index.html>

- ActiveX är en väldigt kraftfull teknologi eftersom säkerhetsnivån är låg för access till program på klientens dator.
- ActiveX komponenter kan interagera med nästan alla program på klientens dator eftersom den har full access till operativsystemet. En ActiveX komponent skriven med VB kan exempelvis öppna ett fönster från en C++ applikation och sedan använda det aktuella fönstret i en Java Applet.
- ActiveX kan presentera program i webbläsaren som kan hantera grafiska manipulerbara objekt.

Nackdelar med ActiveX

- ActiveX komponenter har full access till klientens hårddisk. Digitala signaturer är den enda säkerhetsmekanismen som finns.
- En activeX komponent som är bekräftad med en digital signatur kan tvingas in i egenmäktiga operationer som t.ex. att starta program på hårddisken Detta kan leda till stora skador på klientens hårddisk.
- ActiveX är en klientbaserad teknologi vilket innebär att programlogik för att hantera operationer mot en webbserver måste skrivas i ett serverbaserat programspråk

Fördelar med Java applets

- Java applets baseras på "Java Security Model" som skyddar användaren från fel som kan komma ifrån exempelvis felaktiga datatypkonverteringar eller olagliga minnes accesser.
- Javas säkerhetsmekanism upptäcker problem med hantering av parametrar vilket gör att program inte kan startas på klientens dator.
- Java applets kan presenteras i webbläsarna Internet Explorer, Netscape Navigators samt Opera.
- Java applets möjliggör hantering av grafiska manipulerbara objekt på Internet.

Nackdelar med Java applets

- Java applets kan endast utvecklas med programspråket Java
- Java applets har endast stöd för att använda ett fåtal APIs
- Java applets har en begränsad åtkomst till operativsystemet som den exekveras på och blir följaktligen mindre kraftfull.

4.3 Val av konverteringsteknologier för test

Teknologierna som undersöks i skrivbordsundersökningens första del har många liknande egenskaper. Ett antal fördelar med teknologierna, utifrån GUTABs frågeanalysmodul sett, har dock större betydelse vid valet av vilka konverteringsteknologier som ska testas.

Samtliga teknologier i skrivbordsundersökning del 1 kan manipulera exceldokument vilket är en grundförutsättning för att teknologin ska väljas. Vidare visar undersökningen att PHP och JSP kan köras på olika plattformar och på olika webbservrar. Detta har dock ingen större betydelse i sammanhanget eftersom Microsofts plattform prioriteras i första hand.

Samtliga teknologier i skrivbordsundersökning 1 stöder användandet av affärsobjekt. Affärsobjekten i de olika teknologierna utvecklas dock med olika programspråk där JSP använder programspråket Java. PHP och ASP använder affärsobjekt som utvecklas med VB och ASP.NET använder VB.NET (affärsobjekt i ASP.NET benämns .NET-objekt).

Samtliga teknologier har således kapacitet att utföra, i stort sett, samma saker. Det avgörande för valet av vilka teknologier som ska testas blir därmed vilket programspråk som teknologin baseras på.

ASP och ASP.NET baseras på VB vilket kommer att medföra den tekniskt sätt minsta arbetsinsatsen vid en eventuell konvertering. Dessutom ses detta som positivt ur förvaltningssynpunkt eftersom VBScript och VBA baseras på VB¹.

VBA-applikationen kommer att konverteras i första hand och för detta väljs teknologierna ASP som använder COM-objekt utvecklade i VB6 samt ASP.NET som använder .NET-objekt. Det har visat sig att programspråken som teknologierna baseras på utgör den avgörande skillnaden varför valet av teknologier för test görs utifrån detta kriterium.

Båda teknologierna baseras på VBScript, VB samt VB.NET som är nära besläktat med VBA och kommer därför att ge den minsta arbetsinsatsen vid konverteringen. Dessutom har GUTAB erfarenhet av VB samt VBA programmering och kommer därmed att kunna underhålla den kommande webbapplikationen med egna resurser.

¹ VB.net är en vidareutveckling av VB

4.4 Teoretisk referensram

Arbetets teoretiska referensram syftar till att ge en introduktion till de teknologier samt tekniker som kommer att användas för att konvertera den utvalda funktionaliteten ifrån GUTABs frågeanalysmodul. Vidare introduceras de teknologier med vilka grafiska manipulerbara objekt kan skapas.

Den teoretiska referensramen beskriver Internets uppbyggnad för att ge en ökad förståelse för hur serverbaserade webbprogrammeringsteknologier fungerar. Vidare beskrivs de två klientbaserade teknologierna ActiveX och Java applets, samt Microsofts två webbaserade teknologier ASP samt ASP.NET.

Klient/Server-modellen¹

Internet baseras idag på klient/server-modellen som utgår ifrån att två datorer samarbetar med att utföra en uppgift och skickar information mellan varandra. Dessa två datorer utgörs av en klient och en server.

Vidare fungerar klient/server modellen på följande sätt: Klientdatorn skickar en begäran till serverdatorn och klienten får tillbaka den begärda mängden data formaterat exempelvis som ett HTML-dokument². Detta scenario kallas för begäran/svarsmodellen och är en viktig del i klient/servermodellen.

Serverdatorn i denna modell är en dator som lagrar information om en webbplats. Klienten utgörs av en webbläsare som vill tolka webbsidan.

ActiveX³

ActiveX är ett set av teknologier från Microsoft som används för att göra webbplatser interaktiva. ActiveX kan paketera program som är utvecklade i ett antal olika programspråk till ett objekt som antingen exekveras i en programutvecklingsmiljö, ActiveX program, eller i en nätverksmiljö, ActiveX kontroll.

ActiveX baseras på OLE (Object Linking and Embedding) som ursprungligen användes till att dela information och format mellan två datorer på en dator. ActiveX komponenter blir tillgängliga för olika applikationer genom Microsoft Common Objekt Model (COM).

ActiveX komponenter kan skrivas i ett antal olika programspråk. VB är dock det vanligaste programspråket men activeX komponenten kan likväl utvecklas med Visual C++ och Java.

¹ Chris Payne, 2002

² Kan även var XML, XSL, PDF-dokument etc.

³ Eric J. Grossman, 2000

ActiveX bidrar med en hög grad av interaktivitet till webbplatser och möjliggör presentation av multimedia effekter, interaktiva objekt och sofistikerade applikationer i klientens webbläsare.

Java applets¹

Java program kan delas in i två kategorier, applikationer och applets. Skillnaden mellan en Java applikation och en Java Applet är att applikationen är ett självständigt program som exekveras lokalt på en arbetsstation.

Java Applet är en speciell typ av Java program som är anpassade för att kunna exekveras direkt från en Javakompatibel webbläsare. Applets anses i störst utsträckning vara mest lämpliga att använda i olika typer av webbprojekt.

Java Applet möjliggör för programutvecklaren att skapa dynamiska interaktioner samt levande animationer på en statisk HTML-sida.

Affärsobjekt²

Affärsobjekt används för att hålla samman viktig programkod som inte hanteras i användargränssnittet. Affärsobjektet omfattas av viktig affärslogik i webbapplikationen.

I webbapplikationer tillämpas ofta treskiktsmodellen vilket innebär att programmet delas in i tre skikt som omfattas av ett presentationsskikt, ett skikt för affärslogik/objekt samt ett dataskikt. Användandet av komponenter leder till att skiktet för affärslogiken/objekten tydliggörs i applikationen.

Genom att använda affärsobjekt blir det lättare att separera och definiera programmet och kod som inte ska återfinnas i gränssnittet finns i affärsobjektet istället för på sidorna som genererar webbplatsens utseende.

Det faktum att koden kapslas in i affärsobjektet leder till att mängden återanvändbar kod ökar samt att programkoden inte blir tillgänglig för icke programutvecklare. Den separerade kompilerade programkoden ökar även prestandan i applikationen. Ytterligare en vinst med att använda affärsobjekt är att resultatet av en ändring kan påverka många olika delar i webbapplikation.

¹ Liang, Y. Daniel, 2001

² Chris Payne, 2002

Active Server Pages¹

Internet bestod tidigare till stor del av statiska webbsidor som fokuserades på utseendet. Idag har Internet förändrats och det finns ett behov av att utveckla dynamiska webbsidor med en hög grad av interaktivitet.

Microsofts har utvecklat en teknologi som heter Active Server Pages med vilka dynamiska och interaktiva webbsidor kan skapas. ASP kan betraktas som ett ramverk bestående av objekt och rutiner som möjliggör för programmeraren att blanda programlogik med HTML-kod på en webbsida.

ASP-sidorna måste lagras på en webbserver som bland annat har till uppgift att tolka ASP-sidans programlogik. Microsofts webbserver Internet Information Server används i störst utsträckning och kan installeras på operativsystemen Microsoft Windows, 2000, NT och XP². Funktionalitet och programlogik på ASP-sidan skrivs med olika scriptspråk där Visual Basic Scripting Edition, även kallat VBScript, används i störst utsträckning.

ASP är ett serverbaserat ramverk vilket innebär att all programlogik som finns på ASP-sidan genereras av webbservern innan den skickas till klienten, som i det här fallet utgörs av en webbläsare.

Interaktiviteten på ASP-sidorna baseras till stor del på information som finns lagrad i databaser. Via ASP kan programmeraren manipulera data i databasen och styra webbplatsens innehåll efter vad klienten vill se. I ASP kan även serverns filsystem manipuleras på olika sätt.

Active Server Pages .NET³

Active Server Pages.NET är en teknologi som är utvecklad av Microsoft som gör det möjligt att enkelt skapa och kontrollera dynamiska webbsidor. ASP.NET är Microsofts senaste version av ASP som baseras på en ny programmeringsmodell.

Tidigare i detta kapitel beskrevs klient/server-modellen där det tydliggörs att webbserverns hantering av ASP-sidor utgår ifrån begäran/svars – modellen. ASP.NET baseras på en annan modell för kommunikation mellan två datorer, den händelsestyrda modellen. I praktiken kan detta betraktas som en kombination av begäran/svars modellen och den händelsestyrda modellen eftersom .NET miljön simulerar en händelsestyrd modell.

Denna modell utgår ifrån att webbservern är inställd på att vänta på att något ska hända hos klienten. När detta sker vidtar servern åtgärder och utför uppgiften utan att klienten har begärt att webbservern ska utföra uppgiften.

¹ Scott Mitchell m.fl., 2000

² Alternativt kan Personal Web Server installeras på Windows 95/98.

³ Chris Payne, 2002

ASP.NET knyter därmed ihop klient och server¹ vilket leder till att programutvecklaren inte behöver fokusera på att hantera och svara på begäran från klienten i samma utsträckning som i ASP.

I ASP.NET skrivs programlogik i ett ramverk som heter .NET Framework vilket är en uppsättning objekt och mallar som Microsoft tillhandahåller. .NET Framework innehåller bakomliggande funktioner som ASP.NET har tillgång till. Samtliga program som har utvecklats i .NET består av nyckelfunktioner som gör dem kompatibla, säkra och stabila.

Utveckla COM-objekt i ASP respektive .NET-objekt i ASP.NET

En stor nackdel med att använda COM-objekt, som är utvecklade med VB6, är att COM-objektet måste registreras i Windows registry för att operativsystemet ska kunna exekvera den. När sedan affärslogik i COM-objektet uppdateras måste den avregistreras och sedan registreras om vilket kräver onödigt arbete.

Problemet med att registrera COM-objekt har lösts i ASP.NET. .NET-objektet sparas automatisk i en katalog i filsystemet som heter bin. Detta medför att affärsobjektet automatisk hittas när webbapplikationen exekveras. Komponenten behöver endast kopieras till bin katalogen.

I ASP.NET blir inte det kompilerade affärsobjektet² låst när det exekveras vilket innebär att programmeraren inte behöver avsluta den process som den aktuella exekveringen har erövrat hos webbservern. **Flera versioner av samma komponent med samma namn kan dessutom samexistera.**

Därmed blir hanteringen av komponenter som är utvecklade med VB6 väldigt omständlig då webbserverns process som hanterar exekveringen av en viss applikation måste avbrytas för att kunna registrera om den aktuella DLLen.

Fördelar med ASP.NET jämfört med ASP

Skillnaden mellan att utveckla webbsidor i ASP.NET jämfört med ASP är stor. Metodiken för att utveckla webbapplikationer skiljer sig radikalt. Grunden till ASP är operativsystemet Windows samt webbservern Internet Information server. Dessa två delar har alltid varit separerade varför även ASPs funktionalitet har varit begränsad. Följande fördelar har identifierats:

ASP.NET ingår som en del av operativsystemet inom ramen för .NET. ASP.NET använder således många objekt som traditionella program får tillgång till i alla .NET applikationer. Detta kan jämföras med ASP som endast består av sex inbyggda objekt³.

¹ I ASP.NET exekveras klientscript i bakgrunden som tar reda på aktuella händelser hos klient.

² Affärsobjekt, komponent och DLL kan för det här arbetet betraktas ha samma innebörd.

³ Genom att skapa objekt explicit kan man dock använda fler än 6 objekt i ASP.

ASP baseras på teorin att klienten och servern är separerade medan ASP.NET har bevisat motsatsen. ASP.NET knyter ihop klient- och serverkod vilket leder till att webbutveckling i .NET mer liknar traditionell programutveckling och baseras inte längre på begäran/svarsmodellen.

ASP använder tolkade skriptspråk medan ASP.NET använder kompilerade programspråk. Programlogik i ASP.NET kan skrivas med ett antal olika programspråk såsom VB, C++ och C#.

ASP.NET har programmeringstekniska förbättringar jämfört med ASP som exempelvis större enkelhet och bättre säkerhet. Dessutom är cachningen mycket mer utvecklad i ASP.NET.

En stor fördel med sessionshanteringen i ASP.NET är att sessionerna går att vidmakthållas över flera webbservrar¹. Sessionshanteringen har blivit förenklad och mer kraftfull eftersom den hanterar problematiken med ett inbyggt sessionsstöd. Detta har gjort att sessionshanteringen i ASP.NET har blivit väldigt tillförlitlig och fungerar även om klienten inte accepterar cookies.

¹ En begränsning med sessioner i ASP var att det inte kunde bevaras mellan olika webbservrar.

4.5 Test av konverteringsteknologier

Följande kapitel syftar till att beskriva konverteringsarbetet i sin helhet. Funktionalitet som ska konverteras definieras av uppdragsgivaren. Utifrån kraven¹ skapas ett användarfall för att illustrera flödet i de blivande webbapplikationerna.

Webbapplikationernas uppbyggnad enligt treskiktsmodellen illustreras samt den testmiljö som används i konverteringsarbetet. Slutligen presenteras de utvecklingsverktyg och APIs som används i för konverteringen.

4.5.1 Val av funktionalitet

I samråd med GUTAB har funktionalitet som ska konverteras valts ut. GUTAB anser att det är intressant att visa vilka möjligheter det finns att konvertera en del av programlogiken i frågeanalysmodulen. Det är även intressant att visa vilken kapacitet konverteringsteknologierna har att återskapa grafiska objekt (diagramkomponenter) som idag finns i frågeanalysmodulen.

4.5.2 Användarfall

Utifrån utvald funktionalitet har följande användarfall skapats:

1. Användaren går igenom ett antal frågor. En fråga visas åt gången och användaren besvarar frågorna.
2. Användarens resultat sparas i en frågeomgång i ett Exceldokument.
3. Användaren klickar på knappen "Visa resultat" och resultatet av varje fråga visas tillsammans med ett uträknat medelvärde på alla frågor som tillhör en rubrik.
4. Användaren klickar på knappen "Visa diagram" och ser ett linjediagram som visar Rubrikernas medelvärden.

¹ Kraven har inte specificerats i en kravspecifikation eftersom konverteringsarbetet inte anses vara ett "utvecklingsprojekt" utan betraktas följaktligen endast som ett steg i konverteringsarbetet.

4.5.3 Treskiktsmodell tillämpad på webbapplikationerna

Webbapplikationerna är uppbyggda enligt treskiktsmodellen som syftar till att separera programlogik (VB.NET och VBScript) från utseende (HTML) i en webbapplikation. Treskiktsmodellen består av följande tre skikt (se Figur 1.4):

Presentationsskikt

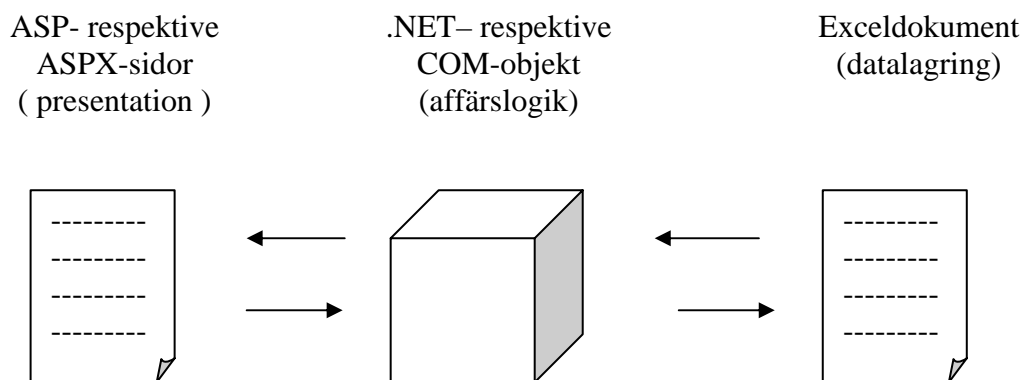
Presentationsskiktet i webbapplikationerna omfattas av ASP- respektive ASPX-sidor. Dessa sidor presenterar applikationernas användargränssnitt med html-kod. Tabeller, formulär, knappar och typsnitt för text återfinns på dessa sidor.

Skikt för affärslogik

I skiktet för affärslogik finns de COM- respektive .NET-objekten som utgör webbapplikationernas affärslogik. Objekten kommunicerar med de angivna datakällorna, utför matematiska beräkningar samt hanterar inläsning av data ifrån klienten.

Datalagringskikt

I webbapplikationen utgörs datakällorna av exceldokument som tillhandahåller frågor samt information om företag, intervjupersoner samt vilken behörighetsgrupp intervjupersonerna tillhör.



Figur 1.4 – Treskiktsmodellen tillämpad på webbapplikationerna.

4.5.4 Testmiljö

Webbapplikationerna skapas i en testmiljö som är anpassad till en skrämpupplösning på 800x600 pixlar. Testmiljön finns lokaliserad på en webbserver och fungerar som lagringsplats för webbapplikationerna. I testmiljön skapas det utseende samt den affärslogik som kommer att bilda webbapplikationerna.

4.5.5 APIs¹

För att kunna konvertera funktionalitet ifrån GUTABs frågeanalysmodul används ett antal klassbibliotek i ASP.NET samt VB6.

Följande APIs används i ASP.NET-applikationen:

- *ChartFX for .NET*
- *Microsoft Excel 10.0 Object Library*

Följande APIs används i ASP-applikationen:

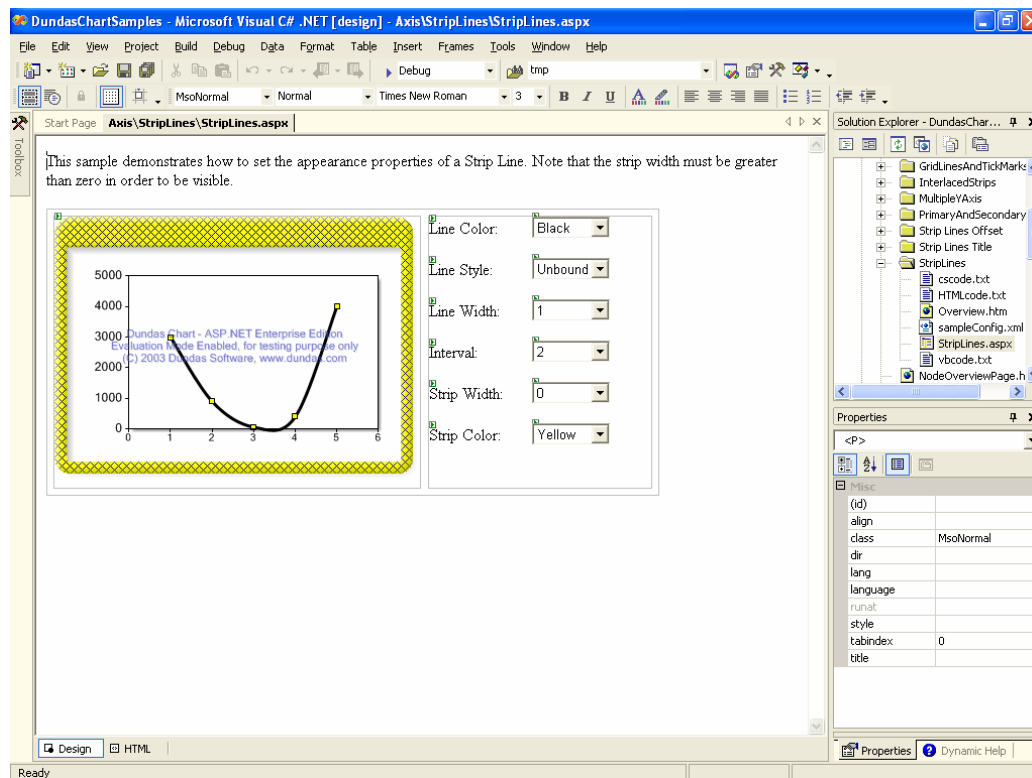
- *ChartFX for VB6*
- *Microsoft Excel 10.0 Object Library*

4.5.6 Utvecklingsverktyg för testning

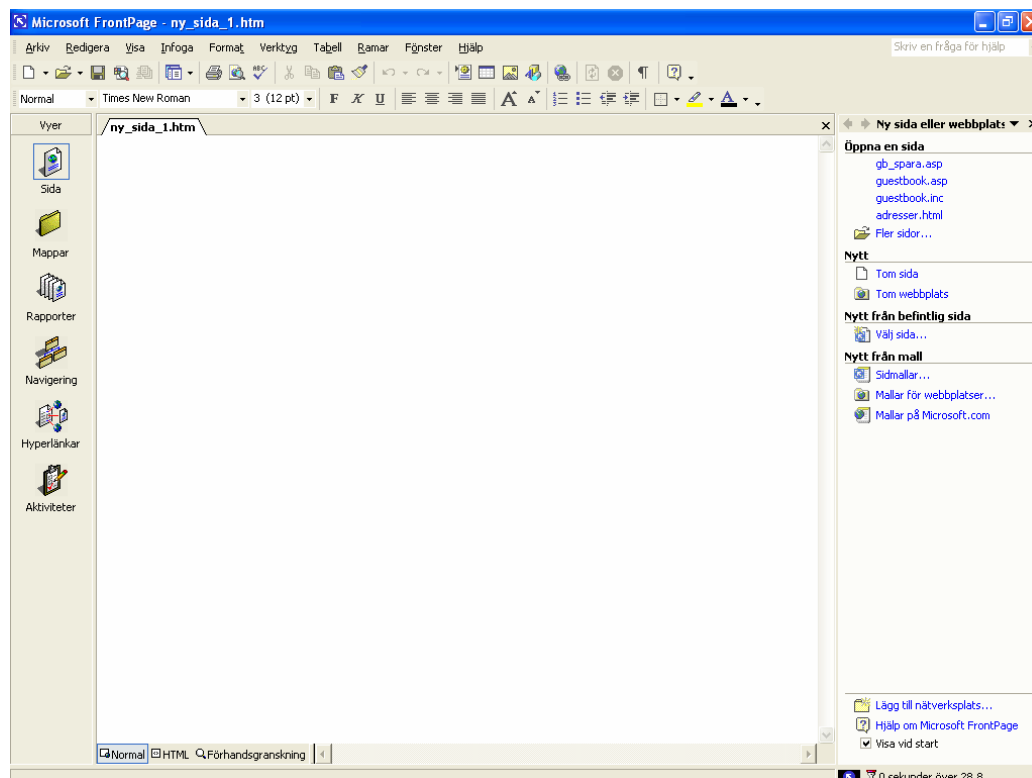
I konverteringsarbetet används följande tre programvaror för att utveckla webbapplikationerna:

- Microsoft Visual Studio.net Enterprise Architect för ASP.NET applikationen.
- Microsoft Frontpage för utveckling av ASP-sidor i ASP applikationen.
- Microsoft Visual Basic 6.0 Enterprise Edition för COM-objekt i ASP-applikationen.

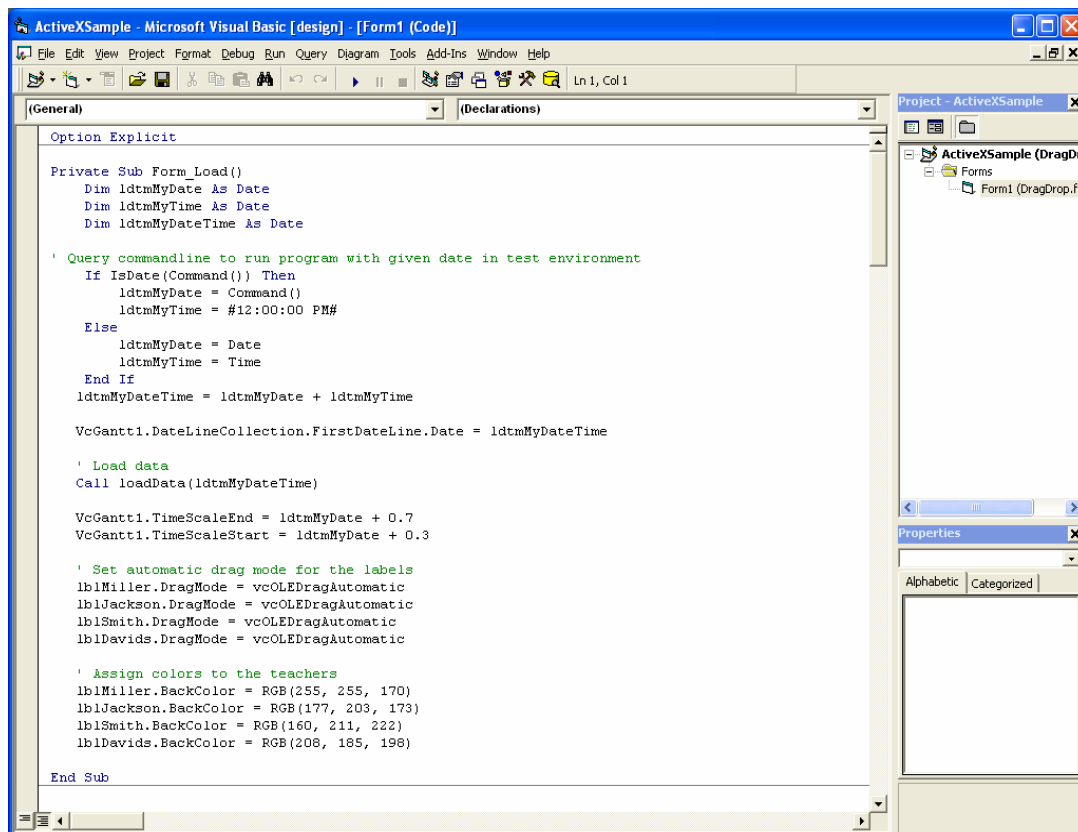
¹ Kapitlet redogör endast för de APIs som manuellt har lagts till för webbapplikationerna.



Figur 1.5 – Utvecklingsmiljön i Microsoft Visual Studio .NET



Figur 1.6 – Utvecklingsmiljön i Microsoft Frontpage



Figur 1.7 – Utvecklingsmiljön i VB6

4.6 Testutvärdering

Utvald funktionalitet från GUTABs frågeanalysmodul har konverterats med teknologierna ASP samt ASP.NET. Ett underlag finns därmed för att utvärdera teknologiernas kapacitet att återskapa programlogik och grafiska objekt från GUTABs frågeanalysmodul till webbläsaren Internet Explorer.

All utvald funktionalitet har konverterats och visar därmed att hela GUTABs frågeanalysmodul kan konverteras till Internet Explorer.

De grafiska objekten kunde återskapas, dock endast med externa resurser i form av befintliga COM- respektive .NET-objekt. Företaget Software FX¹ har utvecklat ovannämnda objekt som integrerades med webbapplikationerna.

De grafiska objekten skapar linjediagram som GUTAB ansåg var kritiska att konvertera. Denna aktivitet medför en liten fördröjning när webbapplikationerna exekveras eftersom diagrammet sparas som en bild innan det presenteras i webbläsaren. Vidare finns det begränsningar i antalet tecken som kan tilldelas värdena på x-axeln. Vid för långa rubriknamn uteblir två av tre rubriknamn i diagrammet.

Webbapplikationerna sparar användarens resultat i exceldokument. Exceldokumentet skapas varje gång en användare har angett sina svar. Denna aktivitet är märkbart tidskrävande eftersom ett nytt exceldokument skapas varje gång webbapplikationen exekveras.

Programspråken har bra stöd för de egenutvecklade COM respektive .NET-objekten och kommunikationen mellan asp- respektive ASPX-sidorna och COM respektive .NET-objekten fungerar utan anmärkningar.

Webbapplikationerna presenterar frågorna i frågeomgången utan märkbart dröjsmål, noterbart är dock att användarens Internetuppkoppling kommer att ha betydelse för det generella intrycket av webbapplikationens prestanda.

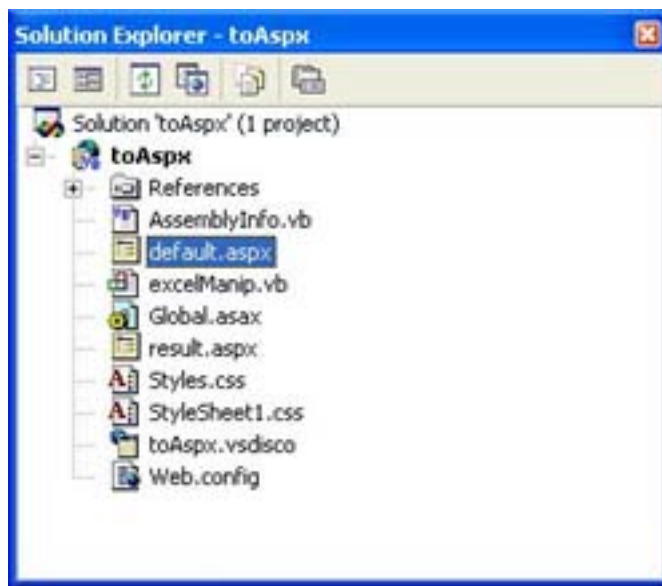
Under utvecklingsarbetets gång har jag som programutvecklare haft möjligheten att iaktta utvecklingsmiljöerna och kan därför nu ge en subjektiv bedömning av dessa med fokus på utvecklingsmiljöernas grad av användarvänlighet.

Utvecklingsverktyget VS.NET, som användes för att utveckla ASP.NET applikationen, är konstruerat så att programutvecklaren inte behöver spendera energi på omkringliggande uppgifter. Verktyget genererar HTML-kod automatiskt, administrerar .NET-objekt dynamiskt samt möjliggör för programutvecklaren att på ett väldigt enkelt sätt skapa webbapplikationens utseende genom den så kallade "klicka & dra" principen som tillämpas på grafiska objekt.

¹ En avgränsning gjordes till att endast testa Software FXs produkter, <http://www.softwarefx.com>

Någon motsvarighet till VS.net för ASP finns inte. Efter att ha utvecklat två webbapplikationen kan jag konstatera att det underlättar väldigt mycket för programutvecklaren när en stor del av utvecklingsarbetet genereras automatiskt av VS.net.

Ytterligare en stor fördel med ASP.NET är att en webbapplikations filer skapas i ett projekt enligt strukturen i figur 1.5. I projektet finns de filer som programutvecklaren arbetar med, vissa skapas automatiskt av VS.net och vissa skapar programutvecklaren manuellt.



Figur 1.5 – En projektfil exemplifierat enligt den mappstruktur som skapas i VS.net för en ASP.NET applikation.

Då samtliga filer som ingår i webbapplikationen återfinns i projektet blir både förvaltnings- och utvecklingsarbetet väldigt effektivt. Dessutom kan webbapplikationen testsköras i utvecklingsmiljön genom att använda VS.NETs inbyggda webbläsare.

Programlogiken i webbapplikationerna är skriven i de .NET- och COM-objekt som tidigare har nämnts. En stor skillnad kan urskiljas vid administrering av dessa. I ASP.NET ingår .NET-objektet i projektfilen och all hantering av densamma sköts av VS.net. Dessutom länkas hela projektet ihop till en DLL vid kompilering.

I ASP-applikationen får däremot programutvecklaren hantera COM-objektet själv. Vid kompilering länkas COM-objektet ihop till en DLL som måste registreras i operativsystemet.

Programutvecklaren måste spara om DLLen och ersätta den med en ny när ändringar har gjorts. Detta gör VS.net automatiskt med .NET-objektet. För att underlätta test av COM-objektet kan en konfiguration göras så att COM-objektet kan användas i runtime-läge utan att objektet är registrerat i operativsystemet.

De teknologier som har testats kan generera identiska användargränssnitt. Dessa kan dock inte bli identiska med gränssnittet i GUTABs befintliga frågeanalysmodul. Detta eftersom den är uppbyggd med användarformulär vilket inte går att åstadkomma med vare sig ASP eller ASP.NET.

Kommande funktioners realiserbarhet har undersökts med syftet att redogöra för vilket stöd teknologierna i skrivbordsundersökning 2 har att skapa grafiska manipulerbara objekt. Detta redogörs i kapitlet "Slutsatser".

Webbapplikationerna är anpassade till en skärmupplösning på 800*600 pixlar. Detta är justerat med hjälp av de tabeller som styr bredden och höjden på den aktuella sidan.

Ett antal skillnader kunde urskiljas i webbapplikationernas tekniska lösningar. Det konstaterades att webbapplikationerna är identiska utseendemässigt och innehåller samma funktioner. Däremot kunde inte alla problem lösas på samma sätt då ASPs kapacitet var begränsad.

En viktig frågeställning som uppstod under utvecklingsarbetets gång var hur svaren som anges i frågeomgången kunde mellanlagras för varje gång en sida besöktes. Applikationerna är uppbyggda så att användaren svarar på en fråga i frågeomgången varefter samma sida laddas om och presenterar en ny fråga.

I ASP-applikationen användes sessions objekt för att mellanlagra svaren på servern för varje gång sidan laddades om. En variabel av typen array deklarerades som en sessionsvariabel och lagrade samtliga svar i frågeomgången.

Eftersom ASP.NET är en vidareutveckling av ASP och därmed har utvidgad kapacitet gentemot ASP testades möjligheterna att deklarera motsvarande lagringsmedia i ASP.NET för svaren i frågeomgången som en "Shared"¹ variabel. Lagringsmediet utgjordes av ett datatable- objekt som fanns lokaliserat i ASP.NET applikationens .NET-objekt. Genom att använda denna typ av deklaration kunde lagringsmediet sparas i .NET-objektet för varje gång sidan laddas om och inte i sessionsobjektet som i ASP-applikationen.

Det visade sig dock att detta inte var en bra lösning. Det upptäcktes att om två klienter eller fler öppnade objektet samtidigt fick dessa dela på lagringsmediet. Detta innebar att samtliga klienters operationer mot objektet sparades i samma objekt. Detta förutsatt att klienterna använde objektet vid samma tidpunkt.

¹ Möjligheten att deklarera en variabel som shared är en egenskap som tillhör vb.net.

Att mellanlagra svaren i sessionsvariabler kan resultera i problem. Om klienten inte accepterar cookies fungerar inte webbapplikationerna. ASP.NETs sessionshantering är vidareutvecklad och mer säker än sessionshanteringen i ASP vilket kan betraktas som fördelaktigt.

Det bör tilläggas att ett array-objekt är ett mindre kraftfullt objekt än en datatabell, detta har dock ingen större betydelse eftersom objekten endast används till att lagra en persons svar i frågeomgången.

Vidare kan en skillnad i lagringsmedia för frågeomgångens frågor identifieras. Frågorna finns lagrade i exceldokument och placeras in i en datatabell i ASP.NET. Motsvarande objekt i VB6 som heter recordset har använts.

Testet som genomfördes visade att användandet av exceldokument som datakälla bidrog till två komplikationer. När webbapplikationerna läser data ifrån ett exceldokument blir dokumentet skrivskyddat. Andra klienter som eventuellt använder rutiner för att operera mot samma exceldokument vid samma tidpunkt kommer då inte att få åtkomst till exceldokumentet. Då flera klienter använder webbapplikationen samtidigt kan det således finnas risk att vissa klienter kommer att få invänta åtkomst till datakällan.

Första raden i ett exceldokument kan inte läsas av en webbapplikation eftersom raden läses in som kolumnnamn. Idag lagrar GUTAB data enligt en särskild struktur där den första raden inte innehåller kolumnnamn utan består av rådata. Denna struktur måste ändras om GUTAB vill kunna använda exceldokument från VBA-applikationen i webbapplikationen¹.

I en framtida webbapplikation rekommenderas att använda en databas som datakälla, vilket GUTAB har för avsikt att göra i framtiden. Webbapplikationens prestanda blir därmed högre. Dessutom minskar risken för att åtkomstkonflikt uppstår i datakällan.

GUTABs frågeanalysmodul syftar idag delvis till att flera användare ska kunna använda programmet samtidigt. Att använda exceldokument som datakälla kan därför inte betraktas som det optimala alternativet.

¹ GUTAB planerar att använda exceldokument som är skapade i VBA-applikationens administratörsdel i webbapplikationen.

4.7 Val av konverteringsteknologi

Skribbordsundersökningen delades in i två delar eftersom det konstaterades att konvertering av befintlig funktionalitet och implementering av kommande funktionalitet inte kunde genomföras med samma teknologi.

Därmed fattas två beslut, det första redogör för vilken teknologi som är mest lämplig för att konvertera befintlig funktionalitet. Det andra valet redogör för vilken teknologi som är mest lämplig för att implementera kommande funktionalitet.

Val av teknologi för befintlig funktionalitet

Ett seminarium har genomförts tillsammans med GUTAB som syftade till att fastställa den teknologi som är mest lämplig att använda vid en konvertering av företagets befintliga frågeanalysmodul till webbläsaren Internet Explorer.

Seminarier utgick från utvärderingen som gjordes i föregående metodsteg. Aspekter som utvecklingsmiljö, administrering av affärsobjekt, användarvänlighet och utseende diskuterades.

Det finns många fördelar med ASP.NET jämfört med ASP¹. ASP.NET tar webbutvecklingen till nya höjder där VS.net ger programutvecklaren möjlighet att utveckla webbapplikationer som baseras på den händelsestyrda programmeringsmodellen och liknar därmed mer traditionell programutveckling.

ASP.NET medför dels programmeringstekniska förbättringar såsom större enkelhet och bättre säkerhet samt fundamentala förbättringar såsom förbättrat sessionsstöd, lättare administrering av affärsobjekt, kompilerad källkod samt högre prestanda.

Under seminariet konstaterade vi att utvecklingsmiljön VS.net är användarvänlig då alla resurser finns samlade i samma utvecklingsmiljö samt möjliggör för programutvecklaren att enkelt skapa webbapplikationens utseende genom att tillämpa "klicka & dra"-principen.

GUTAB fick möjlighet att studera utvecklingsmiljön VS.net och kom fram till att den påminner om utvecklingsmiljöer som företaget har använt vid utvecklande av VB och VBA applikationer vilket ansågs som positivt.

Det faktum att Microsoft lanserar ASP.NET hårt som framtidens teknologi vid utveckling av interaktiva och dynamiska webbapplikationer är en aspekt som GUTAB har tagit hänsyn till i bedömningen.

¹ Chris Payne, 2002, sid. 23

Till sist ansåg även GUTAB att det är väldigt positivt att programutvecklaren endast behöver spendera energi på uppgifter som berör utvecklingsarbetet. Därigenom blir utvecklings- och förvaltningsarbetet väldigt effektivt och webbapplikationen kan därmed underhållas med små resurser.

Ovanstående fördelar som har nämnts med ASP.NET blev avgörande vid valet av lämplig konverteringsteknologi. Eftersom GUTAB dessutom vill kunna erbjuda en modern webbapplikation som även skall vara attraktiv i framtiden anses ASP.NET vara den teknologi som är mest lämplig att använda vid en eventuell konvertering.

Val av teknologi för kommande funktionalitet

Skrivbordsundersökningens andra del listade för- respektive nackdelar med teknologierna ActiveX samt Java applets som kan användas för att skapa applikationer på Internet som innehåller grafiska manipulerbara objekt.

Baserat på underlaget som skrivbordsundersökningens andra del genererade beslöt GUTAB att ActiveX teknologin är mest lämplig för att implementera kommande funktionalitet i webbapplikationen eftersom ActiveX-programmen är kraftfulla, kan utvecklas med VB, och kan köras på Microsofts plattformar.

5 Slutsats

Arbetet har resulterat i förslag på lämpliga teknologier för att konvertera funktionalitet i GUTABs frågeanalysmodul samt implementera kommande funktionalitet i ett program som kan köras i Internet Explorer.

Ett viktigt konstaterande som undersökningen har medfört är att de teknologier som behandlas skiljer sig åt eftersom de agerar i olika delar av programmeringsmodellen begäran/svars- modellen. ASP.NET som valdes för att konvertera befintlig funktionalitet är en serverbaserad teknologi. ActiveX-program som valdes för att implementera kommande funktionalitet är en klientbaserad teknologi.

Arbetet är således ett förslag på hur GUTABs webbapplikation kan se ut då de väljer att använda en serverbaserad teknologi för att konvertera befintlig funktionalitet och en klientbaserad teknologi för att implementera kommande funktionalitet.

Följaktligen kan GUTABs framtida webbaserade applikation se ut på följande två sätt;

- En webbapplikation som är utvecklad i ASP.NET och som endast består av funktionalitet från GUTABs befintliga frågeanalysmodul.
- En webbapplikation bestående av funktionalitet från GUTABs befintliga frågeanalysmodul samt ett integrerat ActiveX-program som körs hos klienten. Utöver tidigare nämnd funktionalitet tillkommer programlogik för att spara respektive hämta data till respektive från webbservern.

GUTAB kan inte i dagsläget ge besked om vilken webbapplikation som kommer att användas då det finns andra aspekter, vilka ligger utanför ramen för detta arbete, som måste övervägas innan detta kan fastställas.

Däremot vet GUTAB att om de väljer att utveckla en webbaserad applikation så kommer de att konvertera den befintliga frågeanalysmodulen i ett första steg för att sedan i ett eventuellt kommande steg 2 integrera ett ActiveX-program med densamma.

Undersökningen konstaterar även att det är fullt möjligt att konvertera GUTABs frågeanalysmodul till Internet Explorer. All funktionalitet kunde återskapas i webbapplikationerna.

Kompletteringar gjordes dock för att skapa grafiska objekt. Befintliga COM respektive .NET-objekt användes som innehöll funktionalitet för att skapa olika typer av diagram.

6 Diskussion

Efter att ha genomfört det här examensarbetet kan jag konstatera att arbetet till stor del har baserats på GUTABs befintliga frågeanalysmodul. Undersökningen har även genererat ett förslag på lämplig teknologi för kommande funktioner som eventuellt kommer att integreras med webbapplikationen i framtiden. Detta har dock inte konstaterats genom att testa teknologierna.

I undersökningen föreslås en serverbaserad teknologi som mest lämplig för konverteringen. En lämplig fortsättning på det här arbetet är att undersöka möjligheterna att konvertera befintlig funktionalitet från VBA-applikationen till en klientbaserad teknologi.

Baserat på skrivbordsundersökningens andra del beslutade GUTAB att ActiveX teknologin är mest lämplig att använda för att implementera kommande funktionalitet. Det vore därför intressant att veta om det går att konvertera befintlig funktionalitet till ett ActiveX-program samt visa hur ActiveX-programmet kommunicerar med en definierad server genom aktiviteter som "hämta data" och "spara data". Detta skulle medföra att både befintlig- och kommande funktionalitet kan utvecklas i samma program.

Det här examensarbetet fick sin naturliga infallsvinkel baserat på de kriterier som arbetades fram. Kriterierna rangordnades inte varför samtliga kriterier hade samma status i undersökningen. Däremot hade en annan infallsvinkel till problemet kunnat identifieras om kriteriet "Kommande funktioners realiserbarhet" rangordnats till nummer ett. Det hade då blivit naturligt att undersöka möjligheten att utveckla all funktionalitet i ett ActiveX-program.

7 Reflektion

Att genomföra ett examensarbete själv kräver en stor arbetsinsats, mycket tid och tålamod av studenten. Detta kan jag konstatera efter att ha genomfört detta examensarbete på C nivå vid Högskolan Dalarna.

En viktigt steg när man gör ett examensarbete är att göra en lämplig avgränsning anpassad till kursens innehåll. Jag måste tyvärr medge att jag misslyckades med det. Jag har kommit fram till att jag inte spenderade tillräckligt mycket tid för att avgränsa arbetet.

Min inställning till det definierade uppdraget var att det skulle gå att genomföra, jag reflekterade aldrig över om arbetet var för stort. Jag började arbeta mig in i uppdraget och insåg att uppdragsgivaren hade gett mig ett intressant uppdrag.

Jag tror att det hade varit mer rimligt om två studenter hade genomfört uppdraget tillsammans. Idag är jag nöjd med att jag har genomfört mitt examensarbete själv, däremot tror jag att resultatet kunde ha blivit bättre om två studenter hade gjort det tillsammans.

I början gick arbetet väldigt långsamt framåt, jag visste inte vad eller hur jag skulle arbeta. Steg 1 blev därför att skapa en metod som jag anpassade till uppdraget. Metoden syftade till att arbeta fram ett antal kriterier tillsammans med mina handledare på högskolan och GUTAB som skulle spegla konverteringsarbetet.

Kriterierna arbetades fram och konverteringen av utvald funktionalitet kunde startas. En mycket stor del av den tid som jag lade ner på arbetet spenderades på konverteringen. Från början var det inte tänkt att en konvertering skulle ingå i arbetet och det diskuterades huruvida teknologierna kunde utvärderas i teorin kontra praktiken.

Uppdragsgivaren menade att utvärderingen kunde genomföras i teorin medan högskolan menade att det var svårt att bevisa hur lämplig en teknologi är för ett visst ändamål utan att testa teknologins kapacitet i praktiken.

Uppdragsgivaren hade skrivit ner en kortfattad uppdragsdefinition ifrån början. Efter det diskuterade vi igenom uppdragsdefinitionen varefter vi tillsammans gjorde en tidsplan. Detta upplägg ändrades av handledaren när arbetet startade utan att uppdragsgivaren och handledaren tillsammans diskuterade det nya upplägget. Jag ställde mig på skolans sida och baserade min metod på att utföra ett test av två teknologier som kunde vara lämpliga för en eventuell konvertering.

En stor orsak till att detta hände kan ha varit att kommunikationen mellan uppdragsgivaren och högskolan var bristfällig. Skolan försvarade sig genom att säga att examensarbetet till stor del ska baseras på självständigt ansvarstagande av

studenten. Detta var för mig klart, men jag tror faktiskt att det hade blivit mycket lättare om samtliga inblandade parter från och med arbetets start hade definierat uppdraget tillsammans.

Eftersom uppdraget var unikt så visste jag inte hur jag skulle gå till väga för att lösa uppgiften. Det hade därför känts rätt om handledaren ifrån högskolan hade varit med så att han från början kunde ha styrt uppdraget, från skolans sida sett. Dessutom hade handledaren då haft en klar bild över vad uppdraget egentligen gick ut på.

Risken finns ju faktiskt att jag tolkade min uppdragsgivare fel vilket kanske gav min handledare intrycket av att styra mig åt fel håll när jag förklarade uppdraget för honom, jag säger dock inte att det är så! Min uppfattning är dessutom att uppdragsgivaren skulle kunna vinna mycket på att träffas i ett tidigt skede för att göra dem själva medvetna om vad de egentligen vill ha ut av uppdraget.

Det första som borde göras i ett examensarbete är därför att definiera uppdraget mer i detalj. Det skulle även vara lämpligt att ge ett förslag till arbetets uppläggning. När detta förslag har arbetats fram bör högskolans handledare ge synpunkter varefter uppdragsgivaren får möjlighet att ta ställning till detta. Om inte detta görs så finns det en stor risk att arbetets resultat riskeras.

Min uppdragsgivare GUTAB anser att handledaren inte ska få tillåta sig att styra examensarbetaren till att ändra i uppdragets formulering och innehåll, utan att först komma överens med uppdragsgivaren om det. Jag håller med.

Det bör tilläggas att det hela gick bra till sist. Mycket arbete hade dock kunnat sparas ifall dessa problem inte hade uppstått.

Kontentan blir därmed att det är väldigt viktigt att alla tre parter tillsammans definierar uppdraget samt att en bra kommunikation mellan dessa parter upprättas för att undvika missförstånd. Då finns rätt förutsättningar för att generera ett bra resultat utifrån den definierade uppdragsbeskrivningen.

8 Ordlista

.NET Framework – En uppsättning programutvecklingsobjekt och mallar ifrån Microsoft. Står för de bakomliggande funktionerna i ASP.NET

API – Står för Application Programming Interface. Ett API är ett klassbibliotek innehållande objekt och rutiner.

Array – Är en datatyp som kan användas i både ASP och ASP.NET. Motsvaras av en flerdimensionell datastruktur.

ASPX-sida – Sidan som presenterar utseendet i en ASP.NET-applikation.(motsvarigheten till HTML-delen på en ASP-sida)

COM-objekt – Avser i detta arbete en komponent utvecklad i VB6 bestående av programlogik. COM-objektet initieras på en ASP-sida som innehåller programlogik för att anropa rutiner i objektet.

Cookies – En textfil som lagras på klientens dator innehållande information om klienten.

Datatable – Är ett objekt i ASP.NET som motsvaras av en postuppsättning med data med ett antal kolumner och rader.

DLL – Ett exekverat och kompilerat program som har länkats ihop till en komponent. Används ofta i webbapplikationer som externa objekt med utvidgad funktionalitet.

Grafiska objekt – Med grafiska objekt menas färdigutvecklade COM-objekt som består av funktionalitet för att generera olika typer av resultatdiagram. Exempel på diagram är stapel-, cirkel- och linjediagram.

Grafiska manipulerbara objekt – Funktionalitet som möjliggör för användaren att på Internet manipulera grafiska objekt (såsom exempelvis en stapel i ett gantt-diagram) enligt den så kallade ”klicka & dra” principen.

Recordset – Är ett objekt i ASP (VB) som motsvaras av en postuppsättning med data.

Runtime – Benämning på ett program som exekverar.

9 Källförteckning

Följande kapitel redogör för de källor som har använts i arbetet. Följande publicerade källor och Internetkällor används:

Publicerade källor:

Chris Payne, Lär dig ASP.NET på 3 veckor, pagina, Göteborg, 2002

Daniel Y. Liang, Introduction to Java Programming 3rd edition, New Jersey, 2001

Scott Mitchell m.fl., Lär dig Active Server Pages 3.0 på 3 veckor, pagina, Göteborg, 2000

Internetkällor:

Eric J. Grossman, "ActiveX:Security Issues", 2000,
<<http://www.saic.com/healthcare/hipaa/activex.pdf>>, 2003-04-10

John Lim, "7 Reasons Why PHP is Better than ASP", 2000-09-14,
<<http://php.weblogs.com/php ASP 7 reasons>>, 2003-06-16

Merlin Hughes, "Javabeans and ActiveX goes head to head", March 1997,
<<http://www.javaworld.com/javaworld/jw-03-1997/jw-03-avb-tech-p3.html>>,
2003-05-14

Microsoft Corp., "What is ActiveX?", <<http://www.active-x.com/articles/whatis.htm>>, 2003-05-15

Microsoft Corp., "What Is Visual Basic for Applications?", 2001-01-30,
<http://msdn.microsoft.com/vba/prodinfo/backgrounder.ASP#whatis>, 2003-04-19

Gbdirect, "Active Web Sites and Comparison of Scripting Languages",
<http://training.gbdirect.co.uk/courses/perl/comparison_php_versus_perl_vs ASP_jsp_vs_vbscript_web_scripting.html>, 2003-04-23

Harish Kamath, "PHP and COM", 2002-04-23,
<http://www.devshed.com/Server_Side/PHP/PHPandCOM/page3.html>, 2003-05-01

Sun Microsystems, "Comparing Java Server Pages and Microsoft Active Server Pages Technologies", <<http://java.sun.com/products/jsp/jsp-ASP.html>>, 2003-04-25