



Ontwikkeling van een Software Agent

Naam : P.M.S. Boekholt
Studentnummer : 99006658
Studierichting : Vormgeving en Ontwerp van Interactie (VIA)
Blok : 2004-1.1
Examinatoren : P.B van der Sluijs, C.H.T. Tan
Afstudeerbedrijf : i-Concept InternetWorks BV
Bedrijfsmentor : C.A. Hall

Referaat

Een afstudeerverslag van Philip Boekholt, studerende aan de Haagse Hogeschool, opleiding Vormgeving en Ontwerp van Interactie, te Den Haag.

Descriptoren:

- Afstuderen
- Vormgeving en Ontwerp van Interactie (VIA)
- Coldfusion
- i-Concept InternetWorks BV
- MySQL
- Software Agent

Voorwoord

Voor u ligt het afstudeerverslag van Philip Boekholt met betrekking tot de ontwikkeling van een Software Agent. Deze opdracht is uitgevoerd in het kader van het afstuderen voor de Haagse Hogeschool, studie Vormgeving en Ontwerp van Interactie.

Graag wil ik van de gelegenheid gebruik maken om enkele mensen te bedanken die de afstudeerperiode een belangrijke rol hebben gespeeld. Ik wil alle werknemers van I-Concept bedanken voor de medewerking die zij mij gaven tijdens het uitvoeren van de opdracht en dan met name mijn bedrijfsmentor, Chris Hall voor de begeleiding.

Verder bedank ik mijn examinatoren, Pieter van der Sluijs en Cecilia Tan voor hun feedback op mijn verslag en advies bij het bedrijfsbezoek.

Philip Boekholt
Rotterdam, Juni 2004

Inhoudsopgave

1. Inleiding.....	6
2. Het bedrijf.....	7
2.1 Algemene organisatie omschrijving.....	7
2.2 Geschiedenis	7
2.3 Organisatiestructuur i-Concept Internetworks	8
2.4 Doelstelling	8
3. De opdracht.....	10
3.1 Probleemstelling	10
3.2 Doelstelling	10
4. Plan van aanpak.....	11
4.1 Activiteiten	11
4.2 De planning.....	12
4.3 Op te leveren producten	12
4.4 Benodigde software.....	13
4.5 Methoden en technieken.....	13
5. Definitiestudie	14
5.1 Inventariseren en analyseren van de huidige situatie	14
5.2 Inventariseren van de eisen.....	15
5.3 Inventarisatie van mogelijk nieuwe DBMS pakketten.....	17
5.4 Noodzaak van een mogelijk nieuw besturingssysteem.....	18
5.5 Het vaststellen van de ontwikkelomgeving	18
5.6 Het systeemconcept.....	19
5.6. Pilotplan.....	28
6. Pilotontwikkeling	29
6.1 Pilot server.....	29
6.2 Pilot database	42
6.3 Pilot software agent.....	64
7. Procesevaluatie	83
8. Productevaluatie.....	85
8.1 Plan van aanpak.....	85
8.2 Definitiestudie Software Agent	85
8.3 Testplan/Testrapport	85
8.4 De server	85
8.5 Software Agent	85
8.7 CMS Software Agent.....	86
Literatuur- en internetverwijzingen	
Bijlagen	

1. Inleiding

Dit afstudeerverslag doet rapport van het uitgevoerde afstudeerproject met een omvang van 16 weken, dat Philip Boekholt bij stagebedrijf I-Concept InternetWorks BV heeft volbracht. De opdracht is een verplicht en afsluitend onderdeel van de opleiding Vormgeving en Ontwerp van Interactie aan de sector informatica van de Haagse Hogeschool.

Het maken van dit verslag heeft de volgende beoogde doelen:

- Evalueren of de gemaakte keuzes in het proces weloverwogen waren;
- Nagaan hoe de student is omgegaan de problemen die hij is tegengekomen.

Dit verslag is in eerste instantie gericht aan mijn examinatoren dhr. P.B van der Sluijs en mw. C.H.T. Tan en een bedrijfsgecommitteerde, die allen tijdens de examenzitting aanwezig zullen zijn. Daarnaast heb ik dit verslag geschreven voor mijn bedrijfsmentor Chris Hall die aan de hand van dit verslag inzicht kan krijgen in hoe ik de afstudeerperiode bij I-Concept heb ervaren.

De inhoud bestaat uit de hoofdstukken die het verloop van het proces beschrijven en een bronvermelding van de literatuur die ik tijdens het afstuderen heb gebruikt. In de externe bijlagen heb ik delen opgenomen van de producten die ik tijdens het afstuderen heb ontwikkeld.

In het tweede hoofdstuk beschrijf ik als eerste de organisatie rond het project om een beeld te scheppen van het bedrijf en het systeem waar het om ging binnen het project. In het derde hoofdstuk beschrijf ik wat de opdracht is. Hoofdstuk vier beschrijft de aanpak van het project met de toegepaste methoden. Hoofdstuk vijf beschrijft de voorbereiding die ik heb getroffen voordat ik kon gaan ontwikkelen. Hoofdstuk zes zal in het teken staan van het ontwikkelproces. Tot slot in hoofdstuk zeven en acht evalueer ik de afstudeerperiode door te kijken naar hoe het proces gegaan is en hoe de kwaliteit van de producten is geworden.

2. Het bedrijf

2.1 Algemene organisatie omschrijving

i-Concept levert Internet consultancy, projectmanagement en conceptontwikkeling op het gebied van E-business, E-HRM en E-learning. De onderneming ontstond in 1997 door de verzelfstandiging van de internetdivisie van de Europese modemfabrikant en BBS Pionier Bausch Datacom. In 2000 werd de organisatie uitgebreid met e-HRM specialisten van Arbeidsmarkt.com. Medio oktober 2001 sloot het designbureau RoundSquare zich aan bij i-Concept, waardoor i-Concept een stevige eigen designafdeling voor projecten en eigen producties erbij kreeg.

i-Concept werkt voor organisaties als Philips, Arcadis, Nutsbedrijf Regio Eindhoven, Gemeente Arnhem, Gemeente Tilburg, Ministeries van Economische Zaken en OC&W, Kennisnet, Stichting Bedrijfstakregelingen Dakbedekkingsbranche, Universiteitsverenigingen EFV en SVIIB.

2.2 Geschiedenis

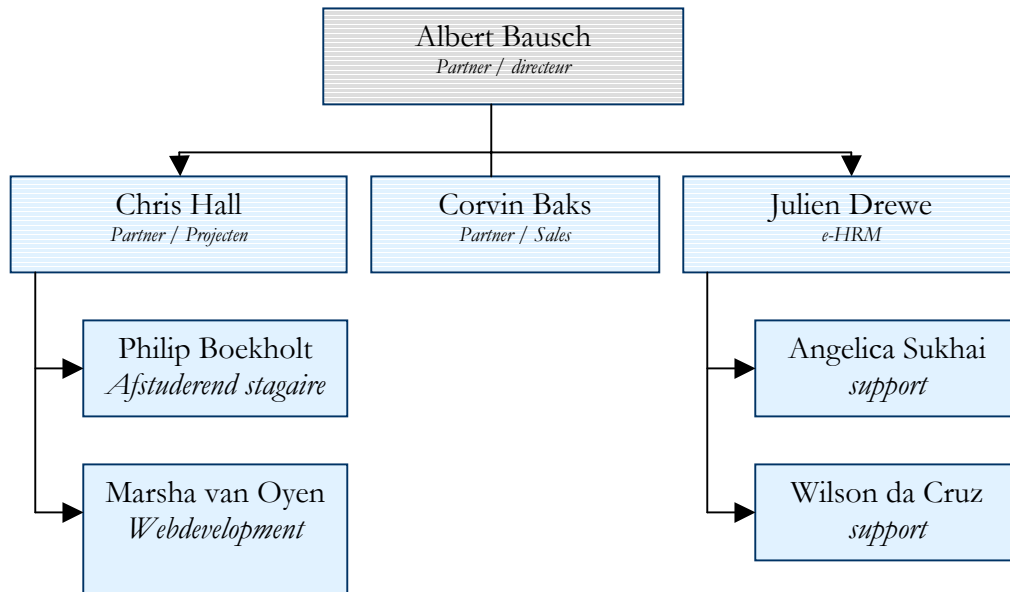
i-Concept startte in 1997 als spin-off van de Europese modemfabrikant en BBS pionier Bausch Datacom.

Begin jaren '90 waren moeilijke tijden voor het bedrijfsleven en de ICT branche in het bijzonder. Bausch Datacom wist in deze periode een leiderspositie op te bouwen in de modemmarkt. Het bedrijf verwierf vooral bekendheid door het succesvolle partnerschap met Postbank Girotel (hét Bausch-modem), haar pioniersrol als leverancier van Internet Service Providers en de inkomstenbelastingaangifte via modem voor de Belastingdienst. Anders dan bijvoorbeeld concurrent PTT, richtte Bausch Datacom zich van meet af aan op gebruikers van online toepassingen. Dit hield bijvoorbeeld in: zo veel mogelijk offline en niet zoals bij Videotext zoveel mogelijk online.

Vanaf 1994 kwamen ICT ontwikkelingen in een stroomversnelling. De steeds sneller veranderende wereld vergde een even zo groot aanpassingsvermogen binnen alle gelederen van het bedrijfsleven. Een effectieve toepassing van informatie- en communicatietechnologie (ICT) gold steeds meer als de kritieke succesfactor. Tijdige en betrouwbare informatie, efficiënte vormen van communicatie en uitgebalanceerd kennismanagement bleken de onmisbare onderdelen van een soepel draaiende organisatie. De doorbraak van het internet en de daaraan gerelateerde technologieën lagen hieraan uiteraard ten grondslag. Internettechnologie werd wereldwijd omarmd als defacto standaard en wijdverspreid geïntegreerd in tal van zakelijke en industriële toepassingen. Deze brede opmars van deze standaardtechnologie had als belangrijk positief effect dat de toekomstgerichte ondernemers sneller en goedkoper kon beschikken over bedrijfsspecifieke software en de afhankelijkheid van dure exotische programmatuur tot een minimum beperkt kon houden. Deze ontwikkeling lag precies in de lijn van de visie van de oprichters van i-Concept.

In 1999 nam Alex Bausch afscheid van i-Concept om het verzelfstandigde adviseurnetwerk Lift-off uit te bouwen. Albert Bausch zette i-Concept voort en samen met de huidige partners werd i-Concept succesvol als de intranet en e-HRM expert gepositioneerd.

2.3 Organisatiestructuur i-Concept Internetworks



Figuur 1 - Organisatiestructuur

2.4 Doelstelling

2.4.1 Korte termijn

De onderneming heeft op korte termijn de doelstelling om de efficiëntie verder te verbeteren door enerzijds meer te doen voor dezelfde klanten (diversificatie) en anderzijds door het technologisch raamwerk te verfijnen. Door een verdere structurering van ontwikkelmethodes en modulebeheer, worden de mogelijkheden bevorderd om kwalitatief hoogwaardige maatwerkoplossingen bij nieuwe en bestaande opdrachtgevers te implementeren. Het verhogen van efficiëntie is ook zichtbaar in de samenwerking tussen de verschillende afdelingen binnen i-Concept. Door de optimalisatie van bedrijfsprocessen, eventueel met gebruik van ICT toepassingen, zal het rendement van de huidige activiteiten toenemen. Hiernaast zal de mogelijkheid om nieuwe diensten te ontwikkelen toenemen.

2.4.2 Lange termijn

Op lange termijn wil i-Concept zich meer concentreren op het oplossen van die organisatievraagstukken, waar ICT toepassingen een sleutelrol spelen. Daarbij wordt in de personele expansie vooral gezocht naar concept- en projectontwikkelaars en het operationele

ontwikkelwerk in toenemende mate uitbesteed aan de leden van het eigen ework.nl e-business netwerk. Door het verhogen van de efficiëntie binnen de huidige activiteiten en het stimuleren van de samenwerking tussen de verschillende afdeling zal de waarde van de totale dienst voor de klanten toenemen.

3. De opdracht

3.1 Probleemstelling

In de huidige situatie heeft i-Concept een semi-automatische robot die het internet kan afscannen op gecommuniceerde kennis en deze kennis kan indexeren in een kennisdatabase. Op deze manier kan deze kennis als naslagwerk hergebruikt worden. Deze robot, of beter gezegd software agent, moet in de huidige situatie nog wel bediend en aangestuurd worden door een werknemer.

Zoals het er nu naar uit ziet, is de capaciteit van de huidige databaseomgeving bijna bereikt. Het huidige DBMS is namelijk MS Acces. Hierdoor is het haast onmogelijk om een nieuwe module, zoals de software agent, toe te voegen bij de huidige internet modules en tegelijkertijd de performance van het dataverkeer optimaal te houden.

Een ander probleem is dat de huidige software agent de uitvoering van de bestaande processen en procedures verstoort, omdat deze “agent” nog altijd aangestuurd moet worden door medewerkers.

3.2 Doelstelling

Het doel van de opdracht is het ontwerpen en ontwikkelen van een volledig automatische software agent die gecommuniceerde kennis of informatie indexeert in een kennisdatabase. Deze kennisdatabase moet op een gemakkelijke wijze toegankelijk zijn voor de medewerkers. Er zal dan ook op applicatieniveau een user interface gerealiseerd moeten worden om dit doel te kunnen verwezenlijken.

Daarnaast moet er aan de module software agent extra functionaliteit toegevoegd worden, zoals het beheren van de opgeslagen kennis en het gemakkelijk kunnen uitwisselen van kennis. Tevens moet de module universeel zijn, zodat het makkelijk toegepast kan worden op de reeds bestaande modules.

Om de software agent te kunnen verwezenlijken zal er ook gekeken moeten worden naar het capaciteitsprobleem op het gebied van DBMS. Er zal een nieuwe DBMS moeten worden aanbevolen en moeten worden toegepast.

4. Plan van aanpak

4.1 Activiteiten

De activiteiten zijn onderverdeeld in een aantal fasen van het project.

In de definitiestudie komen de volgende activiteiten voor:

- Het opzetten van een plan van aanpak voor het opstellen van een definitiestudie.
- Inventariseren van de wensen en eisen van de opdrachtgever met betrekking tot de module “software agent”.
- Inventariseren van DBMS-en en (eventueel) nieuwe operatingsystemen.
- In combinatie met het vorige punt zal er ook een inventarisatie gemaakt moeten worden van mogelijke ontwikkelomgevingen waarmee de nieuwe module gemaakt kan worden.
- Een systeemconcept opstellen.
- Een moduleconcept opstellen.
- Een pilotplan opstellen.
- De resultaten verwerken in de definitiestudie.

In de pilotontwikkelfase komen de volgende activiteiten voor:

- Een pilotstrategie bepalen.
- De resultaten uit de definitiestudie gebruiken om pilotontwikkelfase op te zetten en per onderzoek kijken of er nog verfijning nodig is.
- Pilots opzetten.
- Per pilot pilotdelen opzetten (indien nodig).
- Lay-outs opstellen voor de informatieve schermen van de “software agent”.
- Een guide voor de functionaliteiten van de “software agent” opstellen.
- Per pilot onderzoek doen naar de specificaties.
- De pilots ontwikkelen.
- De pilots testen.
- White Box Testing.
- Documentatie schrijven over de ontwikkelde pilots.

In de invoeringsfase komen de volgende activiteiten voor:

- Verifiëren en valideren van de ontwikkelde pilots.
- Black Box Testing.
- Het implementeren van de pilots.
- Evaluatierapport schrijven over de werking van de module “software agent”.
- Documentatie schrijven over de eventuele bugs in de module “software agent”.
- De pilotontwikkelfase itereren om bugs op te lossen (indien nodig).

4.2 De planning

Week	Activiteit	Product
Startweek (2-6 feb)	Opstellen Plan van aanpak	plan van aanpak
2 (9-13 feb)	Opstellen definitiestudie	definitiestudie
3 (16-20 feb)	Opstellen pilotontwikkelpplan	pilotontwikkelpplan
4 (23-27 feb)	Ontwikkelen pilot(delen) 1	
5 (1-5 maart)	Ontwikkelen pilot(delen) 1	
6 (8-12 maart)	Ontwikkelen pilot(delen) 1	Pilot 1
7 (15-19 maart)	Invoering pilot 1, Ontwikkelen pilot(delen) 2	
8 (22-26 maart)	Ontwikkelen pilot(delen) 2	
9 (29 maart – 2 april)	Ontwikkelen pilot(delen) 2	
10 (5-9 april)	Ontwikkelen pilot(delen) 2	Pilot 2
11 (12-16 april)	Testen pilot 2, Ontwikkelen pilot(delen) 3	
12 (19-23 april)	Invoering pilot 2, Ontwikkelen pilot(delen) 3	
13 (26-30 april)	Ontwikkelen pilot(delen) 3	
14 (3–7 mei)	Ontwikkelen pilot(delen) 3	
15 (10-14 mei)	Ontwikkelen pilot(delen) 3	Pilot 3
16 (17-21 mei)	Invoering pilot 3, Schrijven bouwplan	
17 (24-28 mei)	Schrijven verslag	
18 (31mei – 4juni)	Schrijven verslag	

4.3 Op te leveren producten

De volgende producten zullen worden opgeleverd:

- Plan van aanpak.
- Definitiestudie.
- Ontwikkelpplan.
- Testplan/testrapport.
- Een werkende geautomatiseerde “software agent” module.
- Een content management systeem voor de “software agent” module.
- Systeemdokumentatie over de module “software agent”.
- quick reference card over de module “software agent”.

4.4 Benodigde software

- Windows (98, NT, 2000) / Linux
- Macromedia Coldfusion
- Macromedia Coldfusion server
- Microsoft office
- Photoshop 7
- mySQL server
- MS SQL server 2000
- MS Windows 2000 server

4.5 Methoden en technieken

4.5.1 Methoden

Het hele project zal ingedeeld worden volgens de fasering van de methode IAD. Voor alle drie de opgenoemde fasen zullen alleen de benodigde activiteiten uit de methode IAD gebruikt gaan worden.

4.5.2 Technieken

Tijdens het project zal er gebruik gemaakt gaan worden van de volgende technieken:

- Systematisch bronnenonderzoek
- Interviewtechnieken
- State Diagram (GUI design)
- Window hiërarchie diagram (GUI design)
- Taak diagram (GUI design)
- Conceptueel gegevensmodel ERD
- Relationeel model
- Implementatie model (DBMS)
- White Box Testing (voor ontwikkelen)
- Black Box Testing (voor acceptatietest)

5. Definitiestudie

In deze fase heb ik me bezig gehouden met het inventariseren van de systeemeisen en het uitwerken van die systeemeisen in een systeemconcept. In dit hoofdstuk zal ik uitleggen op welke wijze ik de systeemeisen heb geïnventariseerd en welke overwegingen ik heb gedaan en welke beslissingen heb genomen met betrekking tot het systeemconcept.

5.1 Inventariseren en analyseren van de huidige situatie

5.1.1 Doelgroepanalyse

Ik vond het niet nodig om een doelgroepanalyse te doen, omdat de software agent weinig tot geen interactie behoeft met een medewerker.

5.1.2 De website

De website heb ik op webpagina's en codering geïnventariseerd en geanalyseerd. Dit heb ik gedaan met behulp van een tabel. In deze tabel heb ik alle pagina's van de website opgenomen. Per pagina heb ik gezocht naar links naar andere pagina's. Door op de website naar deze links te navigeren en weer terug ben ik erachter gekomen of de pagina wel of niet werd gebruikt. Dit heb ik in de tabel vermeld. Pagina's met achter- of voorvoegsels als "backup", "BU" of iets wat daar op lijkt, heb niet opgenomen in de tabel.

5.1.3 De database

De website, www.arbeidsmarkt.com, is database gestuurd. Om inzicht te krijgen in de database, heb ik een inventarisatie gemaakt. Er waren geen documenten en modellen aanwezig van de database. Ik heb daarom bij de inventarisatie ook gelijk gekeken uit welke tabellen, tupels, relaties en restricties de database bestond. Bij de inventarisatie merkte ik op dat sommige tabellen geen inhoud hadden. Voor deze tabellen nam ik aan dat ze niet meer werden gebruikt. Dit ben ik nagegaan met behulp van Macromedia Dreamweaver MX. Met dit programma kan je in de webpagina's zoeken naar tekst. Ik heb toen in de webpagina's gezocht naar namen van de lege tabellen. Indien ik een tabelnaam niet voor kwam in de webpagina's, kon ik eruit concluderen dat de tabel niet meer werd gebruikt. Ik heb toen een implementatiemodel gemaakt van allen de tabellen die wel werden gebruikt. Dit implementatie model zal ik namelijk nodig hebben bij de overzetting van MS Acces naar MySQL. Bij die overzetting wilde er zeker van zijn dat ik geen ongebruikte tabellen heb overgenomen.

Bij de inventarisatie van de data die in de MS Acces database was opgeslagen, werd mij al snel duidelijk dat de huidige database de situatie arbeidsmarkt.com nauwelijks nog aan kon. De tabel cv bevatte meer dan 26000 records. Ik heb toen gecontroleerd of de site daadwerkelijk traag was. Ik heb me toen aangemeld als recruiter en heb toen vervolgens een zoekactie gedaan naar cv's van mensen uit de ICT branche. Ik moest bij de zoekactie meer dan 2 minuten wachten voordat er eindelijk een resultaten scherm werd vertoond. Hiermee heb ik bevestigd gezien dat de database aan zijn overcapaciteit zit.

5.2 Inventariseren van de eisen

Om de systeemeisen vast te kunnen stellen heb ik eerst een oriënterend gesprek gehad met mijn bedrijfsmentor. In dit gesprek kwam in grote lijnen de bedoeling van de software agent naar voren. Van deze informatie heb ik aantekeningen gemaakt en erover nagedacht over hoe ik de software agent concreet zou kunnen maken met mijn kennis van programmeren.

Daarna had ik ook met de opdrachtgever een gesprek gehad over de eisen waaraan de robot moest voldoen. Ik heb dat gesprek meer als een interview gevoerd, omdat ik steeds ging door vragen. Een echte voorbereiding voor dit interview had ik kunnen doen, omdat ik omdat ik geen goede gesloten vragen kon bedenken. Ik kon namelijk niet alleen maar gesloten vragen stellen over hoe de software agent op technologisch opzicht moest gaan opereren, omdat ik zelf helemaal onbekend was op dat gebied. Wel heb ik gesloten vragen gebruikt om de informatie die hij mij vertelde bevestigd te krijgen. Wel kon ik navragen met welke technologieën zij denken de software agent gemaakt kon worden. Echter, het bleef over het algemeen bij open vragen, omdat ikzelf hier niet dieper op kon inhaken. Het gesprek was van mijn kant non-directief, omdat ik zijn gedachtestroom met betrekking tot de agent zoveel mogelijk boven water wilde krijgen. Ik heb hem in feite zoveel mogelijk laten praten. De functionaliteit van de software agent was mij wel al duidelijk geworden tijdens het eerste gesprek met mijn bedrijfsmentor. Ik heb hier wel nog een aantal gerichte vragen kunnen stellen op het gebied van interactie met de medewerkers en hoe het moest gaan opereren op het internet. Deze vragen had ik puur gesteld ter bevestiging van wat mijn bedrijfsmentor mij had verteld, zodat ik er zeker van wist dat er hier geen onduidelijkheden over konden bestaan.

De richtlijnen die zij mij hebben gegeven op het gebied van de technologische invulling, ben ik verder gaan onderzoeken. Dit waren richtlijnen met betrekking tot programmeertalen als XML en SOAP. Binnen het bedrijf had niemand echt kennis van programmeertalen of technologieën waarmee je een software agent mee zou kunnen maken. Mijn bron van informatie over software agents was daarom voornamelijk het Internet. Ik heb op het Internet zoveel mogelijk informatie proberen te vinden betreffende technologieën waarmee ik de software agent mee zou kunnen bouwen.

Verder had ik besloten om geen workshops te houden over de wensen en eisen van de software agent. De tijd die ik hierin moest insteken zou niet veel meer informatie opleveren dan het gesprek dat ik voorheen al had gehad. Ik vond het daarom niet de moeite waard om nog een workshop te houden. Bovendien zouden dan belangrijke bedrijfsprocessen stil komen te staan, omdat de bezetting minimaal was.

In onderstaand overzicht worden de gestelde eisen opgenoemd die in het interview naar voren zijn gekomen. Ik heb de eisen ingedeeld naar categorie.

Functionele eisen

- De software agent moet vacatures van het Internet kunnen opslaan in een formaat dat gebruikt kan worden op arbeidsmarkt.com
- De software agent moet onderscheid kunnen maken tussen goede en slechte vacatures. Met een goede vacature wordt bedoeld dat een vacature geschikt is om op arbeidsmarkt.com te plaatsen.
- De software agent moet niet al te moeilijk te configureren zijn
- Gevonden vacatures moeten opgeslagen kunnen worden in de database
- De gevonden vacatures moeten snel benaderbaar zijn voor de medewerkers
- Plaatsing van gevonden vacatures door de software agent moet niet moeilijk zijn voor medewerkers
- Er moet automatisch back-ups gemaakt kunnen worden van de vacatures
- Er moet automatisch back-ups gemaakt kunnen worden van de database

Interface eisen

- Het configuratiescherm van de agent moet overzichtelijk zijn
- Het configuratiescherm van de agent moet gemakkelijk te bedienen zijn
- De software agent moet qua kleurgebruik passen binnen de huidige intranet
- De schermindelingen en structuur van het configuratiescherm moeten bekend voorkomen bij de gebruiker

Performance eisen

- De software agent mag de server niet te zwaar belasten. Het cpu verbruik mag niet langer zijn dan 10 seconden.
- De software agent mag andere websites niet negatief beïnvloeden qua performance
- De data die de software agent gevonden heeft, moet snel uit de database gehaald kunnen worden. De database moet een snellere dataoverdracht hebben dan MS Acces. De overdracht mag niet langer dan 5 seconden duren.
- De database moet een grotere opslagcapaciteit aan kunnen zonder dat daarbij de response tijd exponentieel achteruit gaat.
- De server waarop de database draait moet optimaal geconfigureerd zijn voor grote bezoekersaantallen. De response tijd mag niet exponentieel achteruit gaan.

Operationele eisen

- De software agent dient herbruikbaar te zijn voor andere zoekacties

Beveiligingseisen

- De database dient beveiligd te zijn en mag alleen via interne kanalen gemodificeerd worden zijn
- De server dient beveiligd te zijn tegen buitenstaanders en virussen
- De software agent dient beveiligd te zijn en mag alleen door eigen medewerkers aangestuurd en gemodificeerd worden

5.3 Inventarisatie van mogelijk nieuwe DBMS pakketten

Bij het inventariseren van eventueel nieuw te gebruiken RDBMS pakketten, heb ik het Internet geraadpleegd. In de eerste plaats was ik op zoek gegaan naar recensies van webmasters en databasebeheerders en forums over MySQL en MS SQL 7/2000 om erachter te komen welk database pakket de grootste voorkeur kreeg. Deze twee database pakketten kwamen namelijk in het tweede gesprek met de opdrachtgever ter sprake.

Nadat ik de mogelijkheden, de sterke en zwakke punten van zowel MySQL als MS SQL 7/2000 had geïnventariseerd uit de recensies en forums, moest ik advies uitbrengen welk pakket het beste geschikt is voor de software agent. Hieronder ziet u de belangrijkste punten ter vergelijking en welke oordelen MySQL en MS SQL hebben gekregen.

	MySQL	MS SQL
Schaalbaarheid	<i>Voldoende/goed</i>	<i>Uitstekend</i>
Snelheid	<i>Uitstekend</i>	<i>Goed</i>
Interne check op ACID	<i>Nee, voert geen extra queries uit om de data integriteit te controleren</i>	<i>Ja, voert extra queries uit wanneer er data wordt gewijzigd of verwijderd</i>
Configureerbaar	<i>Goed en gemakkelijk</i>	<i>Goed en gemakkelijk</i>
Stabiliteit	<i>Uitstekend</i>	<i>Goed</i>
Conform ANSI-99 SQL	<i>Ja</i>	<i>Ja</i>
Afwijkingen met betrekking tot SQL	<i>Kan geen subqueries uitvoeren</i>	<i>Kan wel subqueries uitvoeren</i>
Behandeling van datum	<i>Datum kan alleen in Amerikaans notatie (yyyy/mm/dd) ingevoerd worden</i>	<i>Kan meerdere datumnotaties aan</i>
Besturingssysteem	<i>Linux / Windows</i>	<i>Windows</i>
Prijs	<i>Gratis</i>	

Uit het overzicht valt duidelijk uit af te leiden, dat MS SQL 7/2000 op een aantal punten beter voldoet dan MySQL. Ik heb uiteindelijk toch besloten om MySQL als nieuwe database te gaan gebruiken voor de software agent en arbeidsmarkt.com. Ik heb hiervoor gekozen, omdat in de eerste plaats de snelheid van het ophalen van data uit de database van cruciaal belang was. Ten tweede moest de database server stabiel zijn. i-Concept kon het zich niet veroorloven dat arbeidsmarkt.com offline ging, omdat ze dan potentiële klanten konden mislopen en bestaande klanten konden verliezen. Als derde punt die ik had meegenomen in mijn besluit was de prijs. Een gratis pakket dat bijna net zo goed is als een vooraanstaand DBMS pakket, is financieel gezien natuurlijk veel gunstiger.

De punten waarbij MySQL minder scoort dan MS SQL of ontbreken zouden voor de huidige website voor eventuele problemen kunnen zorgen. Ik ben nagegaan of er in de webpagina's gebruik werd gemaakt van subqueries die niet worden ondersteund door MySQL. Ik kwam erachter dat er een aantal subqueries waren gebruikt die herschreven moesten worden. Ik had toen besloten om de inventarisatie van de website uit te breiden met extra informatie op het gebied van queries en subqueries, zodat ik later gemakkelijker kan terugzien welke pagina's eventueel wel of niet herschreven moesten worden.

Ik heb ook nog gekeken of er bij pagina's met queries met een 'delete'-statement of een 'update'-statement, gerelateerde tabellen worden gewijzigd. Ik heb hiernaar gekeken, omdat ik bij mijn eerdere vergelijking tussen MySQL en MS SQL 2000 erachter kwam dat MySQL geen interne controle uitvoert op het gebied van data-integriteit. MS Acces doet dat wel, dus het was voor mij zaak om na te gaan of er in de webpagina's zelf al een controle wordt uitgevoerd hierop of niet.

Een ander punt waarop MySQL verschilt met MS Acces is dat de datum in MySQL altijd in Amerikaanse notatie moet worden ingevoerd: eerst het jaar, dan de maand en daarna de dag. Ook dit ben ik nagegaan bij de webpagina's. Bij deze inventarisatie ben ik erachter gekomen dat er aardig wat pagina's met een datum werken. Deze zullen allemaal veranderd moeten worden.

5.4 Noodzaak van een mogelijk nieuw besturingssysteem

Voor de inventarisatie van een eventueel nieuw te gebruiken besturingssysteem, heb ik eerst geïnventariseerd hoe ons eigen netwerk is opgebouwd. Zodoende kon ik beter inzicht krijgen en beter inschatten wat de beste keuze van besturingssysteem is voor de nieuwe server. Bij i-Concept draait het netwerk op een Windows 2000 server en zijn de werkstations ook Windows 2000. De “dedicated server” in Amsterdam is een Windows NT server. Hier kon ik uit opmaken dat de medewerkers op de werkvloer een Windows omgeving gewend zijn. Ik ben daarom nagegaan of MySQL ook kan draaien in een Windows omgeving. Dit heb ik gedaan door de officiële site van MySQL te raadplegen. Daaruit bleek dat MySQL wel op tal van besturingssystemen kon draaien en ook op een Windows 2000 server.

5.5 Het vaststellen van de ontwikkelomgeving

Coldfusion van Macromedia is de belangrijkste programmeertaal bij i-Concept. Gezien deze situatie leek het mij verstandig om de software agent dan ook te gaan schrijven in Coldfusion. Op die manier kunnen de opdrachtgever en de bedrijfsmentor zelf eventuele modificaties aanbrengen.

Het wil echter niet zeggen dat ik andere programmeertalen of communicatietechnologieën eventueel ernaast kan gebruiken om dit te combineren met Coldfusion. Ik heb daarom via google.nl gezocht naar de mogelijkheden van Coldfusion met betrekking tot integratie van andere programmeertalen en protocollen. Bij het vond ik talen als XML, HR-XML, SOAP, screenscraping en WDDX. Van de meeste talen had ik wel eens iets gehoord of erover gelezen, maar ik heb er nog nooit echt webpagina's mee geschreven.

Via de officiële site van Macromedia kwam ik erachter dat WDDX wel al geïmplementeerd was in Coldfusion.

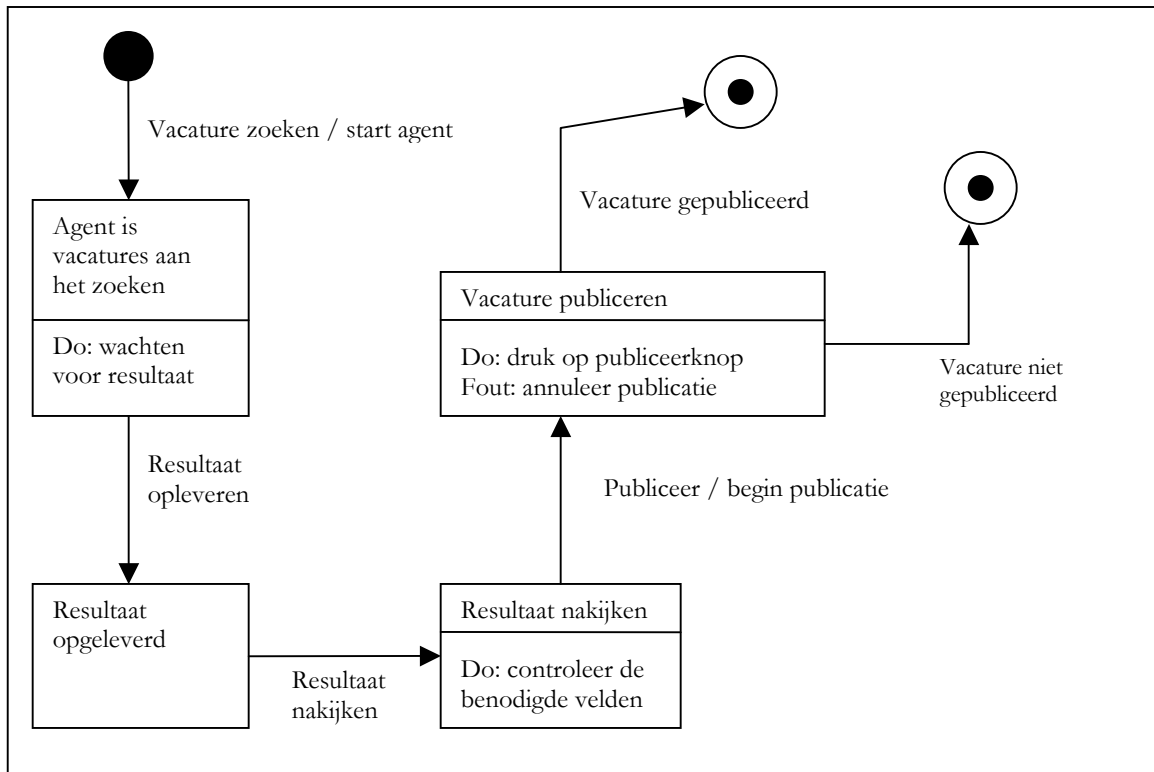
5.6 Het systeemconcept

5.6.1 Functionele eisen

Op basis van de functionele systeemeisen heb ik een systeemconcept ontwikkeld in de vorm van de volgende modellen: een state diagram, een activity diagram en een taakdiagram. Daarnaast heb ik op basis van de functionele systeemeisen die betrekking hebben op de database een aantal modellen opgezet te weten: het Object (EER) model, het relationele model en het fysiek implementatie model.

5.6.1.1 State diagram

Ik wilde inzicht krijgen in de user input dat nodig was om de software agent op te starten. Hiervoor heb ik een state diagram gebruikt. In figuur 2 is een overzicht te zien van het opgestelde state diagram

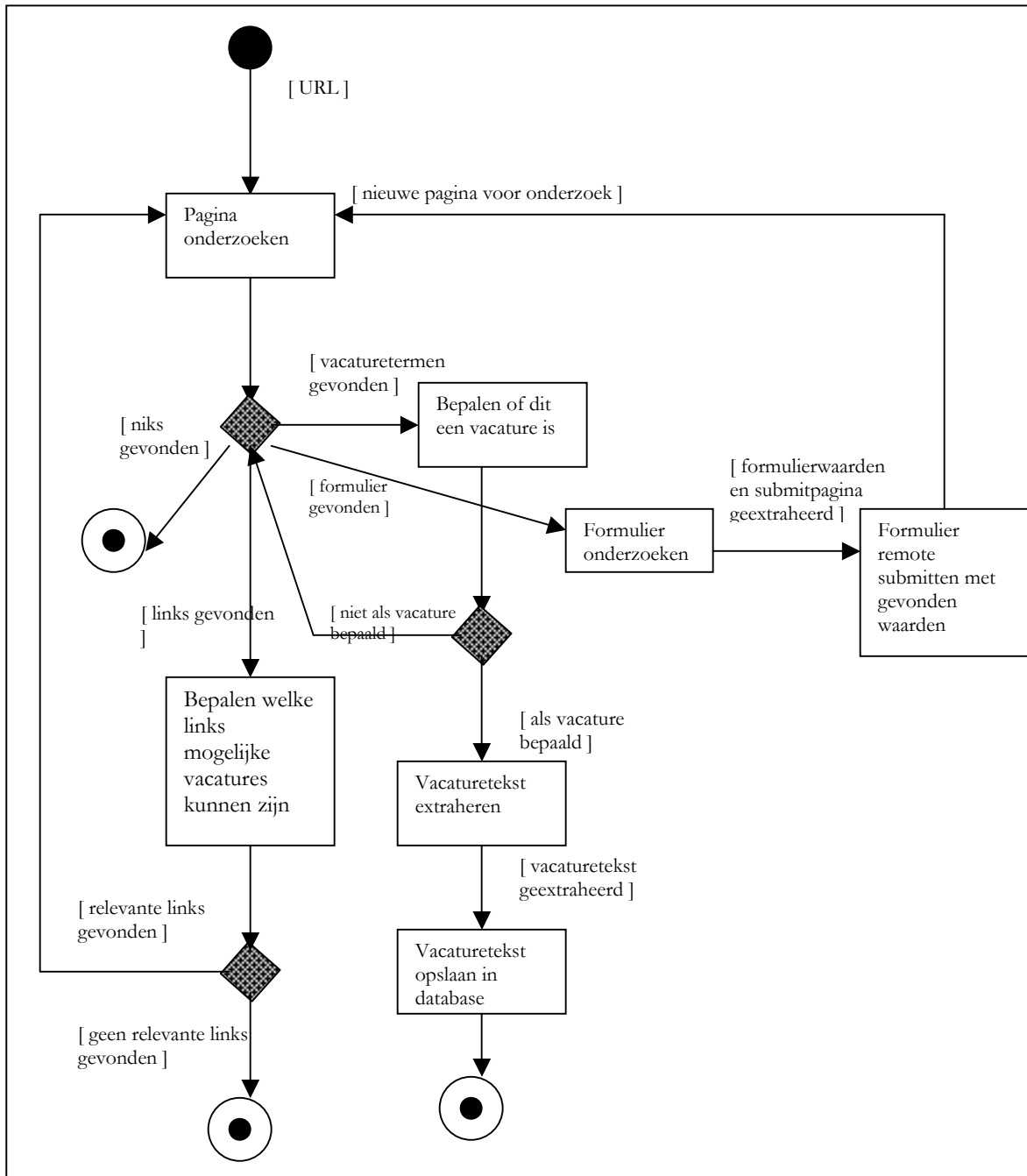


Figuur 2 – State diagram van de software agent

Achteraf gezien was het geen goede keuze om een state diagram te maken, omdat ik dit diagram verkeerd heb gebruikt. Het diagram heeft mij uiteindelijk helemaal geen duidelijkheid verschaft. Ik heb dit diagram dan ook niet meer gebruikt

5.6.1.2 Activity diagram

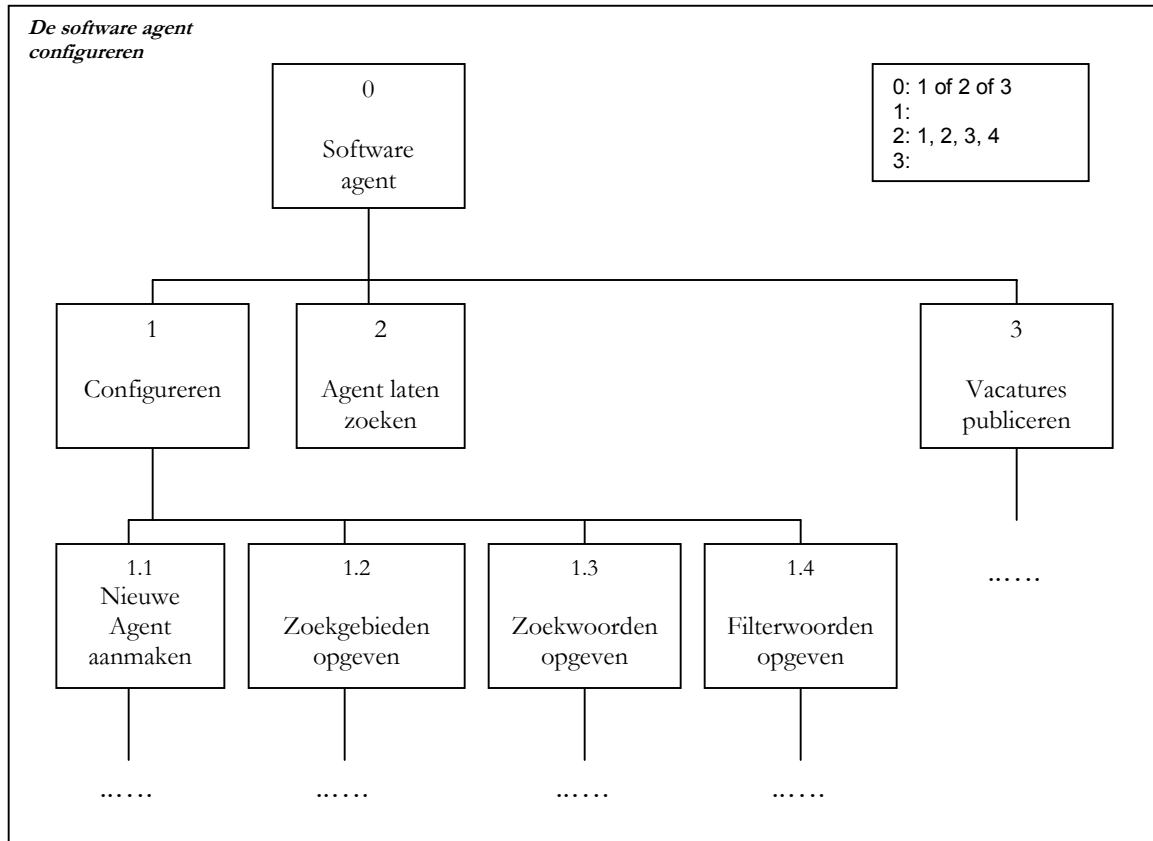
Veel van de functionele eisen van de software agent hebben betrekking op zaken onder de motorkap van de software agent. Ik heb daarom besloten om een activiteiten diagram te maken van de software agent, omdat je dan goed kan zien wat de software agent daadwerkelijk moet gaan doen. In figuur 3 is een overzicht te zien van de opgestelde diagram.



Figuur 3 – Activity diagram software agent

5.6.1.3 Taakdiagrammen

Op basis van de functionele eisen met betrekking tot de te verrichten handelingen en op basis van de geïnventariseerde taken van de huidige situatie, heb ik een taakdiagram opgesteld voor de nieuwe situatie. Ik heb dit diagram gemaakt, omdat het mij later bij het ontwerpen en ontwikkelen van de configuratieschermen kan helpen. Figuur 4 geeft een beeld van hoe het taakdiagram eruit ziet. Voor een volledige uitwerking van de taakdiagrammen verwijs ik u door naar de definitiestudie.



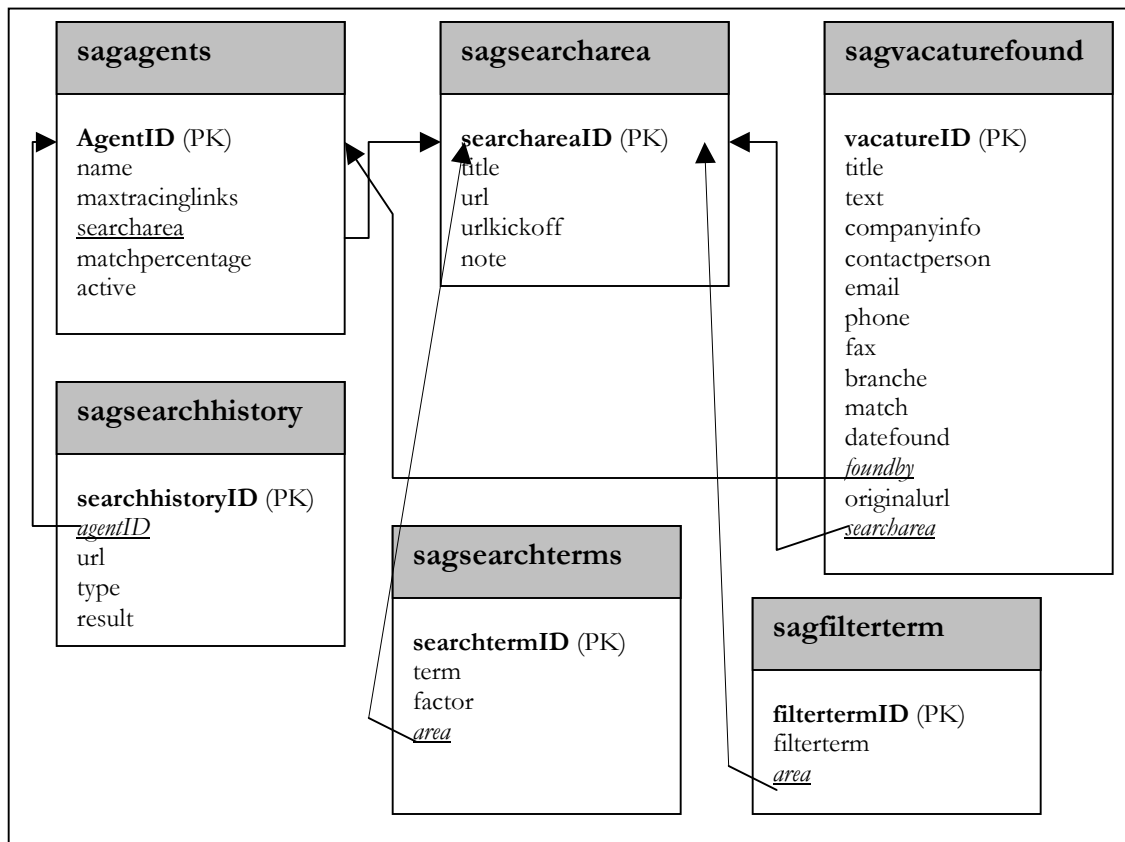
Figuur 4 – Taakdiagram software agent configureren

5.6.1.4 EER model

De huidige database van arbeidsmarkt.com heb ik geïnventariseerd en geanalyseerd (zie hoofdstuk “Inventariseren en analyseren van de huidige situatie”). De overstap die gemaakt gaat worden van MS Acces naar MySQL heeft geen gevolgen voor de huidige relaties tussen de geïnventariseerde tabellen en tupels. Voor een overzicht van de EER model van de huidige database verwijst ik u door naar de definitiestudie.

Om de informatie vereisten van de tabellen van de software agent vast te leggen, heb ik een EER model gemaakt. Figuur 5 geeft u een overzicht van de nieuw toegevoegde tabellen en hun relaties met elkaar.

Door toevoeging van de software agent zal de huidige database wel moeten worden uitgebreid met een aantal tabellen te weten: sagagents, sagfilterterm, sagsearcharea, sagsearchhistory, sagsearchterms, sagvacaturefound. Ik heb alle tabelnamen van de software agent laten beginnen met ‘sag’, omdat dit de afkorting is voor de agent binnen het intranet.



Figuur 5 – EER model

5.6.1.5 Relationeel model

Om er zeker van te zijn dat de nieuwe tabellen en tupels genormaliseerd zijn, heb ik een relationeel model van het EER model gemaakt. Mijn doel hierbij was om tabellen met eventuele meervoudige of samengestelde attributen op te splitsen. Uiteindelijk hoefde ik geen correcties aan te brengen in de tabellen. Zie figuur 6 voor een overzicht van het relationeel model.

- SagAgents (**agentID**, name, maxtracinglinks, matchpercentage, active, searcharea)
 - Primaire sleutel is agentID
 - Vreemde sleutel is searcharea refereert aan tabel sagsearcharea
- SagFilterTerm (**filtertermID**, filterterm)
 - Primaire sleutel is filterterm
- SagSearchTerms (**searchtermID**, term, factor)
 - Primaire sleutel is searchtermID
- SagSearchArea (**searchareaID**, title, url, urlkickoff, note)
 - Primaire sleutel is searchareaID
- SagSearchHistory (**searchHistoryID**, agentID, url, type, result)
 - Primaire sleutel is searchhistoryID
 - Vreemde sleutel is agentID refereert aan tabel sagAgents
- SagVacatureFound (**vacatureID**, title, text, bedrijf, contactperson, email, telefoon, fax, datefound)
 - Primaire sleutel is vacatureID
 - Vreemde sleutel foundby refereert aan tabel sagAgents
 - Vreemde sleutel searcharea refereert aan tabel sagSearchArea

Figuur 6 – Relationeel model software agent

5.6.1.6 Implementatiemodel

Nadat ik het relationeel model had gemodelleerd, heb ik een implementatie model van de database gemaakt. Met dit model kon ik goed nagaan welke wijzigingen er op het gebied van datatypen doorgevoerd moesten worden in verband met de conversie van de MS Acces database naar de MySQL database. Om er zeker van te zijn dat ik de juiste nieuwe datatypen heb gekozen, heb ik de documentatie van MySQL erop nageslagen. In figuur 7 ziet u de het implementatiemodel van de tabel “SAGagents” in MySQL.

```
DROP TABLE IF EXISTS sagagents

CREATE TABLE sagagents (
    agentID SMALLINT(6) auto_increment NOT NULL,
    name VARCHAR(50) NULL,
    maxtracinglinks INT(11) NOT NULL
    searcharea INT(11) NOT NULL,
    matchpercentage DOUBLE NOT NULL,
    active smallint(6) NOT NULL
    KEY (agentID)
)

DROP TABLE IF EXISTS sagvacaturefound

CREATE TABLE sagvacaturefound (
    vacatureID BIGINT(11) auto_increment NOT NULL,
    title VARCHAR(100) NULL,
    bedrijf VARCHAR(100) NULL,
    text LONGTEXT NULL,
    contactpersoon VARCHAR(100) NULL,
    email VARCHAR(100) NULL,
    telefoon VARCHAR(100) NULL,
    fax VARCHAR(100) NULL,
    datefound INT(8) NOT NULL,
    KEY (vacatureID)
)
```

Figuur 7 – Nieuwe table ‘users’ in MySQL

Tijdens het valideren van het implementatiescript van pilotdeel “Het overzetten van data” van de pilot “Database” heb ik een fout opgemerkt op de website. De datum werd namelijk niet goed weergegeven. De maanden en de dagen waren namelijk omgedraaid. Ik had toen besloten om voor alle velden met “DATE” als gegevenstype te veranderen in een numerieke waarde met een maximale waarde van 8 karakters. De belangrijkste reden voor deze oplossing is nog wel dat je zelf de opmaak van de datum in handen hebt en niet afhankelijk bent van instellingen van een bepaalde server. Ik heb toen het implementatiemodel moeten itereren. Ik heb gekozen om de datum die bij “datefound” wordt opgeslagen, op te slaan als een numerieke waarde. Door de datum op te slaan als een numerieke waarde, voorkom ik dat de datum op de server verkeerd wordt weergegeven en dat de datum verkeerd wordt ingevoerd in de datum. Dit is van essentieel belang omdat veel tabellen een datumveld bevatten en de site ook veel functionaliteiten heeft die afhankelijk zijn van een datum. Voor meer informatie verwijst ik u door naar paragraaf 6.2.2.2.3.

Verder ziet u datatypes als 'BIGINT' en 'SMALLINT'. In de MS Acces database komen deze typen overeen met respectievelijk 'LONG INTEGER' en 'INTEGER'. De grootte van het getal tussen de haakjes heb ik bepaald aan de hand van de waarden in de MS Acces database. Een tupel dat automatisch incrementeel in waarde stijgt en al een waarde heeft dat uit 5 karakters bestaat, heeft veel ruimte nodig om te kunnen groeien. Een tupel met BIGINT(20) als datatype kan dus flink groeien. Op deze manier heb ik gekeken naar alle numerieke tupels.

5.6.2 Interface en operationele eisen

De software agent zal als een module worden gemaakt zodat het makkelijk geïmplementeerd kan worden in het intranet van i-Concept. Omdat het intranet van i-Concept al een standaard lay-out heeft, heb ik besloten om de lay-out van de module los te koppelen van zijn functionaliteiten. De software agent zal namelijk worden gemaakt met de standaard templates van het intranet waarin de kleurinstellingen van het intranet al zijn opgenomen.

Met de gedachtegang dat de term 'software agent' medewerkers wellicht kan afschrikken, heb ik besloten om de interface van het configuratiescherm van de agent zo simpel mogelijk te houden. Ik zal geen dropdown-menu's of een andere vorm van verborgen knoppen gaan gebruiken, maar juist simpele buttons die direct leiden tot het gewenste scherm.

Daarnaast heb ik besloten om de indelingen van de schermen zoveel mogelijk te laten lijken op andere indelingen van toepassingen binnen het intranet, zodat het voor een medewerker lijkt alsof hij in een vertrouwde omgeving werkt.

Verder heb ik besloten dat het zoeken van de agent ingesteld kan worden met behulp van parameters. Zodoende kan de agent ook voor andere doeleinden gebruikt worden.

5.6.3 Performance en beveiligingseisen

5.6.3.1 Windows 2000 server

Uit een eerdere inventarisatie en analyse heb ik besloten om een Windows 2000 server te gaan gebruiken waarop de site, de database en de software agent komen te draaien. Ik heb hiervoor gekozen omdat MySQL ook in een Windows omgeving kan draaien en omdat de opdrachtgever en bedrijfsmentor al met een Windows besturingssysteem bekend zijn.

Om de server zo goed mogelijk te kunnen laten presteren wilde ik de server gaan tweakken. Dit wilde ik gaan doen door een aantal Windows services uit te schakelen zodat er meer geheugen beschikbaar was voor de nodige toepassingen. Welke services dat precies waren, wist ik nog niet precies. Dit zou ik verder nagaan bij het opzetten van het pilotontwikkelplan.

Op het gebied van beveiliging van de server heb ik gekeken hoe ik de gebruikersrechten kon toewijzen, omdat niet iedereen van alles mocht uitvoeren. Zo moest er een gebruiker zijn die alleen maar back-ups mocht maken en een gebruiker die volledige toegang had: de

administrator. Deze informatie kwam niet ter sprake tijdens het eerste gesprek met de opdrachtgever, maar later toen ik aan hem vroeg hoe de server ingesteld moest worden. Hij gaf daarna ook aan dat er een virusscanner geïnstalleerd moest worden die op de achtergrond moest kunnen draaien en waarbij het mogelijk was dat gevonden virussen gelijk werden opgeruimd zonder daarbij de server een halte toe te roepen. Ik heb toen het Internet afgezocht voor een dergelijk virusscanner en heb toen het programma 'AVG Virusscanner' gevonden.

De server zou uiteindelijk naast de eerste server in Amsterdam komen te staan. De back-ups die gemaakt zouden worden van de tweede server, moesten geplaatst worden op de eerste server. Ik heb ervoor gekozen om een programma ('Double Image') hiervoor te gaan gebruiken die het mogelijk maakt om op een veilige manier verbinding te leggen de twee servers en de back-ups kopieert van de ene server naar de andere server. Vanwege de tijdsplanning, leek het me niet verstandig om uit te gaan zoeken hoe je dit met een eigen geschreven batchscript kan verwezenlijken. Bovendien waren er al velen mij voorgedaan die een dergelijk programma hadden geschreven en die dat bovendien gratis op het Internet aanbood.

5.6.3.2 Coldfusion server

Gezien mijn keuze voor de te gebruiken programmeertaal en communicatietechnologie, moest ik de server ook uitrusten met een Coldfusion server die het mogelijk maakt om Coldfusionpagina's te interpreteren en uit te voeren. Deze server kon geoptimaliseerd worden door een aantal instellingen te veranderen te weten de time-out van de server en het aantal verbindingen dat is toegestaan op de server. Op het gebied van veiligheid, wilde ik de volgende maatregelen treffen. In de eerste plaats wilde ik een wachtwoord gebruiken om toegang tot het administratiegedeelte van de Coldfusion server af te schermen. Daarnaast wilde ik eventuele foutmeldingen van de software agent afschermen met behulp van standaard pagina's voor foutmeldingen, zodat bezoekers en eventueel kwaadwilligen niet kunnen achterhalen wat er fout ging bij de software agent en hier dus niet op kunnen inspelen.

5.6.3.3 MySQL database

MySQL was voor mij nog vrij nieuw. Ik heb daarom verschillende sites waaronder de officiële site van MySQL geraadpleegd voor informatie over hoe je een MySQL database zo goed mogelijk kon configureren voor een bepaald systeem.

Uiteindelijk heb ik op de officiële site informatie gevonden betreffende het configureren van de database. Deze informatie heb ik opgenomen in de definitiestudie. Het betreft informatie over verschillende configuratiemogelijkheden die dienen als voorbeelden. Ik heb gekozen om een zogenaamde “my-medium” configuratie te hanteren, omdat ik weet dat de server 256mb ram geheugen beschikt en deze configuratie daarop was ingesteld. Figuur 8 geeft u beeld van hoe het configuratiebestand “my.ini” eruit ziet.

```
# This will be passed to all mysql clients
[client]
#password=*****
port=3306

# The MySQL server
[mysqld]
port=3306
#socket=MySQL
skip-locking
set-variable      = key_buffer=16K
set-variable      = max_allowed_packet=1M
set-variable      = thread_stack=64K
set-variable      = table_cache=4
set-variable      = sort_buffer=64K
set-variable      = net_buffer_length=2K
server-id         = 1

# Uncomment the following if you want to log updates
#log-bin

basedir = c:/mysql/
datadir = c:/mysql/data/

[mysqldump]
quick
set-variable      = max_allowed_packet=16M

[mysql]
no-auto-rehash

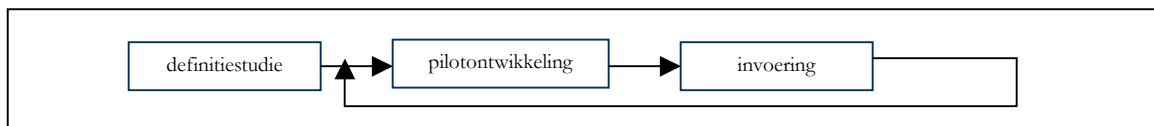
[myisamchk]
set-variable      = key_buffer=8M
set-variable      = sort_buffer=8M
```

Figuur 8 – Configuratiebestand van de MySQL server

Verder heb ik een aantal instellingen aangepast met betrekking tot de logfiles, het cachen van queries en de verbindingen met de server. Uit ervaring weet ik dat de logfiles van de MySQL server tijdens een ontwikkeling van een product enorm groot kunnen worden. Ik heb daarom gekozen om geen logfiles aan te maken, omdat die onnodig veel ruimte op de harde schijf in beslag nemen. Daarnaast kan het logproces ook veel geheugen in beslag nemen met als gevolg dat de server trager wordt. Met de informatie dat het zoeken op de huidige site flink in snelheid is gedaald, wilde ik de queries gaan cachen zodat veelvoorkomende zoekacties snel resultaat opleveren voor de bezoekers.

5.6 Pilotplan

De systeemeisen en het systeemconcept heb ik voor de software agent kunnen inventariseren. Ik ging er daarom van uit dat de fase “definitiestudie” eenmalig was. Daarnaast was er weinig tijd om workshops of uitgebreide interviews met de gebruikers en opdrachtgevers te doen, omdat zij hun kostbare tijd in andere commerciële projecten dienen te steken. Een iteratie van de definitiestudie zou niet gemakkelijk gaan. Ik heb daarom besloten om het project ‘incrementeel’ op te leveren. De twee fasen pilotontwikkeling en invoering wilde ik namelijk wel volgens de iteratieve benadering ontwikkelen en invoeren in de organisatie. Figuur 9 geeft een schematisch overzicht van welke fasen ik ga itereren.



Figuur 9 – Schematisch overzicht de strategie incrementeel opleveren

Het systeemconcept heb ik opgesplitst in drie pilots te weten de server, de database en de software agent. Ik heb het project in deze pilots opgesplitst, omdat ik op deze manier het project voor mezelf een duidelijke structuur heb kunnen geven. Ik wist nu welke pilots afhankelijk van elkaar waren en hoe ze afhankelijk van elkaar waren.

Ik heb aan de hand daarvan ook een volgorde van de te ontwikkelen pilots kunnen bepalen. Als eerste wilde ik de server op zetten, omdat de pilots “database” en “software agent” niet kunnen functioneren zonder een server. Vervolgens wilde ik de database en alles wat daarmee te maken had maken en in orde brengen, omdat de software agent deels afhankelijk was van de database. Immers, zonder een database is het moeilijk om gevonden data op te slaan. De pilot “software agent” heb ik ingedeeld voor het realiseren van de gewenste functionaliteiten die de software agent dient te hebben.

Iteratie 1

Omdat er juist bij deze pilot “Software Agent” veel nieuwe technologische aspecten erbij komen, was de technische invulling van de software agent nog niet helemaal duidelijk. De gewenste functionaliteit had ik wel beschreven bij de systeemeisen. Echter, het was vooraf zeer moeilijk in kaart te brengen hoe de wat het uiteindelijke resultaat zou moeten worden. Daarom was ik voor deze pilot overgestapt van iteratiestrategie “incrementeel opleveren” naar een evolutionair iteratiestrategie. Als het nodig was, dan zou ik de definitiestudie voor deze pilot ook itereren. Een andere reden waarom ik voor deze pilot voor een evolutionaire aanpak heb gekozen, is omdat deze iteratiestrategie goed past bij het karakter van programmeren en valideren. Ik zal steeds stukje voor stukje moeten gaan programmeren en valideren, omdat het meeste programmeerwerk nog nieuw voor mij is. De software agent zal daarom ook op die manier stukje bij beetje evolueren naar zijn uiteindelijke vorm. Voor meer informatie hierover verwijs ik u door naar paragraaf 6.3.1

6. Pilotontwikkeling

6.1 Pilot server

6.1.1 Pilotontwerp

Voor de server heb ik in het pilotontwerpplan een opsomming gemaakt van de benodigde software pakketten die ervoor moeten zorgen dat de ‘agent’ op een stabiele en beveiligde manier kan opereren. Elk software pakket heb ik als een pilotdeel ingedeeld, omdat ik op die manier elk pilotdeel beter kon beheersen. De pilot “server” bestond uit vijf pilotdelen namelijk de windows 2000 server, de webserver, de Coldfusion server, “Double Image”, “AVG virusscanner” en “Remote desktop”. Met de gekozen indeling waren de pilotdelen compact en was de pilot gemakkelijker afleidbaar. Aan de hand van de onderlinge afhankelijkheden, heb ik de pilotdelen onderling geprioriteerd. Figuur 10 geeft een overzicht van de pilotdelen met hun onderlinge prioriteiten.

Prioriteit	Pilotdeel
1	De Windows 2000 server
2	De webserver
3	De Coldfusion server
4	AVG virusscanner
5	Double image
6	Remote Desktop

Figuur 10 – de pilotdelen van de pilot “server” en hun prioriteiten

De webserver, de colfusion server en de programma’s “AVG virusscanner”, “Double Image” en “Remote Desktop” waren allemaal afhankelijk van de windows 2000 server. De windows 2000 server moest daarom gedwongen als eerste worden geïnstalleerd en geconfigureerd worden. Vervolgens heb ik gekozen om de webserver te installeren omdat anders de Coldfusion server niet goed zou werken. Om de backups virusvrij te houden, heb ik eerst het gratis programma AVG virusscanner geïnstalleerd en daarna het programma Double Image.

6.1.2 De installatie en configuratie

6.1.2.1 Windows 2000 server

Volgens de planning ben ik eerst de windows 2000 server gaan installeren. Helaas ging de installatie hiervan in eerste instantie niet goed. Veel gekopieerde bestanden bleken corrupt te zijn. Ik heb toen gevraagd om de originele installatie cd. Bij de installatie had ik de keuze uit verschillende soorten mogelijkheden van formatteren: FAT, FAT32 en NTFS. FAT staat voor “File Allocation Table” en geeft aan hoe de harde schijf aan clusters wordt opgedeeld. NTFS (NT Filesystem) en is een andere soort indeling dan de FAT en de FAT32. Met het oog op beveiliging heb ik gekozen voor de NTFS, omdat met deze “filesystem” beveiliging

biedt voor bestanden. Hiermee kunnen we onze data beter beveiligen tegen hackers en kwaadwilligen.

Nadat ik de server had geïnstalleerd, had ik eerst alle updates voor dit softwarepakket gedownload en geïnstalleerd. Nu was ik er zeker van dat de server op het gebied van beveiliging up-to-date was en dat er geen zogenaamde beveiligingsgaten meer aanwezig waren.

Vervolgens heb ik de verschillende gebruikersgroepen aangemaakt en gedefinieerd voor toegang tot de server. Dit heb ik gedaan via de “Local policy” configuratiescherm. Dit zijn de volgende “gebruikers”:

- Admin
- Backup

De gebruiker “admin” is een afgeleide van de standaard aanwezige “administrator”. Ik heb deze tweede administrator account aangemaakt, om zodoende mijn bedrijfsmentor en de opdrachtgever toegang te geven tot de server met alle rechten. Zij moeten immers snel en met alle rechten bij de server kunnen komen, om eventuele problemen met de server op te kunnen lossen. De gebruiker “backup” heeft niet alle rechten. Hij kan geen instellingen van de server veranderen. Het enige wat hij kan doen is nagaan of het backuppen goed is gegaan en indien nodig handmatig backup’s uitvoeren.

Ook ben ik nagegaan of de “guest” account was uitgeschakeld. Dit account kan namelijk een eventuele ingang zijn voor kwaadwilligen. Verder heb ik alle overige standaard accounts op de server heb ik uitgeschakeld, omdat die niet nodig waren.

6.1.2.2 De webserver

De webserver is nodig om websites te kunnen hosten. Bij de Windows 2000 server wordt standaard het pakket IIS 5.0, “Internet Information System” geïnstalleerd. Normaliter wordt deze webserver gebruikt om zogenaamde asp-pagina’s te hosten. Aangezien de agent op een Coldfusion server gaat draaien heb ik dit moeten aanpassen. In het configuratiescherm voor de toegestane extensies heb ik een extensie toegevoegd te weten “cfm”. Dit is de extensie van Coldfusion webpagina’s.

De webserver IIS hanteert een bepaalde directory structuur om zodoende de webpagina’s te kunnen hosten. Om de structuur zo herkenbaar mogelijk te houden, heb ik de directory structuur van i-Concept’s eerste server overgenomen. Figuur 11 geeft een overzicht van die indeling.



Figuur 11 – Directory structuur van de webserver

Normaliter worden de webpagina's direct in de directory 'wwwroot' geplaatst. Ter beveiliging heb ik voor de website arbeidsmarkt.com en de software agent nog twee niveau's dieper een directory aangemaakt en tevens een virtuele directory in het IIS aangemaakt die verwijst naar de fysiek aangemaakte directory. Een virtuele directory zorgt ervoor dat de website via een bepaald adres te benaderen is en tevens dat eventuele andere websites die ook op de twee server worden gehost niet worden blootgesteld.

6.1.2.3 De Coldfusion server

Na het configureren van de webserver, heb ik de Coldfusion server geïnstalleerd. Deze server maakt het mogelijk om webpagina's geschreven in de Coldfusion taal te kunnen interpreteren en uit te voeren. Met het oog op beveiliging en performance heb ik de server op een aantal gebieden geconfigureerd. Dit betreft de toegang tot het administratiegedeelte van de server, de te gebruiken database stuurprogramma, het cachen van webpagina's en database queries. Daarnaast heb ik instellingen veranderd met betrekking tot foutmeldingen en het vastleggen hiervan.

Het administratiegedeelte van de server is te bereiken via de internetbrowser via een bepaalde poort. Echter, niet iedereen mag hier toegang tot krijgen. Ik heb daarom een login en een wachtwoord ingesteld voor de server.

Coldfusion heeft een aantal standaard verbindingsmogelijkheden met verschillende databases. Voorheen gebruikte i-Concept de 'ODBC' (Open Database Connectivity) verbinding voor hun DSN's (Data Source Name) met MS Acces. Het is echter niet mogelijk om hiermee een verbinding te maken met de MySQL database. Ik heb hiervoor een alternatief moeten zoeken. Ik heb toen de officiële website van MySQL geraadpleegd en kwam toen met de oplossing om de zogenoemde 'MySQL ODBC' stuurprogramma te installeren. Nadat ik dit stuurprogramma had geïnstalleerd bleek dat ik dit stuurprogramma niet kon selecteren via het administratiegedeelte van de Coldfusion server. Uit ervaring weet ik dat een DSN ook te configureren is via een configuratiescherm van Windows zelf. Ik heb toen een DSN aangemaakt met de 'MySQL ODBC' stuurprogramma en vervolgens gekeken of de aangemaakte DSN ook in de Coldfusion server te voorschijn kwam. De aangemaakte DSN met het 'MySQL ODBC' stuurprogramma verscheen toen wel in de lijst van aanwezige DSN's.

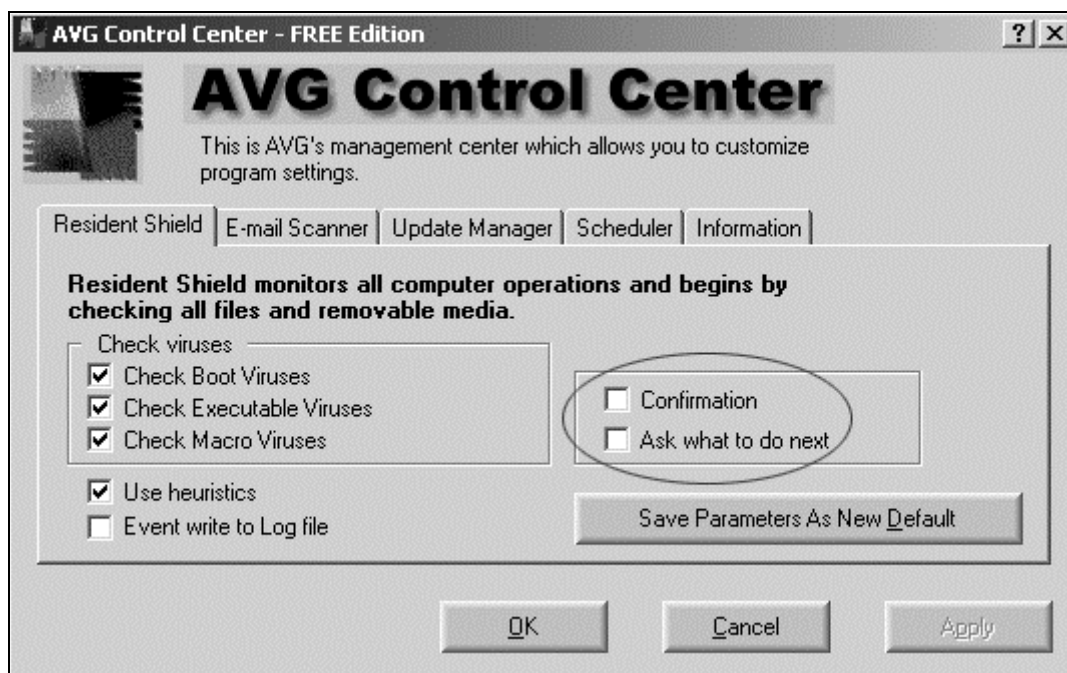
Het cachen van webpagina's en database queries heeft als doel de website en de data op die website sneller te kunnen weergeven aan bezoekers. Het werkelijke cachen van data retrieval moet bij Coldfusion in de code van de webpagina's zelf worden aangegeven. In het administratiegedeelte kan worden aangegeven wat de limiet is voor het aantal queries dat gecached kan worden. Ik wist niet wat het limiet kon zijn bij een server met 256mb aan geheugen. Ik heb toen via google.nl met de zoekacties "values query cache coldfusion". Dit leverde niks bruikbaars op. Ik heb toen gezocht met de zoekactie "query cache limit coldfusion server" en kwam toen op deze pagina terecht:

"http://www.experts-exchange.com/Web/WebDevSoftware/ColdFusion/Q_20782296.html".

Op deze pagina vond ik een voorbeeldsituatie van iemand die de limiet had gesteld op 20 bij een computer met 512mb aan RAM geheugen. Verder werd er aangegeven dat voor 100 cached queries wel minstens 1gb aan geheugen aanwezig moest zijn. Ik heb hieruit kunnen concluderen dat ik de limiet niet te hoog moest zetten, aangezien de server 256mb aan geheugen heeft. Ik heb toen besloten om de limiet van aantal queries dat gecached kan worden te zetten op 10. Dat is de helft van de waarde dat werd aangegeven bij het voorbeeld. Voor het loggen van foutmeldingen kon ik kiezen uit twee mogelijkheden: een korte notatie en een lange notatie. Ik heb gekozen voor een korte notatie, omdat een lange notatie alleen maar meer ruimte in beslag zou gaan nemen.

6.1.2.4 AVG virusscanner

Nadat ik de Coldfusion server had geconfigureerd, ging ik me bezig houden met het installeren en configureren van de gratis virusscanner: AVG. De belangrijke taak die de virusscanner moest hebben, was het kunnen onderscheppen van virussen die werden getransfereerd naar de server. Dit kon namelijk gebeuren indien een werkzoekende op arbeidsmarkt.com een cv document kopieert naar de server. Ik ben daarom in het configuratiescherm van de virusscanner nagegaan of het mogelijk was om directe maatregelen te treffen indien er een virus wordt aangetroffen bij bepaalde kopieer acties. Tevens moest de virusscanner de server niet tot een halte roepen, maar het virus op de achtergrond verwijderen. Figuur 12 geeft u zicht in de configuratie van het programma.



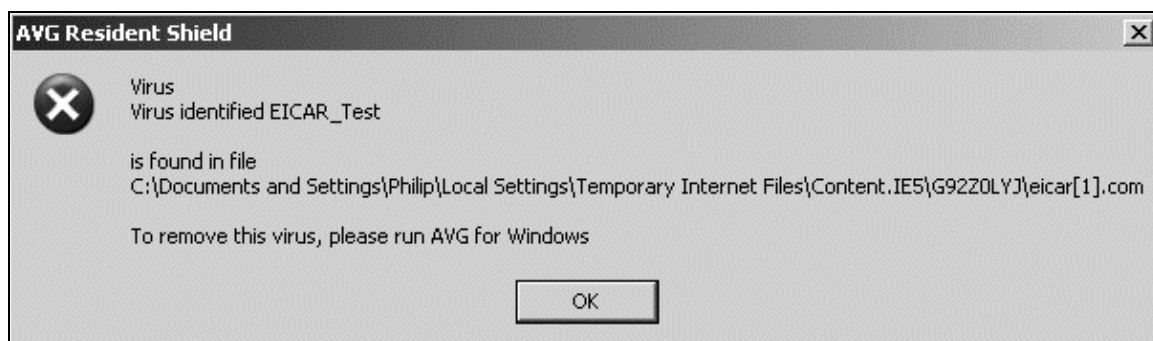
Figuur 12 – Configuratiescherm AVG virusscanner

Zoals u ziet staat er in dikgedrukte letters dat de “Resident Shield” alle computeracties in het oog houdt. Dit zijn dus ook acties als kopiëren en verplaatsen. Deze acties gebeuren juist bij het uploaden naar de server. De opties “Confirmation” en “Ask what to do next” heb ik uitgeschakeld, omdat ik de virusscanner niet wil laten interfereren met de server. De

virusscanner moet zijn taken op de achtergrond blijven uitvoeren. Bij het tabblad “E-mail Scanner” heb ik alle opties met betrekking tot het scannen van e-mail uitgeschakeld, omdat de server geen e-mail ontvangt of verstuurd. Het automatisch updaten van de viruslijst, heb ik gezet om 05.00u in de ochtend, omdat dan de server in een rust toestand verkeert. Het periodiek scannen naar virussen heb ik ook uitgeschakeld, omdat ik ervan uit ging dat de “Resident Schield” alles op zou vangen. Verder zijn er geen potentiële infecties van virussen mogelijk, omdat de server verder geen bestanden en e-mail ontvangt.

Iteratie 1

Na de validatietest bleek dat de virusscanner de server toch tot een halte riep en dat het virus nog niet was verwijderd. Zie figuur 13 voor het resultaat. Voor meer informatie over het testen van dit pilotdeel verwijs ik u door naar paragraaf 6.1.3.2.



Figuur 13 – AVG viruswaarschuwing

Ik heb toen besloten om de virusscanner om de zoveel tijd een scan te laten uitvoeren, omdat ik zeker wist dat de virusscanner dan het gevonden virus automatisch zou verwijderen en hiervan een melding zou maken in een rapport. Ik heb toen het periodiek scannen aangezet, wat ik voorheen niet had aangezet omdat ik ervan uit ging dat de scanner bij een bepaalde computeractie een gevonden virus direct zou verwijderen. De “Windows scheduler” zou dan deze taak moeten uitvoeren ongeacht er een user is ingelogd op de server of niet. De “Resident Shield” heb ik uitgezet, omdat deze functie niet het gewenste resultaat opleverde.

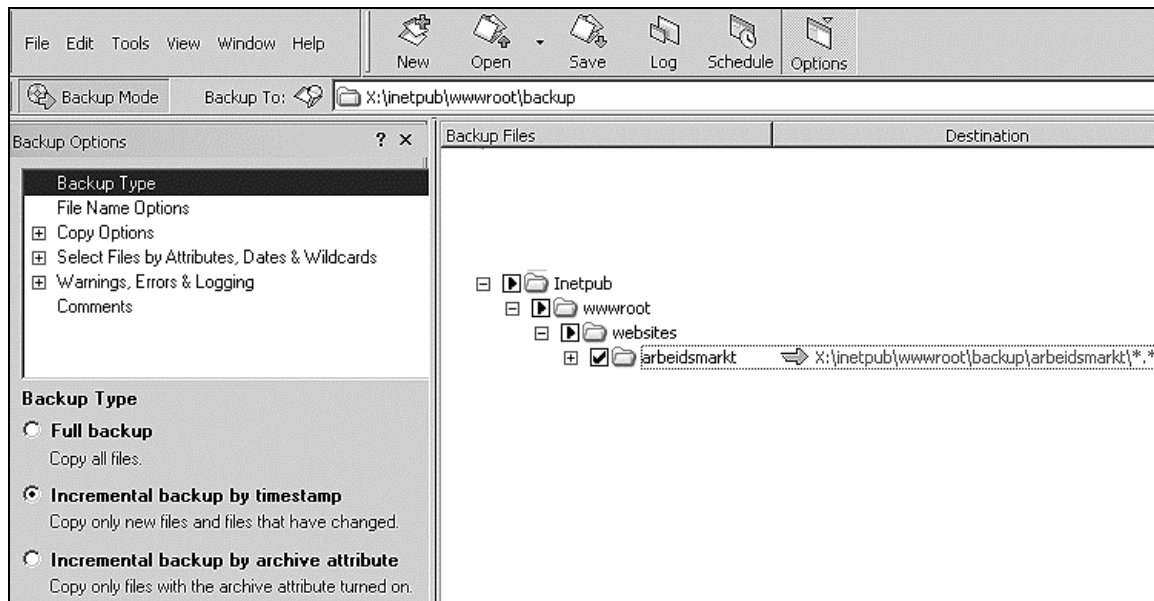
Iteratie 2

Na de tweede validatietest bleek, dat dit pilotdeel nog steeds niet naar behoren werkte. Indien een user was uitgelogd, voerde de “Windows scheduler” de geplande scan niet uit. Ik ging toen via google.nl zoeken met de zoektermen “scheduled task”, “avg” en “user logged off” naar een mogelijke oplossing. De zoekresultaten leverde niks bruikbaar voor mij op. Ik heb toen op de officiële site van AVG Virusscanner gezocht bij de kopjes “Support” en “FAQ”¹. Echter, dit probleem kwam niet voor in de lijst van problemen en oplossingen. Ik heb toen de oplossing van de eerste iteratie opnieuw aangenomen. Echter, dit keer had ik als voorwaarde gesteld dat de user altijd ingelogd moest zijn. Om de server toch vergrendeld te hebben, moest de user de server “locken” in plaats van uit te loggen. De server bevond zich dan alsnog in een beveiligde toestand.

¹ http://www.grisoft.com/faq/us_faqindex.php?id_rodice=26

6.1.2.5 Double Image

Voor het maken van back-ups van de database werd het programma 'Double Image' gebruikt. Dit programma was namelijk goed geschikt om back-ups op een veilige manier automatisch te laten verlopen over een netwerk. De nieuwe server moest zijn back-ups namelijk plaatsen op de eerste server. Figuur 14 geeft een overzicht van de instellingen van het programma.

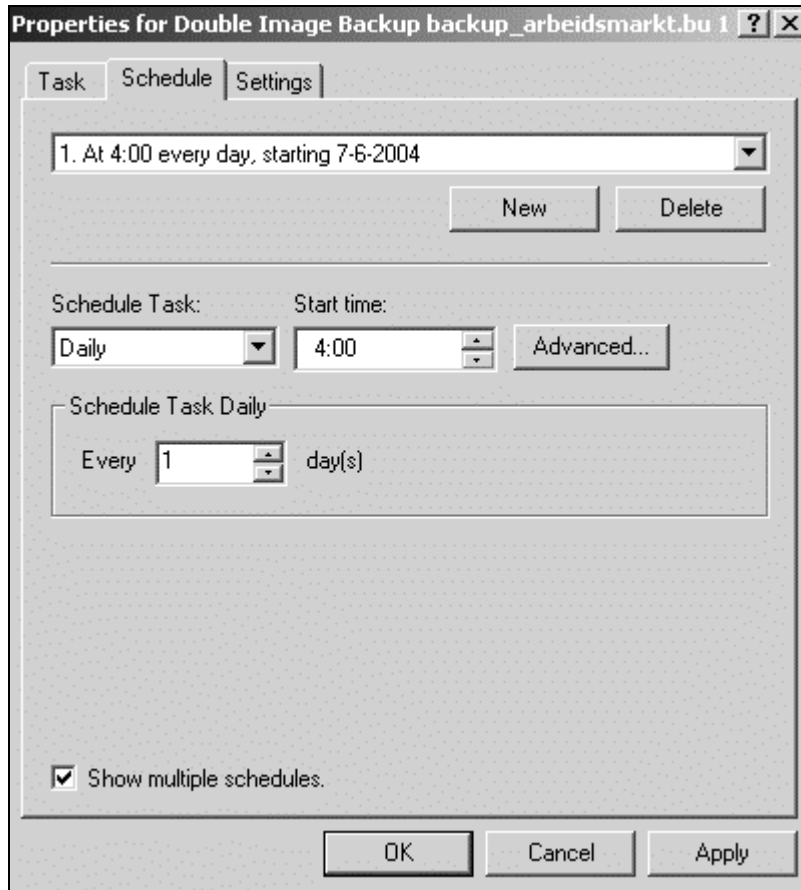


Figuur 14 – Bron en bestemming van de backup

In het programma kon ik aangeven wat de bron was. Dit heb ik gesteld op: "c:\inetpub\wwwroot\websites\arbeidsmarkt". De plaats van bestemming voor de backup heb ik gesteld op: "x:\inetpub\wwwroot\backup". De reden waarom het station de letter X als aanduiding heeft gekregen, is omdat het om een netwerkstation gaat. De werkelijke bestemming staat namelijk op de eerste server. Dit netwerkstation heb ik ook eerst aan moeten maken via de netwerkomgeving van Windows, anders gaf het programma een foutmelding dat het netwerkstation niet aanwezig was.

In figuur 17 ziet u ook dat er drie verschillende backup methodes waren. Ik kon namelijk kiezen uit de volgende methodes: 'Full backup', 'Incremental backup by timestamp' en 'incremental backup by archive attribute'. Ik heb gekozen voor de incrementele backup methode op basis van de tijd, omdat het niet nodig is om steeds bestanden te back-uppen die niet veranderd zijn. Dit laatste zou ook het geval zijn indien ik zou hebben gekozen voor de incrementele backup methode op basis het 'archief'-attribuut van bestanden. Bij deze methode worden namelijk alleen de bestanden gekopieerd, waarbij is aangegeven dat ze als archief beschouwd kunnen worden. Dit zou dus betekenen dat indien archiefbestanden niet zijn gewijzigd ze alsnog worden gekopieerd voor een backup.

Verder heb ik van het back-uppen een taak gemaakt die de Windows server om de zoveel tijd dient uit te voeren. In het configuratiescherm van de taak heb ik opgegeven om de hoeveel tijd en op welk tijdstip er een backup gemaakt moet worden. Figuur 15 geeft u een overzicht van deze instellingen.



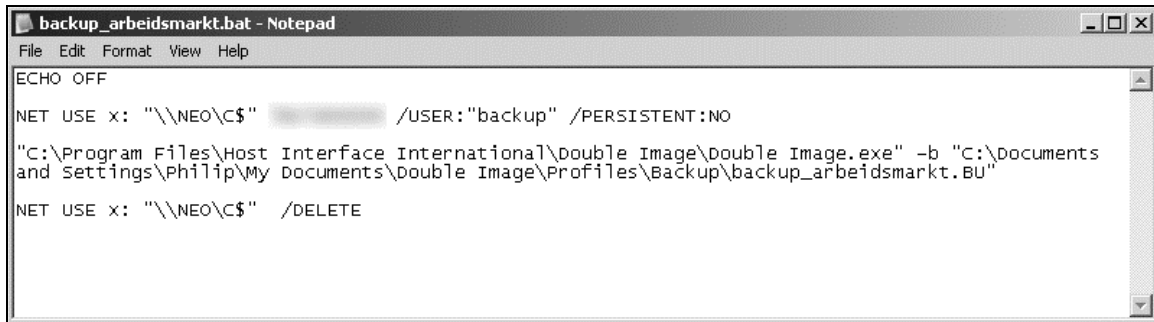
Figuur 15 – Eigenschappen van de geplande taak

Zoals u kunt zien heb ik voor de tijd dat er een backup gemaakt moet worden op 04.00u in de ochtend gezet. Hierbij heb ik aangenomen dat de meeste mensen slapen en dat de server op dat tijdstip niet druk wordt bezocht. Zodoende kunnen bezoekers van de website geen last krijgen van eventuele vertragingen. Om deze aanname bevestigd te krijgen heb ik de logfile van de eerste server geraadpleegd. Uit deze logfile blijkt dat er geen bezoekers rond dat tijdstip de website arbeidsmarkt.com bezoeken.

Iteratie 1

Na de eerste "black-box" functionaliteitstest bleek dat dit pilotdeel niet aan de functionaliteits-, "het automatisch maken van backups van vacatures", voldeed. Voor meer informatie over de functionaliteits-eisen verwijs ik u terug naar hoofdstuk 5.2 "Inventariseren van de eisen". Voor meer informatie over het testen van dit pilotdeel verwijs ik u door naar paragraaf 6.1.3.2. In deze iteratie heb ik geprobeerd te achterhalen wat het probleem zou kunnen zijn voor het niet automatisch uitvoeren van de geplande back-up taak. Ik heb toen op de website van de makers van dit softwarepakket gezocht naar oplossingen. Door hun

site systematisch na te gaan op termen als “FAQ”, “Support” en “Help”, kwam ik terecht op deze webpagina: “How To Automate Secure Network Backups”². Op deze pagina werd precies uitgelegd hoe je op een beveiligde manier geautomatiseerde back-ups kan uitvoeren over een netwerk. Dit kon worden bewerkstelligd met behulp van een geautomatiseerd script, een CMD (command line batch). In figuur 16 ziet u uit welke regels het script bestaat.



```
backup_arbeidsmarkt.bat - Notepad
File Edit Format View Help
ECHO OFF
NET USE x: "\\NEO\C$" /USER:"backup" /PERSISTENT:NO
"C:\Program Files\Host Interface International\Double Image\Double Image.exe" -b "C:\Documents
and Settings\Philip\My Documents\Double Image\Profiles\Backup\backup_arbeidsmarkt.BU"
NET USE x: "\\NEO\C$" /DELETE
```

Figuur 16 – Batchfile voor het automatiseren van back-uppen

Het commando “ECHO OFF” in de eerste regel geeft aan dat de volgende scriptregels niet letterlijk te zien zijn op het scherm. De tweede regel maakt een netwerkstation “X” aan waarbij de gebruiker “backup” verbinding maakt met de computer “NEO”. Vanwege bedrijfsgeheimhouding heb ik het wachtwoord weggehaald. In de derde regel laat ik het programma “Double Image” mijn back-up taak uitvoeren te weten “backup_arbeidsmarkt.bu”. Met de laatste scriptregel verbreek ik de netwerkverbinding en verwijder ik het station “X”. Als laatste heb ik dit script geplaatst in de “Windows scheduler”, zodat het door de server weer op een bepaald tijdstip automatisch uitvoert.

6.1.2.6 Remote Desktop

De server moet ook op afstand te benaderen zijn, omdat de server uiteindelijk in Amsterdam geplaatst zal worden. Om dit te kunnen realiseren heb ik het programma ‘Remote Desktop’ van Microsoft gebruikt, omdat dit programma met een minimum aan data heel snel de schermen van de server op afstand kan laten zien. Nadat de installatie klaar was, was het programma gereed voor gebruik. Er zaten geen beveiligingsopties op, omdat de beveiliging door de Windows 2000 server zelf wordt gedaan. Wanneer iemand verbinding maakt met de server met ‘Remote Desktop’ dan moet hij alsnog inloggen met een van de drie aanwezige accounts die ik eerder had gedefinieerd bij het configureren van de Windows 2000 server.

Iteratie 1

Na de eerste validatie met behulp van de “black-box” testmethode, bleek dat dit programma niet aan de beveiligingseis “de server moet beveiligd zijn tegen buitenstaanders” voldeed. Voor meer informatie over het testen van dit pilotdeel verwijs ik u door naar paragraaf 6.1.3.2. Het probleem lag bij de configuratie van de Windows 2000 server zelf. Deze conclusie heb ik getrokken uit het feit dat het programma “Remote Desktop” zelf niks aan beveiliging doet. Met deze redenering kon ik gericht zoeken naar een oplossing. Ik wilde

² “<http://www.hostinterface.com/DI5/Support/HOWTO-AutomateNetworkRemoteNetworkBackups.htm>”

namelijk weten hoe ik het automatisch beëindigen van een sessie kon uitschakelen. Om achter de oplossing te komen, heb ik via google.nl systematisch gezocht met de zoektermen “local security policy”, “automatic log off” en “idle session” naar oplossingen. Door met deze zoektermen te zoeken kwam ik op deze webpagina terecht: “Planning of Group Policy under Windows 2000” (<http://www.bsi.de/gshb/english/s/s02231.html>). Op deze webpagina werd aangegeven dat de optie “Amount of Idle Time required before suspending session” op 0 minuten gezet moest worden. Standaard heeft deze optie een waarde van 15 minuten dat inhoudt dat de server de user na 15 inactieve minuten automatisch uitlogt.

6.1.3 Beoordeling en test

6.1.3.1 De servers

De Windows 2000 server, de webserver en de Coldfusion server kon ik in principe al testen. Elk pilotdeel kon ik afzonderlijk testen op performance issues als schaalbaarheid, response time, CPU verbruik en geheugen verbruik. Ik heb echter besloten om dit nog niet te doen, omdat ik in feite wil weten wat de resultaten zijn van deze servers in combinatie met de MySQL server. De website arbeidsmarkt.com is immers grotendeels afhankelijk van de database. Daarnaast kon ik het risico lopen dat eventuele aanpassingen aan de servers als gevolg van de behaalde resultaten, niet meer bruikbaar zijn of ongeschikt zijn in combinatie met de MySQL server. In dat geval zouden de behaalde resultaten afgewogen tegen de tijd, een laag rendement hebben. In paragraaf 6.2.3 zal het testen van deze drie servers in combinatie met de MySQL server aanbod komen.

6.1.3.2 AVG Virusscanner

Het programma “AVG Virusscanner” heb ik getest met de “black-box” functionaliteit testmethode. Ik wilde namelijk weten of het programma daadwerkelijk het gewenste resultaat oplevert, zonder daarbij iets van de codering van het programma af te weten. De server moest namelijk virusvrij blijven. Tevens moest de virusscanner de server niet tot een halte roepen, omdat dan alle de andere processen ook stil komen te staan. Voor meer informatie omtrent de beveiligingseisen verwijs ik u terug naar paragraaf 5.2. Voor het testen heb ik een testscenario geschreven. In dit scenario wordt vanaf een computer in het lokale netwerk een virus, de “EICAR_test.com” virus, gestuurd naar de server. Als voorzorgmaatregel heb ik de “removal kit” gedownload mocht er iets fouts gaan. Hiermee kan ik het virus alsnog verwijderen, zonder daarbij de server schade aan te richten. Mijn verwachting was dat de AVG Virusscanner het virus direct zou opmerken en deze op de achtergrond zou verwijderen. Het resultaat van de test was iets anders dan ik verwacht had. De virusscanner riep de server juist wel tot een halte en gaf daarbij tevens een waarschuwingsbericht waarin werd aangegeven dat de user het virus nog moest verwijderen. Dit resultaat was niet gewenst en ik moest hiervoor een oplossing bedenken. Ik had daarom besloten om dit pilotdeel te itereren.

Testen na 1^e iteratie

Na de eerste iteratie van dit pilotdeel heb ik dezelfde testmethode gebruikt als bij de eerste oplossing. De oplossing van de eerste iteratie verschilt in het periodiek scannen en de uitgeschakelde “Resident Shield” met de eerste oplossing. Het periodiek scannen moest ook gebeuren indien de user niet was ingelogd op de server. Wat betreft het testscenario heb ik hetzelfde scenario aangehouden, omdat anders de resultaten misschien door iets anders beïnvloed kunnen worden. Ook heb ik dezelfde voorzorgsmaatregel gehouden als bij de eerste test. Het enige wat ik er wel in moest veranderen was, dat de “Windows scheduler” nu een rol speelde bij het testen. Om het testen met de “Windows scheduler” mogelijk te maken, heb ik aangenomen om de tijd van scannen steeds 1 minuut na aanvang van de test uit te voeren. Mijn verwachting was dat de AVG virusscanner nu wel het virus opmerkt en

deze automatisch verwijderd en dit in een rapport verwerkt. Het resultaat was nu helemaal niet gewenst, omdat de “Windows scheduler” de taak steeds niet uitvoerde. Met andere woorden, er werd nu dus helemaal niet gecontroleerd op virussen. Pas nadat ik handmatig de scanner liet draaien, werd het virus opgemerkt. Ik heb toen besloten om de pilotdeel nog een keer te itereren.

Testen na 2^e iteratie

Bij de tweede iteratie kwam ik met de oplossing om de user altijd ingelogd te laten, maar de server wel te vergrendelen (“locken”) indien de user geen taken hoeft uit te voeren op de server. De server bevond zich dan alsnog in een beveiligde toestand. Voor het testen heb ik wederom dezelfde testmethode gebruikt als bij de eerste en de twee oplossing. Verder heb ik het testscenario behorende bij de eerste iteratie overgenomen. Dit scenario heb ik enigszins aangepast, omdat de user nu niet meer hoeft uit te loggen. In plaats daarvan vergrendelt hij de server. Het resultaat van deze test was nu wel gewenst. Nadat de user de server had vergrendeld, werd er een virus naar de server gekopieerd. Één minuut later begon de AVG virusscanner te scannen. Toen virusscanner klaar was met zijn geplande taak, heb ik in de logfile gekeken naar het resultaat. De “EICAR_test.com” virus was opgemerkt en werd automatisch verwijderd. Figuur 17 geeft u een zicht in het behaalde resultaat van de AVG virusscanner. De draaiende processen heb ik weggehaald in verband met de beveiliging van de server.



Figuur 17 – Logfile van AVG Virusscanner

6.1.3.3 Double Image

Het programma “Double Image” heb ik getest met de “black-box” functionaliteit testmethode. Hierbij heb ik gekeken naar de functionaliteit van het programma zonder daarbij iets af te weten van de code van het programma. Het ging mij puur om of het programma daadwerkelijk deed wat ik ervan verwachtte, namelijk dat het automatisch back-ups ging maken van vacatures en van de database.

Voor deze test heb ik twee scenario's bedacht: een scenario waarbij een user is ingelogd op de server en een scenario waarbij een user niet is ingelogd. De taak voor het uitvoeren van back-ups zijn in allebei de scenario's opgenomen als een geplande taak in de “Windows scheduler”. Hierbij verwijs ik terug naar paragraaf 6.1.2.5 voor meer informatie. Ik heb echter voor beide geplande taken, de tijd van uitvoeren moeten veranderen van 04.00u in de ochtend in een tijdstip waarop ik kon zien dat er een back-up gemaakt werd. Zo heb ik de tijd steeds 1 minuut later gezet voor aanvang van de test. Ik hoefde dan niet steeds lang te wachten voordat de “Windows scheduler” de taak ging uitvoeren. Voor de test heb ik de oude vacatures en de oude MS Acces database gebruikt als data waarvan een back-up gemaakt moest worden. De echte data kon op dat moment nog niet worden gebruikt, want dat was er nog niet. Maar dat maakt in feite niets uit, omdat het puur gaat om het kopiëren van bestanden. In het eerste scenario was het resultaat gewenst. De “Windows scheduler” voerde de geautomatiseerde back-up uit. In het tweede scenario echter, voerde de “Windows scheduler” de taak niet uit. Er gebeurde helemaal niets. Ik heb toen besloten om dit pilotdeel te itereren en te achterhalen wat het probleem zou kunnen zijn.

Testen van iteratie 1 ‘Double Image’

Na de eerste iteratie van dit pilotdeel, heb ik dezelfde “black-box” functionaliteit testmethode gebruikt om te kijken of de gewenste functionaliteit nu wel wordt uitgevoerd. Ook heb ik dezelfde scenario's aangehouden als bij de eerste test, omdat de resultaten anders misschien door iets anders beïnvloed konden worden. In vergelijking met de eerste oplossing, heb ik nu een batchbestand gemaakt dat ervoor moet zorgen dat de back-up taak automatisch wordt uitgevoerd. Voor een gedetailleerd overzicht van dit batchbestand verwijs ik u naar paragraaf 6.1.2.5. De resultaten voor beide scenario's zijn allebei gewenst: het back-uppen van data gebeurt automatisch. Met deze resultaten van de “black-box” functionaliteit testmethode, heb ik dit pilotdeel met succes kunnen valideren: het pilotdeel sluit aan op de vastgelegde functionaliteits “het automatisch back-uppen van vacatures en van de database”.

6.1.3.4 Remote Desktop

Het programma “Remote Desktop” heb ik ook getest op de functionaliteit: maakt het programma een verbinding met de server en dient er ingelogd te worden. Dit heb ik getest met behulp van de “black-box” functionaliteit testmethode. Voor deze tests had ik twee scenario’s geschreven. In het eerste scenario was het de bedoeling om een verbinding te maken met de server en in te loggen waarbij de server geen geautomatiseerde taken uitvoerde. In het twee scenario was het de bedoeling om een verbinding te maken met de server en in te loggen waarbij de server wel een geautomatiseerde taak uitvoert, namelijk het back-uppen van vacatures en de database. Mijn verwachtingen waren dat in allebei de situaties met succes een verbinding gelegd kon worden en dat de server op afstand bestuurbaar en instelbaar was. Daarnaast wilde ik voor beide situaties nagaan of de veiligheid van de server gewaarborgd bleef. Dit wilde ik nagaan door in beide scenario’s de volgende stappen op te nemen: de server “locken”, de verbinding verbreken met de server, vervolgens weer inloggen en nagaan of de computer nog steeds gelocked is. Het locken van de server was nodig, omdat anders de virusscanner niet zou werken in een uitgelogde toestand. Voor meer informatie hierover verwijs ik u terug naar paragraaf 6.1.2.4 en paragraaf 6.1.2.6. Uit het resultaat bleek het volgende. Het programma “Remote Desktop” kon in beide scenario’s met succes een verbinding maken met de server. Het locken van de server ging voor beide scenario’s niet goed. Na 15 minuten werd de locktoestand automatisch opgeheven en bevond de server zich in een onvergrendelde toestand. Ik heb toen besloten om dit pilotdeel te itereren.

Testen van iteratie 1 ‘Remote Desktop’

Voor het testen van de eerste iteratie heb ik dezelfde scenario’s als bij de eerste test aangehouden. Het verschil in de oplossing ligt op het gebied van de beveiliging. Ik heb nu in de “local security policy” van de Windows 2000 server de optie “Amount of Idle Time required before suspending session” op 0 minuten gezet. Dit betekent dat de server voor een onbepaalde tijd in een bepaalde sessie blijft. Voorheen stond deze optie op 15 minuten waardoor een inactieve sessie na 15 minuten automatisch beëindigd zou worden. Ik verwachtte nu dat de locktoestand nu wel door de server gehandhaafd bleef. Bij beide scenario’s bij de stappen betreffende het locken, bleef de server nu wel in een lock toestand. Met dit resultaat heb ik dit pilotdeel met succes kunnen valideren: het programma “Remote Desktop” maakt verbinding met de server en de beveiliging van de Windows 2000 server blijft gehandhaafd.

6.2 Pilot database

6.2.1 Pilotontwerp

Voor de ontwikkeling van de pilot heb ik een pilotonwikkelplan geschreven. In dit pilotonwikkelplan heb ik aangegeven in welke pilotdelen ik de pilot “database” heb opgedeeld. Dit zijn de volgende pilotdelen: installatie en configuratie van MySQL, het overzetten van data, de website arbeidsmarkt.com gereed maken voor MySQL. Ik heb de pilot in deze delen opgesplitst, omdat deze indeling mij een duidelijk beeld gaf van welke stappen er gemaakt moesten worden. Naast de duidelijkheid die het mij verschafte, waren de pilotdelen ook beter beheersbaar omdat ze een compacte vorm hadden aangenomen. Aan de hand van de onderlinge afhankelijkheden en samenhang, heb ik de pilotdelen gerangschikt naar prioriteit. Figuur 18 geeft u een overzicht van deze rangschikking van pilotdelen.

Prioriteit	Pilotdeel
1	Installatie van MySQL
2	Configuratie van MySQL
3	Het overzetten van data
4	De website arbeidsmarkt.com gereed maken voor MySQL
5	De tabellen voor de software agent toevoegen aan de MySQL database

Figuur 18 – Pilotdelen van de pilot “Database” en hun prioriteiten

6.2.2 Bouw

6.2.2.1 Pilotdeel “Installatie en configuratie van MySQL database”

6.2.2.1.1 De installatie van MySQL 4.1

Bij het eerste pilotdeel heb ik me bezig gehouden met de installatie en de configuratie de van MySQL server. Voordat ik daadwerkelijk begon met de installatie en configuratie, heb ik eerst de benodigdheden voor het configureren en beheren van de MySQL database gedetailleerder gespecificeerd. Waar het mij om ging, was dat het configureren en beheren van de MySQL database commando gestuurd is. Het zou veel handiger zijn voor zowel mij als voor mijn opdrachtgever om een duidelijk gevisualiseerde overzicht te hebben van de database en de server. Om erachter te komen of er zulke programma’s in omloop waren, heb ik de officiële site van MySQL bezocht. Op deze site heb ik systematisch gezocht naar links die verwijzen naar “products”. Toen ik de link vond, kreeg ik een lijst producten te zien waarbij de volgende twee programma’s ook op stonden: MySQL Control Center en MySQL Administrator³. Beiden programma’s geven een visueel beeld van de database en de databaseserver. Het programma MySQL Control Center is voor het beheren van de database en het programma MySQL Administrator is voor het beheren en configureren van de databaseserver. Ik heb daarom besloten om deze programma’s ook mee te nemen bij de

³ <http://www.mysql.com/products/>

installatie van MySQL, omdat dat veel makkelijker zal werken dan steeds maar SQL commando's in te voeren.

Bij de MySQL database zelf stuitte ik echter op een probleem. Er bleken verschillende versies in omloop te zijn. Dit zijn de versies: MySQL 4.0 production release, MySQL 4.1 alpha release for development en MySQL 5.0 for development and testing. Omdat ik er zeker van wilde zijn dat ik de juiste versie ga gebruiken en uiteindelijk ga implementeren, heb ik gekozen om de MySQL 4.0 production release te gaan gebruiken. Niet alleen wordt deze versie door MySQL zelf aanbevolen, maar ik heb dit ook nog onderzocht door via google.nl te zoeken naar recensies over de versies 4.0, 4.1 en 5.0. MySQL 5.0 heb ik eigenlijk buiten beschouwing gelaten bij mijn overweging, omdat ik 5.0 puur voor testen en ontwikkelen is, terwijl de website arbeidsmarkt.com productiegericht is. Ik heb gezocht met de termen "MySQL 4.0 4.1" en kwam uiteindelijk op de site van "Dev Shed Forums" en "P2P.WROX.COM" terecht⁴. Op beiden sites werd versie 4.0 van MySQL aanbevolen omdat dit de meest stabiele versie is. Versie 4.1 was volgens velen niet stabiel genoeg. Echter, het had wel nieuwe SQL statements erbij gekregen die MySQL 4.0 nog niet had, namelijk de SQL statements voor subqueries. Met de informatie die ik had verkregen tijdens het inventariseren van de webpagina's van arbeidsmarkt.com (zie paragraaf 5.3), wist ik dat er in de code van de webpagina's gebruik werd gemaakt van subqueries. Indien MySQL 4.1 gebruikt zou worden, dan behoefde veel webpagina's geen aanpassing op het gebied van queries. Echter, de reden dat ik MySQL had voorgesteld voor het gebruiken als de nieuwe database, was dat de database stabiel was. Ik moest hierover dus een beslissing maken: is stabiliteit belangrijker dan het gemakkelijk omzetten van de website?

Gezien het feit dat de site arbeidsmarkt.com op commercieel productieniveau zit, kan ik het me niet veroorloven dat de website onstabiel opereert. Het uitvallen van de database server zou klanten kunnen kosten en daarmee uiteindelijk ook inkomsten. Ik heb daarom gekozen voor MySQL 4.0. Dit heb ik aan de opdrachtgever voorgelegd. Echter, Albert was een voorstander van MySQL 4.1, omdat hij ervan uit ging dat MySQL 4.1 uiteindelijk toch wel de productieversie zou gaan worden. Bovendien vond hij de subquery een onmisbaar SQL statement. Ik heb hem gewezen op de gevaren die versie 4.1 met zich mee brengt. Door de overtuiging dat nieuwe technologieën de toekomst zijn en het vooruit streven naar nieuwe technologieën, kon ik hem niet overtuigen om de productieversie van MySQL te nemen als database server. Ik heb uiteindelijk de MySQL database server versie 4.1 gedownload en geïnstalleerd.

⁴ <http://forums.devshed.com/archive/t-99688>, http://p2p.wrox.com/topic.asp?TOPIC_ID=6735

6.2.2.1.2 Het gereed maken van de database

Voor het gereed maken van de database, moest ik eerst in de “command prompt” omgeving de database server opstarten. Dit was nodig omdat de MySQL database server dan de nodige mappen en bestanden aanmaakt zodat de database daadwerkelijk gebruikt kan worden.

Figuur 19 geeft u een zicht op het gereed maken van de database.

```

C:\>mysqld-max --console
InnoDB: The first specified data file .\ibdata1 did not exist:
InnoDB: a new database to be created!
030521 18:48:29 InnoDB: Setting file .\ibdata1 size to 10 MB
InnoDB: Database physically writes the file full: wait...
030521 18:48:30 InnoDB: Log file .\ib_logfile0 did not exist:
InnoDB: Setting log file .\ib_logfile0 size to 8 MB
InnoDB: Database physically writes the file full: wait...
030521 18:48:31 InnoDB: Log file .\ib_logfile1 did not exist:
InnoDB: Setting log file .\ib_logfile1 size to 8 MB
InnoDB: Database physically writes the file full: wait...
InnoDB: Doublewrite buffer not found: creating new
InnoDB: Doublewrite buffer created
InnoDB: Creating foreign key constraint system tables
InnoDB: Foreign key constraint system tables created
030521 18:48:37 InnoDB: Started
mysqld-max: ready for connections.
Version: '4.0.12-max' socket: '' port: 3306

```

Figuur 19 – MySQL wordt voor het eerst gestart

Zoals u in figuur 19 kunt zien is de database succesvol gereed gemaakt voor verbindingen. MySQL maakt standaard twee accounts aan. Met het oog op de beveiliging wilde ik het zogeheten ghostaccount⁵ in de database verwijderen en voor het standaardaccount “root” een wachtwoord opgeven. Ik ben toen met een nieuw “command prompt” scherm ingelogd in de database en heb toen met SQL statements de hiervoor genoemde issues onder handen genomen. In figuur 20 ziet u de beginsituatie van de database vlak nadat het de benodigde mappen en bestanden heeft aangemaakt.

```

mysql> SELECT Host, User, Password FROM user;
+-----+-----+-----+
| Host      | User  | Password |
+-----+-----+-----+
| localhost | root  |          |
| %         | root  |          |
| localhost |       |          |
| %         |       |          |
+-----+-----+-----+
4 rows in set (0.00 sec)
mysql>

```

Figuur 20 – Beginsituatie van MySQL database

⁵ Een ghostaccount is een account zonder loginnaam en zonder wachtwoord

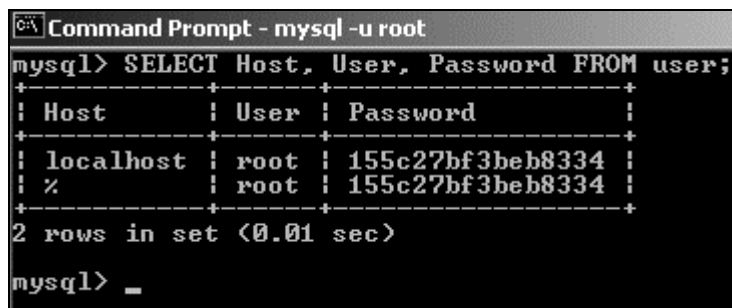
Om het ghostaccount te verwijderen heb ik de volgende query gebruikt:

- “delete from user where user= ‘?’”.

Om de user “root” een wachtwoord te geven heb ik de volgende query gebruikt:

- “update user set password=(‘****’) where user = ‘root’ “.

Vanwege bedrijfsgeheimhouding staan de sterretjes in bij de laatste query voor een wachtwoord. Als resultaat bij de query “select * from user” zie je het ghostaccount niet meer en heeft de user “root” een wachtwoord. Figuur 21 geeft het behaalde resultaat van deze acties weer.



```

C:\> Command Prompt - mysql -u root
mysql> SELECT Host, User, Password FROM user;
+-----+-----+-----+
| Host      | User  | Password      |
+-----+-----+-----+
| localhost | root  | 155c27bf3beb8334 |
| %         | root  | 155c27bf3beb8334 |
+-----+-----+-----+
2 rows in set (0.01 sec)

mysql> _

```

Figuur 21 – Het ghostaccount is verwijderd en de user root heeft een wachtwoord

6.2.2.1.3 MySQL instellen als een Windows service

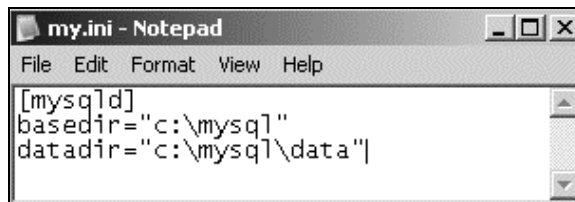
Het was voor mij nu zaak om de database server als een Windows service te laten draaien. Dit houdt in dat de database automatisch opstart wanneer de server wordt opgestart. Bij het configureren van de accounts, maakte ik gebruik van de console van MySQL met behulp van de commando: “c:\mysqld-max –console”. Om de MySQL database als een Windows service te laten draaien, heb ik hetzelfde commando gebruikt maar met een andere parameter: “c:\mysqld-max –install”. Om na te gaan of de MySQL database daadwerkelijk als een Windows service geïnstalleerd was, heb ik de lijst van Windows services in het administratieve gedeelte van de Windows 2000 server geraadpleegd. MySQL kwam daadwerkelijk in de lijst voor als een service. Echter, toen ik de service wilde opstarten, kreeg ik de volgende foutmelding:

“Could not start MySQL Service on Local Computer.
Error 1067: Process Terminates Unexpectedly.”.

Ik heb toen via google.nl gezocht naar mogelijke oplossingen met deze foutmelding als zoektermen. Als resultaat kreeg ik de site van “Dev Shed Forums”⁶. Op deze website vond ik de oplossing waarin stond dat het bestand “my.ini” aangemaakt moest zijn eer de server correct kan starten als een Windows service. Echter, ik ging ervan uit dat dit bestand automatisch aangemaakt werd, nadat de MySQL database voor het eerst was opgestart. Bij het zoeken naar dit bestand op de server, bleek dat dit bestand niet op de server aanwezig was. Achteraf heb ik dus alsnog een bestand aangemaakt met de naam “my.ini” en dit

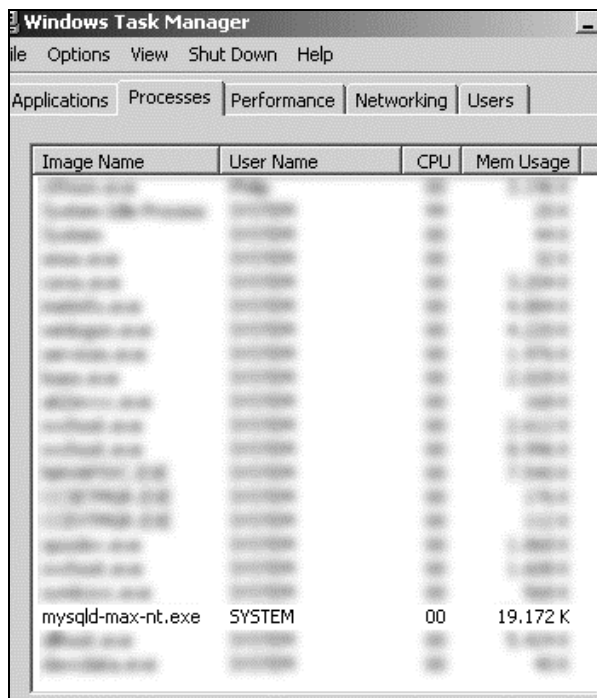
⁶ <http://forums.devshed.com/archive/t-145498>

bestand geplaatst in de root van de directory van de Windows 2000 zoals is uitgelegd op de website “Dev Shed Forums”. Figuur 22 geeft u een beeld van dit “my.ini” bestand.



Figuur 22 – Het opstartbestand My.ini

Om dit gelijk te testen heb ik de oude MySQL Windows service eerst verwijderd en daarna weer opnieuw geïnstalleerd. Bij het opstarten van deze service kreeg ik dit keer geen foutmelding. Om na te gaan of de service daadwerkelijk gestart was, heb ik de “Windows Task Manager” geraadpleegd. Hiermee kon ik precies zien welke processen en services geladen waren op de Windows 2000 server. Met behulp van de “Windows Task Manager” kon ik met succes herleiden dat de MySQL database server als een Windows service was gestart.



Figuur 23 – MySQL database server als een Windows server geladen

6.2.2.1.4 MySQL database server optimaal configureren

Tijdens het opstellen van een systeemconcept met betrekking tot de database, heb ik veel informatie kunnen vergaren omtrent het configureren van de MySQL database server. Uit die informatie heb ik een geoptimaliseerde configuratie gevonden dat geschikt is voor een server met aan 256mb aan ram geheugen. De configuratie was een zogenaamde “my-medium configuratie”. Voor meer informatie hierover verwijst ik u terug naar paragraaf 5.6.3.3. In figuur 24 ziet u hoe het “my.ini” bestand uit figuur 22 is uitgebreid met de extra instellingen.



```
[[client]
#password=*****
port=3306

[mysqld]
port=3306
skip-locking
set-variable      = key_buffer=16K
set-variable      = max_allowed_packet=1M
set-variable      = thread_stack=64K
set-variable      = table_cache=4
set-variable      = sort_buffer=64K
set-variable      = net_buffer_length=2K
server-id         = 1

# Uncomment the following if you want to log
updates
#log-bin

basedir="c:\mysql"
datadir="c:\mysql\data"

[mysqldump]
quick
set-variable      = max_allowed_packet=16M

[mysql]
no-auto-rehash

[myisamchk]
set-variable      = key_buffer=8M
set-variable      = sort_buffer=8M
```

Figuur 24 – Het bestand my.ini geoptimaliseerd volgens het zogeheten “my-medium” configuratie

Of deze instellingen gingen werken wist ik niet 100% procent zeker, omdat de server daarop nog niet was getest. Een performance test van de MySQL database server in combinatie met de Windows 2000 server, de webserver en de Coldfusion server was daarom noodzakelijk om te controleren of de server aan de performance eisen kon voldoen. Waarom ik gekozen heb om de MySQL database server in combinatie met de andere opgenoemde servers te testen, heb ik al eerder aanbod laten komen. Zie paragraaf 6.1.3.1 voor meer informatie hierover. In paragraaf 6.2.3. zal ik u nader uitleggen hoe ik de MySQL database server heb getest op performance.

6.2.2.2 Pilotdeel “Het overzetten van data”

6.2.2.2.1 Het schrijven van implementatiescripts

Het inventariseren van de tabellen had ik al gedaan tijdens de definitiestudie. In het systeemconcept had ik de tabellen correct in kaart gebracht met behulp van een EER model, een relationeel model en een implementatiemodel. Tevens had ik de tabellen uit de MS Acces database nog gecontroleerd of ze goed genormaliseerd waren. Ik kon hierbij aannemen, dat ik dit niet verder behoefde te specificeren. Om dit pilotdeel te kunnen verwezenlijken heb ik me bezig gehouden met de grote vraag. Die vraag was namelijk: hoe kreeg ik de data uit de MS Acces database getransfereerd naar de MySQL database?

Om het me zelf tijd te besparen, heb ik eerst op het Internet gezocht naar gratis programma's die de data uit een MS Acces database kon transfereren naar een MySQL database. Ik heb toen via google.nl gezocht naar dergelijke programma's met de zoektermen “convert ms access mysql”. Helaas leverde deze zoekactie mij geen bruikbare resultaten op. Veel links die als resultaat werden opgegeven, verwezen naar programma's die ofwel shareware waren of naar programma's die een zeer korte testtijd hadden. Ik heb toen mijn zoekactie aangepast: “download free convert ms acces mysql –trial -shareware”. Ook dit leverde mij geen bruikbare resultaten op. Ik heb toen besloten om op de officiële site van MySQL te zoeken naar conversieprogramma's. Ik heb toen de link “products” en vervolgens “downloads” gevolgd. Maar ook hier stonden geen conversieprogramma's, maar wel nog een link “contributed software”⁷. Ik heb toen deze link gevolgd met de veronderstelling dat er een lijst zou verschijnen van software ontwikkeld door “Third Parties”. Op deze pagina heb ik het programma “ATAF-CON” gevonden. Met dit programma was het mogelijk om implementatiescripts te genereren waarmee je de tabellen uit de MS Acces database kan dupliceren in de MySQL database. Figuur 25 geeft u een overzicht van de tabel “Vacature” in de MS Acces database.

	Veldnaam	Gegevenstype
PK	ID	AutoNummering
	bedrijf_ID	Numeriek
	Lokatie_ID	Numeriek
	Functie_ID	Numeriek
	VacTitel	Tekst
	Functieomschrijving	Memo
	Betreft_stage	Numeriek
	Contactpersoon_informatie	Tekst
	cperinfo_email	Tekst
	telefoonnummer_informatie	Tekst
	contactpersoon_sollicitatie	Tekst
	cpersoll_email	Tekst
	reageren_voor	Datum/tijd
	tonen_vanaf	Datum/tijd
	Tonen_tm	Datum/tijd

Figuur 25 – Tabel “Vacature” in de MS Acces database

Veel gegevenstypen uit de MS Acces database, worden in MySQL anders genoemd. Sommigen gegevenstypen worden in MySQL zelfs opgesplitst in specifiekere gegevenstypen. De tupel “bedrijf_ID” is in de Acces database een numeriek gegevenstype van het type “Integer”. In MySQL heb je voor het gegevenstype “Integer” de volgende specifiekere

⁷ <http://solutions.mysql.com/software/>

gegevenstypen: tinyint, smallint, mediumint en bigint. MySQL specificeert hiermee de grote van het getal. In figuur 26 ziet u het gegenereerde implementatiemodel van de tabel “Vacature”. Zoals u kunt zien is het gegevenstype “TEKST” uit de Acces database omgezet naar het type “VARCHAR” in MySQL. Het type “memo” is omgezet in “LONGTEXT” en het type “Datum/Tijd” is omgezet naar het type “DATE”. Verder zijn alle numerieke typen, omgezet naar een gespecificeerde “Integer” in MySQL.

```
CREATE TABLE Vacature (
    ID BIGINT auto_increment NOT NULL,
    Bedrijf_ID BIGINT NULL DEFAULT 0,
    Lokatie_ID SMALLINT NULL DEFAULT 0,
    Functie_ID BIGINT NULL DEFAULT 0,
    VacTitel VARCHAR(100) NULL,
    Functieomschrijving LONGTEXT NULL,
    Betreft_stage TINYINT NULL DEFAULT 0,
    Contactpersoon_informatie VARCHAR(100) NULL,
    Cperinfo_email VARCHAR(100) NULL,
    Telefoonnummer_informatie VARCHAR(100) NULL,
    Contactpersoon_sollicitatie VARCHAR(100) NULL,
    Cpersoll_email VARCHAR(100) NULL,
    Reageren_voor DATE NULL,
    Tonen_vanaf DATE NULL,
    Tonen_tm DATE NULL,
    KEY (Bedrijf_ID),
    KEY (Functie_ID),
    PRIMARY KEY (ID),
    KEY (Lokatie_ID)
)
```

Figuur 26 – Het implementatiemodel van de tabel “Vacature” gegenereerd door “ATAF-CONF”

Ik heb toen vervolgens het implementatiemodel aangepast conform het implementatiemodel die ik in systeemconcept heb gemaakt. Daarnaast heb ik het implementatiemodel verder geoptimaliseerd. Met het optimaliseren bedoel ik het veranderen van de grootte van sommige “VARCHAR” tupels en het aanpassen van het type “INTEGER”. Zoals u kunt zien hebben alle “VARCHAR” tupels een maximale grootte van 100 karakters. Dit neemt onnodig veel ruimte in beslag, omdat die tupels die grootte nooit zullen halen. De tupels van het type “INTEGER” die dienen als vreemde sleutel heb ik allemaal omgezet naar het type “MEDIUMINT”. Op die manier nemen deze tupels niet onnodig veel ruimte in beslag en bovendien hebben ze genoeg gereserveerde ruimte om te kunnen groeien. Andere tupels van het type “INTEGER” heb ik op “TINYINT” gezet, omdat deze tupels de oude “Ja/Nee” (het BOOLEAN-type) uit de MS Acces database waren. Zo heb ik bijvoorbeeld de tupel “Betreft_stage” omgezet naar het type “TINYINT”, omdat dit tupel slechts de waarde 1 of 0 krijgen.

In figuur 27 ziet u de veranderingen die ik heb aangebracht in het gegenereerde implementatiemodel van “ATAF-CON”. Met deze veranderingen op het gebied van grootte van de tupels en de soort gegevenstypen, heb ik het implementatiemodel geoptimaliseerd.

```
CREATE TABLE Vacature (
    ID BIGINT auto_increment NOT NULL,
    Bedrijf_ID MEDIUMINT NULL DEFAULT 0,
    Lokatie_ID MEDIUMINT NULL DEFAULT 0,
    Functie_ID MEDIUMINT NULL DEFAULT 0,
    VacTitel VARCHAR(60) NULL,
    Functieomschrijving LONGTEXT NULL,
    Betreft_stage TINYINT NULL DEFAULT 0,
    Contactpersoon_informatie VARCHAR(100) NULL,
    Cperinfo_email VARCHAR(50) NULL,
    Telefoonnummer_informatie VARCHAR(14) NULL,
    Contactpersoon_sollicitatie VARCHAR(50) NULL,
    Cpersoll_email VARCHAR(50) NULL,
    Reageren_voor DATE NULL,
    Tonen_vanaf DATE NULL,
    Tonen_tm DATE NULL,
    KEY (Bedrijf_ID),
    KEY (Functie_ID),
    PRIMARY KEY (ID),
    KEY (Lokatie_ID)
)
```

Figuur 27 – Het geoptimaliseerde implementatiemodel van de tabel “Vacature”

Het programma “ATAF-CON” genereerde in feite alleen maar implementatiemodellen. De inhoud van die tabellen werd echter niet mee gekopieerd. Indien ik dit programma zou gebruiken voor de overzetting van data, dan had ik de implementatiescripts per tabel stuk voor stuk moeten uitbreiden met het selecteren van data uit de MS Acces database om vervolgens diezelfde data toe te voegen aan de MySQL database. Op de pagina “contributed software” vond ik ook nog een ander gratis softwarepakket, waarmee je wel letterlijk de MS Acces database kon overzetten naar een MySQL database. Dit programma heet “DBManager Professional”. Zowel de tabellen als de inhoud van de tabellen konden getransfereerd worden van MS Acces naar MySQL.

Toch heb ik gekozen om het programma “ATAF-CON” te gebruiken, omdat ik op die manier meer inzicht kreeg in de implementatiescripts en zodoende ook tussentijds beter kon valideren met behulp van de gemaakte implementatiemodellen in het systeemconcept. Daarnaast had ik ook meer controle over het aanpassen van die modellen. Dit was in tegenstelling tot het programma “DBManager Professional” niet het geval. Mocht het in verband met de tijdsplanning of door andere problemen niet lukken met “ATAF-CON” dan kan ik altijd nog “DBManager Professional” gebruiken als redmiddel.

6.2.2.2.2 Het implementatiescript automatiseren

De vele scripts die ik genereerde met “ATAF-CON” heb ik vervolgens bij elkaar gezet in één groot implementatiescript. Dit heb ik gedaan omdat je anders 45 scriptpagina's apart moest uitvoeren en dat is niet handig. Bovendien is dan de kans dat je een tabel overschrijft groter, omdat je in een dergelijke situatie kans hebt om een scriptpagina meerdere keren uit te voeren. In figuur 28 ziet u een gedeelte van de grote implementatiescript. Dit gedeelte heeft betrekking op de tabel “vacature” net zoals in de figuren 25 tot en met 27.

```
<cfsetting enablecfoutputonly="yes">
<cfquery name="dropVacature" datasource="#datasource#">
    DROP TABLE IF EXISTS Vacature
</cfquery>

<cfquery name="createVacature" datasource="#datasource#">
    CREATE TABLE Vacature (
        ID BIGINT auto_increment NOT NULL,
        Bedrijf_ID MEDIUMINT NULL DEFAULT 0,
        Lokatie_ID MEDIUMINT NULL DEFAULT 0,
        Functie_ID MEDIUMINT NULL DEFAULT 0,
        VacTitel VARCHAR(60) NULL,
        Functieomschrijving LONGTEXT NULL,
        Betreft_stage TINYINT NULL DEFAULT 0,
        Contactpersoon_informatie VARCHAR(100) NULL,
        Cperinfo_email VARCHAR(50) NULL,
        Telefoonnummer_informatie VARCHAR(14) NULL,
        Contactpersoon_sollicitatie VARCHAR(50) NULL,
        Cpersoll_email VARCHAR(50) NULL,
        Reageren_voor DATE NULL,
        Tonen_vanaf DATE NULL,
        Tonen_tm DATE NULL,
        KEY (Bedrijf_ID),
        KEY (Functie_ID),
        PRIMARY KEY (ID),
        KEY (Lokatie_ID)
    )
</cfquery>
<cfsetting enablecfoutputonly="no">
```

Figuur 28 – Het implementatiemodel van de tabel “Vacature” in het grote conversiescript

Zoals u in figuur 31 kunt zien, heb ik nog meer SQL codering erbij geschreven. De regel met het SQL statement “DROP TABLE” houdt in dat indien de tabel al bestaat deze eerst wordt verwijderd voordat dezelfde tabel opnieuw wordt aangemaakt. Dit heb ik zo gedaan, omdat het script uiteindelijk als een geautomatiseerd script moet gaan functioneren bij de implementatie. Door dit statement eraan toe te voegen, hoef ik de database vóór de conversie niet te controleren of alle tabellen wel leeg zijn. Bij het implementeren worden namelijk alle data uit de MS Acces database in één keer overgezet naar de MySQL database. Door de tabellen eerste te verwijderen en daarna opnieuw aan te maken in het conversiescript, heb ik de mogelijkheid dat er data meerdere keren kan voorkomen in de MySQL database weggenomen.

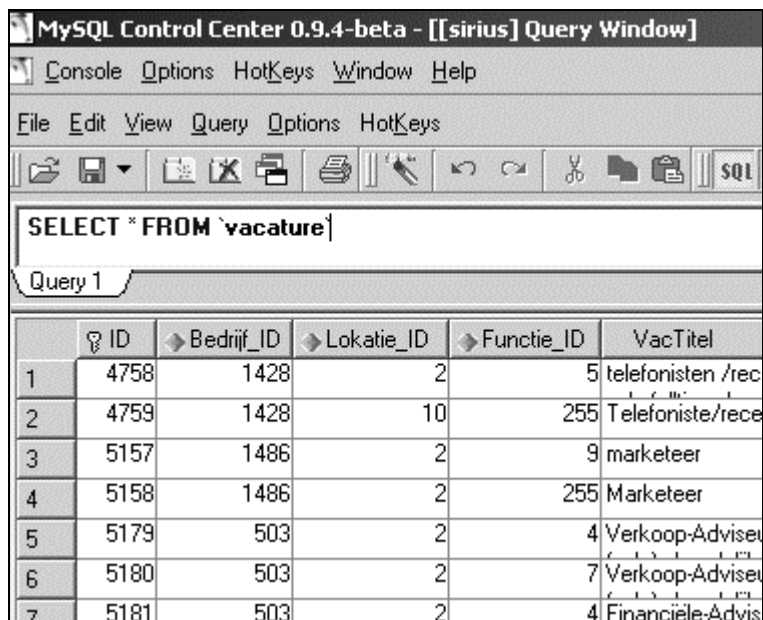
6.2.2.2.3 Het schrijven van scripts voor de dataoverdracht

Omdat er in deze situatie sprake is van twee verschillende typen databases, moest ik ook rekening houden met twee verschillende typen DSN's. MS Acces wordt normaliter benaderd via de ODBC van Microsoft. MySQL echter maakt gebruik van "MySQL ODBC". In figuur 31 ziet u ook de variabele "#DATASOURCE#" staan. Deze variabele houdt de waarde van de DSN van MySQL vast. De andere variabele "#DATASOURCE2#", die overigens niet in de figuur te zien is, houdt de waarde van de DSN van MS Acces vast. Door de waarden van de twee verschillende DSN's vast te houden in variabelen, kon ik met twee verschillende soorten databases tegelijkertijd verbinding maken. Om ervoor te zorgen dat de data kan worden overgezet van de MS Acces database naar MySQL database, moest ik ook een "SELECT" statement opgeven die de data uit MS Acces database selecteerde. Als laatste heb ik gebruik gemaakt van de "INSERT INTO" statement waarmee ik de geselecteerde data uit de MS Acces database invoer in de MySQL database. In figuur 29 kunt u zien hoe de data wordt ingevoerd in de MySQL database.

```
<cfloop query="getVacature">
<cfquery name="insertVacature" datasource="#datasource#">
INSERT INTO Vacature (
ID,
Bedrijf_ID,
Lokatie_ID,
Functie_ID,
VacTitel,
Functieomschrijving,
Betreft_stage,
Contactpersoon_informatie,
Cperinfo_email,
Telefoonnummer_informatie,
Contactpersoon_sollicitatie,
Cpersoll_email,
Reageren_voor,
Tonen_vanaf,
Tonen_tm
)
VALUES
(
#ID#,
<cfif Bedrijf_ID NEQ "">#Bedrijf_ID#<cfelse>NULL</cfif>,
<cfif Lokatie_ID NEQ "">#Lokatie_ID#<cfelse>NULL</cfif>,
<cfif Functie_ID NEQ "">#Functie_ID#<cfelse>NULL</cfif>,
<cfif VacTitel NEQ "">#VacTitel#<cfelse>NULL</cfif>,
<cfif Functieomschrijving NEQ "">#Functieomschrijving#<cfelse>NULL</cfif>,
<cfif Betreft_stage NEQ "">#Betreft_stage#<cfelse>0</cfif>,
<cfif Contactpersoon_informatie NEQ
"">#Contactpersoon_informatie#<cfelse>NULL</cfif>,
<cfif Cperinfo_email NEQ "">#Cperinfo_email#<cfelse>NULL</cfif>,
<cfif Telefoonnummer_informatie NEQ
"">#Telefoonnummer_informatie#<cfelse>NULL</cfif>,
<cfif Contactpersoon_sollicitatie NEQ
"">#Contactpersoon_sollicitatie#<cfelse>NULL</cfif>,
<cfif Cpersoll_email NEQ "">#Cpersoll_email#<cfelse>NULL</cfif>,
<cfif Reageren_voor NEQ "">#Reageren_voor#<cfelse>NULL</cfif>,
<cfif Tonen_vanaf NEQ "">#Tonen_vanaf#<cfelse>NULL</cfif>,
<cfif Tonen_tm NEQ "">#Tonen_tm#<cfelse>NULL</cfif>
)
</cfquery>
</cfloop>
```

Figuur 29 – Data uit MS Acces database wordt ingevoerd in de MySQL database

Velden die geen waarden bevatten, heb ik met een “NULL”-waarde in de MySQL database ingevoerd. Dit heb ik gedaan om ervoor te zorgen dat de database niet meteen vervuld werd met velden die slechts een paar spaties als waarden hadden. Dit zou de omvang database onnodig groot houden. In figuur 30 ziet u het resultaat het implementatiescript. Met het programma MySQL Control Center, heb ik gekeken of de data daadwerkelijk in de MySQL database was overgezet. Met de statemenet `SELECT * FROM vacature`, kreeg ik alle data te zien die ook in de MS Acces database aanwezig was. Dit heb ik gecontroleerd door het aantal records na te gaan met behulp van de functie “Recordcount”.



The screenshot shows the MySQL Control Center interface. The query window displays the query `SELECT * FROM `vacature``. Below the query, the results are shown in a table with 7 rows and 6 columns: ID, Bedrijf_ID, Lokatie_ID, Functie_ID, and VacTitel. The data is as follows:

	ID	Bedrijf_ID	Lokatie_ID	Functie_ID	VacTitel
1	4758	1428	2	5	telefonisten /rec
2	4759	1428	10	255	Telefoniste/rece
3	5157	1486	2	9	marketeer
4	5158	1486	2	255	Marketeer
5	5179	503	2	4	Verkoop-Adviseu
6	5180	503	2	7	Verkoop-Adviseu
7	5181	503	2	4	Financiële-Adviseu

Figuur 30 – Het resultaat van de gegevensoverdracht van MS Acces naar MySQL

Iteratie 1

Tijdens het valideren van het implementatiescript heb ik een fout opgemerkt op de website. De datum werd namelijk niet goed weergegeven. De maanden en de dagen waren namelijk omgedraaid. Ik had toen besloten om voor alle velden met “DATE” als gegevenstype te veranderen in een numerieke waarde met een maximale waarde van 8 karakters. De belangrijkste reden voor deze oplossing is nog wel dat je zelf de opmaak van de datum in handen hebt en niet afhankelijk bent van instellingen van een bepaalde server. Daarnaast is de invoer van de datum als numerieke waarde altijd consistent. Deze verandering betekende echter ook dat ik het gehele implementatiescript moest aanpassen met deze wijziging. In figuur 31 ziet u de wijziging ook doorgevoerd bij het implementatiemodel van tabel “Vacature”

```

<cfquery name="toevoegen" datasource="arbeidsmarkt">
INSERT INTO Vacature(ID,
    Bedrijf_id,
    Functie_id,
    VacTitel,
    functieomschrijving,
    Betreft_stage,
    lokatie_id,
    contactpersoon_informatie,
    cperinfo_email,
    telefoonnummer_informatie,
    contactpersoon_sollicitatie,
    cpersoll_email,
    reagerenvoor,
    tonenvanaf,
    tonentm)
VALUES (#VacsMaxIDacomstats#,
    #form.bedrijf_id#,
    '#form.functie_id#',
    '#form.VacTitel#',
    '#form.functieomschrijving#',
    #betreft_stage#,
    #plaatsid#,
    '#form.contactpersoon_informatie#',
    '#form.cperinfo_email#',
    '#form.telefoonnummer_informatie#',
    '#form.contactpersoon_informatie#',
    '#form.cperinfo_email#',
    '#tmpReageren_voor#',
    '#tmpTonen_vanaf#',
    '#tmpReageren_voor#')
</cfquery>

```

Figuur 31 – Wijziging in het implementatiemodel van tabel “Vacature”

In vergelijking met het implementatiemodel van de tabel “Vacature” in figuur 28 heten de drie tupels met als gegevenstype “DATE” nu anders. Deze velden heten nu: “reagerenvoor”, “tonenvanaf” en “tonentm”. Ik heb in de database voor deze tabel namelijk drie nieuwe tupels aangemaakt met “INT” als gegevenstype. De oude tupels heb ik nog wel in de database laten staan, indien het niet goed zou gaan met de nieuwe tupels. Echter, de ruimte die in beslag werd genomen door de drie oude tupels was maar van tijdelijke duur omdat het gebruik van de nieuwe tupels helemaal geen problemen opleverde. Het was dus onnodig om de oude tupels te bewaren.

6.2.2.3 Pilotdeel “Conversie website arbeidsmarkt.com”

In de definitiestudie heb ik een inventarisatie gemaakt van alle webpagina's van arbeidsmarkt.com die nog werden gebruikt. Hierbij heb ik tevens gekeken naar welke gebruik maakt van de oude MS Acces database. Die inventarisatie was bedoeld voor dit pilotdeel. Voor meer informatie over de inventarisatie verwijst ik u terug naar paragraaf 5.1.3 en 5.1.4. In de vorige paragrafen heeft u kunnen zien hoe ik de data uit de MS Acces database heb overgezet naar de MySQL database. Het was nu zover om de website arbeidsmarkt.com in orde te maken, aangezien er wel het een en ander is veranderd qua gegevenstypen in de MySQL database. Daarnaast is het ook zo dat de SQL syntax van MySQL iets anders is dan de SQL syntax van MS Acces. Het nagaan van alle webpagina's op de juistheid queries van daarom een noodzaak.

Het aanpassen van al die webpagina's heb ik op een gestructureerde manier gedaan. Mijn houvast was de inventarisatielijst uit de definitiestudie. Deze lijst heb ik gebruikt om per pagina aan te geven wat er zoal veranderd moet worden. Op die manier kon ik dit pilotdeel beter beheersen. Daarnaast kon ik sneller ingrijpen indien er een pagina verkeerd was veranderd. In feite heb ik de inventarisatielijst gebruik als een soort debuglijst. Figuur 32 geeft u een beeld van hoe mijn debuglijst eruit zag.

<u>Vacature_publish.cfm</u>	Reagerenvoor, reagerentm aangepast op numerieke datum. Publish naar e -> c
<u>Vacature_publish2.cfm</u>	Idem dito
<u>Vacature_publish_loop.cfm</u>	Publish naar e -> c <pre> <cfif result.reagerenvoor neq 0> <cfset inputDatum =#result.reagerenvoor#> <cfset tmpjaar mid(inputDatum#,1,4)> <cfset tmpmaand mid(inputDatum#,5,2)> <cfset tmpdag mid(inputDatum#,7,2)> <cfset showOutput "/" & tmpmaand & "/" & tmpjaar> <cfelse> <cfset showOutput = "> </cfif> <cfif datecompare(#showOutput#.now()) is -1> </pre>
<u>Vacature_statistiek</u>	Let op datum is hier datum+tijd. Maar aangezien er geen userinput is, kan er weinig fout gaan met datum. Dus hier gebruik maken van <u>datetime</u>
<u>Vacature_aanpassen.cfm</u>	
<u>Vacature_hoelijken.cfm</u>	

Figuur 32 – Debuglijst voor de website arbeidsmarkt.com

Ik heb mijn debuglijst structureel afgewerkt: webpagina voor webpagina. Nadat ik een webpagina had gewijzigd, heb ik deze pagina gelijk gevalideerd in de webbrowser. Dit heb ik gedaan met behulp van de “White-box” testmethode. “White-box” testen is testen aan de hand van de programmacode. Ik heb hierbij incrementeel getest en ontwikkeld wat inhoud dat ik stukje voor stukje mijn scripts heb getest en heb ontwikkeld. Indien ik een blok script niet wilde testen, heb ik dat in commentaar gezet. Zo heb ik voorkomen dat ik misschien wel heel veel foutmeldingen kreeg die allemaal met elkaar te maken konden hebben. Op deze manier kon ik al die code verifiëren op de juistheid ervan. Tegelijkertijd heb ik elke

webpagina ook kunnen valideren op het gewenste resultaat. Immers, de uitkomst moest hetzelfde zijn als bij de oude website. Tijdens het valideren merkte ik de snelheid van MySQL op. Bij de zoekpagina naar cv's duurde het slechts 3 tot 4 seconden voordat je een resultaatsscherm te zien kreeg. Bij de oude MS Acces database duurde het wel 3 tot 4 keer zo langer.

Toen ik bij webpagina's aankwam die betrekking hadden op de vacatures, merkte ik bij het valideren op dat de datum verkeerd werd weergegeven in de webbrowser. De dag en de maand waren omgedraaid. Toen ik aan collega's vroeg hoe het kwam dat de datum verkeerd werd weergegeven, werd mij verteld dat zij dat ook niet wisten. Dit probleem bleek met de oude database ook al voor te komen. Ik heb toen de datuminstelling van de server nagekeken en die stond op Nederlandse datumweergave: maand, dag en jaar. Ik had toen besloten om voor dit probleem geen oplossing te zoeken op het Internet, aangezien ze voorheen ook al te kampen hadden met dit probleem. In plaats daarvan heb ik besloten om de datum op te slaan als een numerieke waarde in de database. De reden waarom ik hiervoor heb ik gekozen is, omdat uit een numerieke waarde heel makkelijk de datum is af te leiden. Daarnaast is de invoer van de datum als numerieke waarde altijd consistent. De belangrijkste reden voor deze oplossing is nog wel dat je zelf de opmaak van de datum in handen hebt en niet afhankelijk bent van instellingen van een bepaalde server. Op die manier kan de website eventueel heel gemakkelijk verplaatst worden naar andere servers, zonder daarbij te hoeven letten op een juiste weergave van de datum. Figuren 33 en 34 geven weer hoe ik de invoer van een datum naar een numerieke waarde converteer en hoe ik de numerieke waarde in gewenste formaat laat zien in de webbrowser.

```
<cfif form.reageren_voor neq ''>
    <cfset dag = mid(#form.reageren_voor#,1,2)>
    <cfset maand = mid(#form.reageren_voor#,4,2)>
    <cfset jaar = mid(#form.reageren_voor#,7,4)>
    <cfif isDate(jaar&"/"&maand&"/"&dag)>
        <cfset tmpReageren_voor = val(trim(jaar&maand&dag))>
    <cfelse>
        <cfset tmpReageren_voor=0>
    </cfif>
<cfelse>
    <cfset tmpReageren_voor=0>
</cfif>

<cfset inputDate = '#dateformat("#Now()"#, "dd/mm/yyyy")#'>
<cfset dag = mid(#inputDate#,1,2)>
<cfset maand = mid(#inputDate#,4,2)>
<cfset jaar = mid(#inputDate#,7,4)>
<cfset tmpTonen_vanaf = '#jaar##maand##dag#>
```

Figuur 33 – Script voor de invoer van datum wordt omgezet naar een numerieke waarde

De invoer van een datum met behulp van een formulier heb ik in feite opgesplitst in drie delen: dag, maand en jaar. Voor de dag pak met de functie de eerste twee karakters van de invoer. Voor de maand pak ik het vierde en het vijfde karakter van de invoer. Het derde karakter is namelijk een “/” teken. Zo heb ik voor het jaar de laatste vier karakters gepakt. Vervolgens heb ik gekeken of de waarde een geldige waarde is door de functie “IsDate” te

gebruiken. Indien het een geldige waarde blijkt te zijn, worden de drie variabelen met alleen maar getallen samengevoegd tot één getal.

```
<cfif reagerenvoor neq 0>
  <cfset datumInput  = '#reagerenvoor# '>
  <cfset tmpjaar      = mid(#datumInput#,1,4)>
  <cfset tmpmaand     = mid(#datumInput#,5,2)>
  <cfset tmpdag       = mid(#datumInput#,7,2)>
  <cfset showDatum    = tmpdag & "/" & tmpmaand & "/" & tmpjaar>
<cfelse>
  <cfset showDatum = ''>
</cfif>
```

Figuur 34 – Script voor omzetten van een numerieke waarde in de juiste datumweergave

Voor de juiste weergave van de datum, heb ik een script geschreven waarbij de dag, de maand en het jaar uit het opgeslagen getal worden extraheert en vervolgens in een variabele worden gezet. Voor het jaar worden de eerste vier karakters gepakt. Voor de maand het vijfde en het zesde karakter en voor de dag de laatste twee karakters. Vervolgens heb ik deze drie variabelen samengevoegd met een “/” ertussen. Op deze manier heb ik de datum correct kunnen weergeven zonder daarbij afhankelijk te zijn van de datum- en tijdstellingen van de server.

Deze wijziging in gegevenstype van datum naar numeriek, betekende echter wel dat ik het implementatiemodel van de tabellen moest aanpassen. Alle tupels in de database die “DATE” als type hadden, moesten veranderd worden in “SMALLINT” met een grootte van maximaal 8 karakters. Ik heb daarom het pilotdeel “Het overzetten van data” moeten itereren. Voor meer informatie hierover verwijst ik u terug naar paragraaf 6.2.2.2.

Die iteratie heeft heel veel tijd gekost, omdat het veel meer werk was dan ik tijdens het opstellen van de planning had ingeschat. Dit tijdsverlies had tot gevolg dat ik achterop raakte met de planning. Ik bevond me nog in week 9, terwijl ik eigenlijk al de pilot had moeten testen zoals in week 11. In figuur 35 ziet u in de planning waar ik eigenlijk had moeten zijn.

Week	Activiteit	Product
8 (22-26 maart)	Ontwikkelen pilot(delen) 2	
9 (29 maart – 2 april)	Ontwikkelen pilot(delen) 2	
10 (5-9 april)	Ontwikkelen pilot(delen) 2	Pilot 2
11 (12-16 april)	Testen pilot 2, Ontwikkelen pilot(delen) 3	

Figuur 35- Planning

Het testen had ik voor deze pilot eventueel kunnen overslaan om zo tijdswinst te boeken voor het ontwikkelen van de software agent. Echter, gezien vanuit het managementoptiek van dit project en gezien vanuit de noodzaak van de wijziging in datatype, kon ik geen concessies doen op deze pilot. Deze pilot was namelijk van groot belang voor zowel het bedrijf als voor de software agent. Het bedrijf zou anders inkomsten mislopen en de software agent zou niet volledig kunnen functioneren, omdat het zijn gevonden resultaten niet kwijt zou kunnen. Ik had toen besloten om de MySQL database server toch te testen op performance.

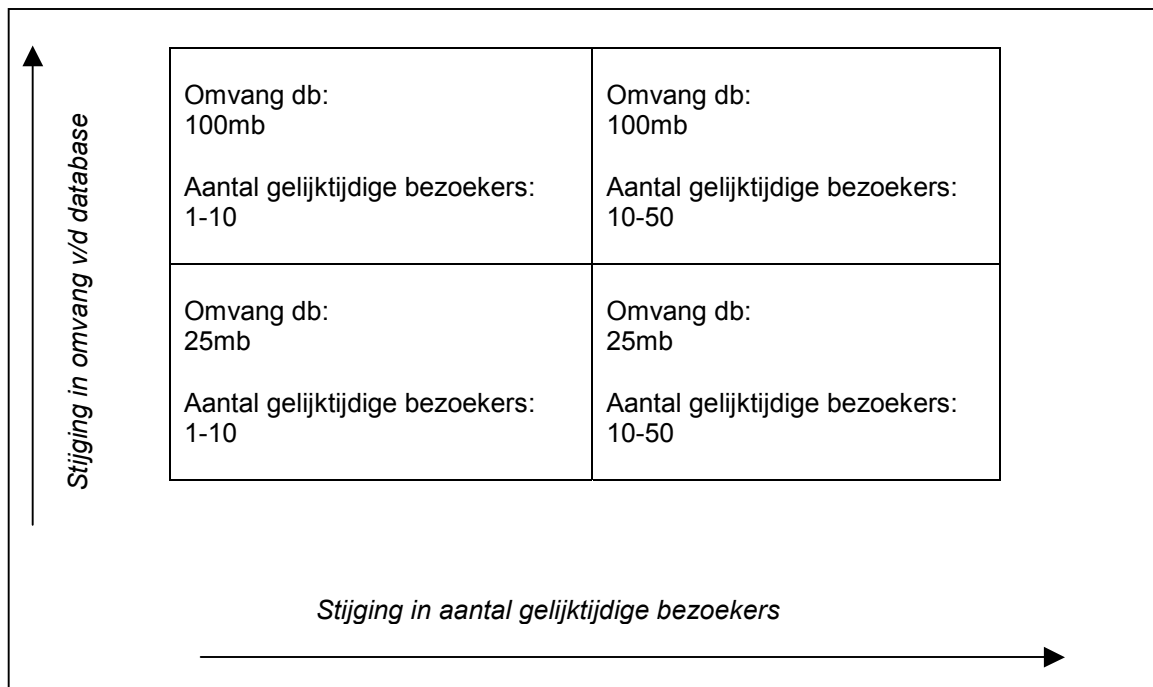
6.2.3 Beoordeling en test

Na het opgelopen tijdsverlies tijdens de eerste iteratie van pilotdeel “Het overzetten van data”, had ik toch besloten om de performance test van de MySQL database server door te zetten. Zoals ik in paragraaf 6.1.3 had beschreven wilde ik de Windows 2000 server, de webserver en de Coldfusion server niet afzonderlijk testen, omdat het uiteindelijk ging om de resultaten van die drie componenten in combinatie met de MySQL database server. De performance eisen omtrent de server hadden betrekking op de schaalbaarheid en de laadprestaties van de server. Die eisen waren de volgende:

- De database moet een grotere opslagcapaciteit aan kunnen zonder dat daarbij de response tijd exponentieel achteruit gaat.
- De server waarop de database draait moet optimaal geconfigureerd zijn voor grote bezoekersaantallen. De response tijd mag niet exponentieel achteruit gaan.

Met de eerste eis, de eis met betrekking op de schaalbaarheid van de server, wordt bedoeld dat indien de database gaat groeien, de response tijd van de website voor gebruikers hooguit evenredig moet gaan groeien. Indien de response tijd sneller zou stijgen dan de omvang van de database, dan zou er iets niet goed zijn geconfigureerd of er zou iets mis kunnen zijn met de hardware. Deze eis is dus niet opgesteld met concrete waarden, maar met verhoudingen, omdat het om de schaalbaarheid gaat. De waarden die opgemeten zullen worden dienen nu als mijlpalen voor de verhouding. De database heeft in ieder geval al winst geboekt op het gebied van snelheid in vergelijking met de oude situatie, vanwege het gebruik van de MySQL database. Voor meer informatie hoe ik dit heb bevestigd gekregen verwijs ik u terug naar paragraaf 6.2.2.3. Het gaat er bij deze eis uiteindelijk om er zeker van te zijn dat de database en de website een goede response tijd naarmate de database groeit. De tweede eis had betrekking op de laadprestatie van de server. De eis houdt in dat indien het bezoekersaantal groter zou worden, de response tijd van de website voor de gebruikers hooguit evenredig mag stijgen. Indien dit in verhouding sneller stijgt dan het aantal bezoekers, dan zou er ook hier iets niet goed zijn geconfigureerd of zou het aan de hardware kunnen liggen.

Voor het testen heb ik een viertal scenario's geschreven. In de eerste drie scenario's wilde ik de laadprestaties testen met een databaseomvang van 25mb. In de laatste drie scenario's heb ik hetzelfde gedaan als bij de eerste drie scenario's, alleen heb ik de database vier keer zo groot gemaakt. Voor de schaalbaarheid heb ik dus twee situaties aangenomen: de huidige omvang van de database en een vier keer zo'n grote omvang. Voor de laadprestatie heb ik ook twee situaties aangenomen: 1 tot 10 gelijktijdige bezoeken en 10 tot 50 gelijktijdige bezoeken. Figuur 36 geeft u een overzicht van de verschillende situaties die ik in de twee scenario's heb getest.



Figuur 36 – De verschillende situaties die ik heb gebruikt voor het testen

Voor alle scenario's heb ik hetzelfde surfpatroon gebruikt. Dit is de weg die elke gesimuleerde gebruiker moet surfen op arbeidsmarkt.com. Aangezien het zoeken naar cv's het zwaarste gedeelte van de gehele website is, heb ik de route over die webpagina's uitgestippeld. In figuur 37 ziet u het surfpatroon dat ik heb gebruikt voor alle scenario's.

Stap	Webpagina	Wat gebeurt er?
1	http:// www.arbeidsmarkt.com /vacature.cfm	De gebruiker logt in als bedrijf bij arbeidsmarkt.com
2	http:// www.arbeidsmarkt.com /zoekcv/	De gebruiker gaat zoeken naar vacatures en tikt een zoekopdracht in: ICT*
3	http://www.arbeidsmarkt.com/zoekcv/searchmain.cfm?searchstring=ICT*&search1=+zoek+	De gebruiker krijgt een resultatenpagina te zien, maar gaat verder naar de volgende resultatenpagina.
4	http://81.4.66.167/zoekcv/searchmain.cfm?searchstring=ICT*&StartRow=1	De gebruiker krijgt een resultatenpagina te zien, maar gaat verder naar de volgende resultatenpagina.
5	http://81.4.66.167/zoekcv/searchmain.cfm?searchstring=ICT*&StartRow=11	De gebruiker krijgt een resultatenpagina te zien en ziet een interessante cv
6	http://81.4.66.167/cv/3/cv25089.htm	De gebruiker bekijkt de cv

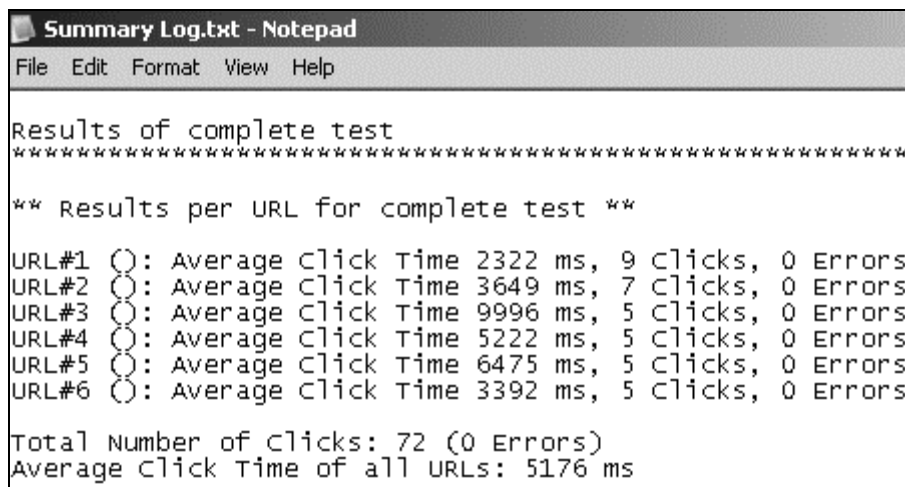
Figuur 37 – Scenario “zoeken naar cv”

Voor het simuleren van die bezoekers heb ik het gratis programma “Webserver Stress Tool 6” gebruikt. Met dit programma kan ik een “x” aantal mensen simuleren die de website gelijktijdig bezoeken. Om een representatief mogelijke uitkomst te krijgen, heb ik ook een aantal condities opgelegd voor het testen. Dit waren de volgende condities:

- Gemiddelde tijd voor vertraging tussen het klikken: 10 seconden
- Maximale tijd dat de gebruiker op de site is: 2 minuten
- Maximale snelheid waarmee de gebruiker surft: 60 Kbyte/s
- De gebruikte browser voor het surfen: MS Internet Explorer 5.5

Door deze condities constant te houden en alleen de factor “omvang v/d database” en de gesimuleerde actoren te veranderen, heb ik de schaalbaarheid en de laadprestaties van de MySQL database server in combinatie met de webserver en de Coldfusion server kunnen meten.

Bij elk scenario heb ik ook het CPU verbruik in de gemeten. Indien de CPU bij een scenario voortdurend honderd procent in beslag zou nemen, dan zou dit betekenen dat de processor het niet aan kan of dat de server niet is geconfigureerd. Indien de CPU voortdurend nauwelijks wordt gebruikt, dan zou dit betekenen dat de server niet goed is geconfigureerd. In de figuur 38 ziet u de resultaten van het eerste scenario: de database heeft een omvang van 25mb en er zijn maximaal 10 gelijktijdig gesimuleerde gebruikers.



```
Summary Log.txt - Notepad
File Edit Format View Help

Results of complete test
*****

** Results per URL for complete test **

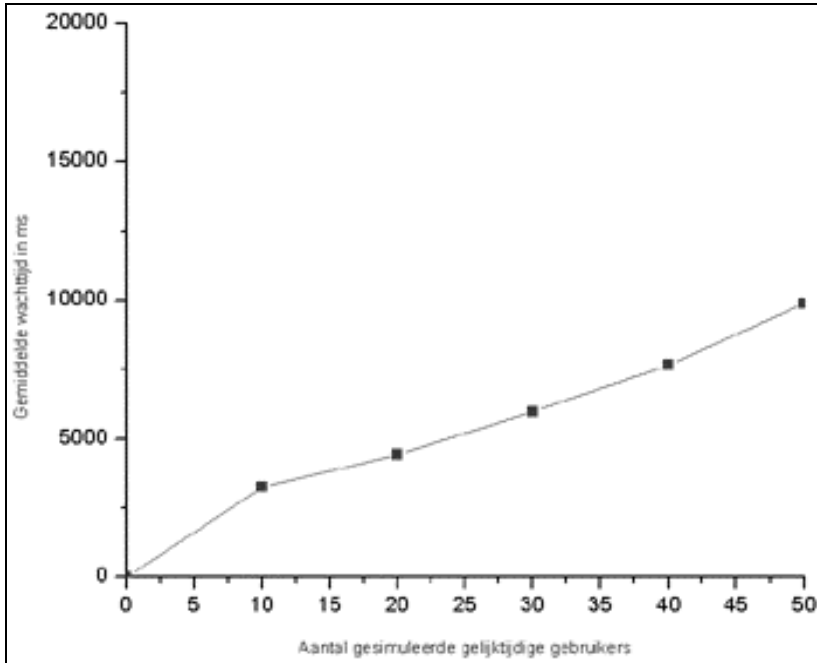
URL#1 () : Average Click Time 2322 ms, 9 Clicks, 0 Errors
URL#2 () : Average Click Time 3649 ms, 7 Clicks, 0 Errors
URL#3 () : Average Click Time 9996 ms, 5 Clicks, 0 Errors
URL#4 () : Average Click Time 5222 ms, 5 Clicks, 0 Errors
URL#5 () : Average Click Time 6475 ms, 5 Clicks, 0 Errors
URL#6 () : Average Click Time 3392 ms, 5 Clicks, 0 Errors

Total Number of Clicks: 72 (0 Errors)
Average Click Time of all URLs: 5176 ms
```

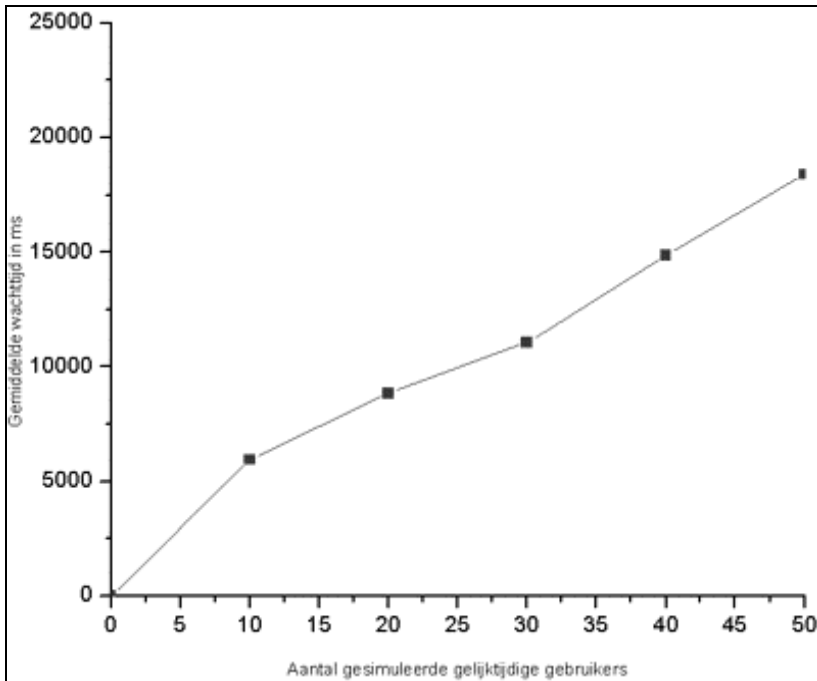
Figuur 38– Resultaten scenario 1, database omvang 25mb en 1-10 simultaan gebruikers

De term “average click time” is de gemiddelde wachttijd die al de gebruikers moesten wachten. Dit is hetzelfde als de response time waar ik het eerder over had. De gemiddelde wachttijd voor het gehele scenario is 5,1 seconden, waarbij de eerste zoekpagina, link 3, de langste wachttijd heeft van een gemiddelde van 9,9 seconden. Dit is het scenario met een database met een omvang van 25mb en een maximum van 10 gelijktijdig gesimuleerde gebruikers. Van al die zes scenario’s heb ik twee grafieken gemaakt. Voor 1 grafiek heb ik aangenomen dat de database een omvang heeft van 25mb. Voor de andere grafiek heb ik aangenomen dat de database een omvang van 100mb heeft.

Figuur 39 geeft het eindresultaat weer met een database met een omvang van 25mb. In figuur 40 ziet u het eindresultaat met een database van 100mb. Op de verticale as heb ik de gemiddelde wachttijd in milliseconden uitgezet. Op de horizontale as het aantal gesimuleerde gebruikers.



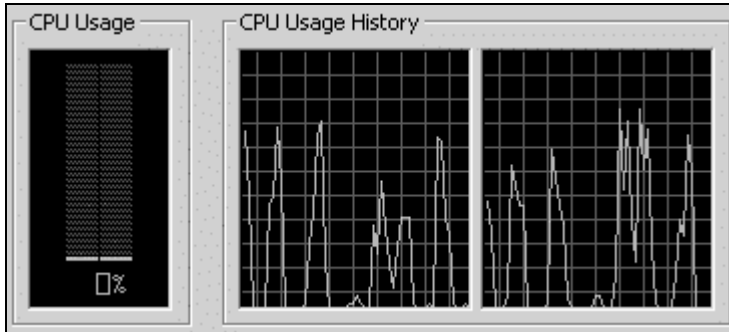
Figuur 39 – Eindresultaat met een database van 25mb



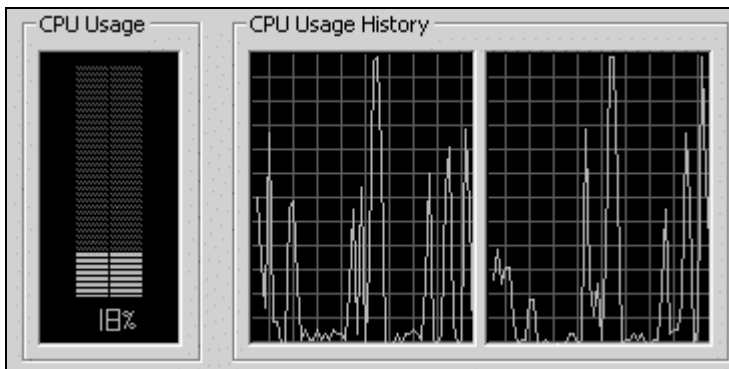
Figuur 40 – Eindresultaat met een database van 100mb

Uit de behaalde resultaten kan ik concluderen dat de “response time” niet exponentieel stijgt, indien de database vier keer zo groot wordt. Een database met een omvang van 25mb dat meer dan 26000 records bevat aan cv's, is dus nog schaalbaar. Indien de lijn wel exponentieel zou stijgen, dan zou de database een niet veel meer kunnen groeien, omdat de wachttijd extreem stijgt. Hieruit kon ik concluderen dat de webserver, de MySQL database server en de Coldfusion server goed was geconfigureerd.

Voor beiden situaties heb ik ook CPU verbruik gemeten. Figuren 41 en 42 geven u een beeld van het CPU gebruik bij een database met 25mb respectievelijk 100mb qua omvang.



Figuur 41 – Het CPU verbruik bij een database met 25mb



Figuur 42 – Het CPU verbruik bij een database met 100mb

Zoals u kunt zien heeft de server met een database met een omvang van 100mb de processor meerdere malen op 100% nodig. Echter, de server had de processor niet constant voor 100% nodig. Hieruit kon ik concluderen dat de MS Windows 2000 server goed was geconfigureerd en dat er nog geen updates nodig waren op het gebied van hardware.

6.2.4 De implementatie

Nadat ik de server had getest op de performance eisen, heb ik me beziggehouden met de implementatie van de nieuwe server. Hiervoor heb ik een kleine plan van aanpak geschreven. In dit plan van aanpak heb ik beschreven hoelang de invoering ongeveer zou gaan duren, welke resultaten deze invoering zou gaan opleveren en welke documentatie erbij geleverd zou gaan worden. De duur van de invoering zou ongeveer 20 tot 30 minuten duren. Deze informatie heb ik verkregen door de tussentijdse validaties die ik had uitgevoerd tijdens het ontwikkelen van de implementatiescripts. Als documentatie heb ik een korte handleiding geschreven over het configureren van de MySQL database server en hoe je deze kon beheren. Ik heb juist van dit pakket een korte handleiding geschreven, omdat niemand binnen het bedrijf iets af wist van MySQL. Over de website en over de Coldfusion server en de webserver had men wel kennis van. Als op te leveren product heb ik voor deze invoering het volgende product beschreven:

- Een geconfigureerd Windows 2000 server met geïnstalleerde software pakketten
 - MySQL database server
 - de website arbeidsmarkt.com
- Een kleine handleiding over het configureren en beheren van de MySQL database server

Daarnaast heb ik in het plan van aanpak ook een klein invoeringsscenario geschreven waarin ik de verschillende te ondernemen stappen heb genoteerd. Voor een goede invoering moesten namelijk eerst een aantal stappen van te voren worden genomen voordat de server daadwerkelijk ingevoerd kon worden. De eerste stap was het nakijken van de aangemaakte DSN's voor zowel de MS Acces database als de MySQL database op de nieuwe server. Indien deze nog niet aan waren gemaakt, moesten deze alsnog via het "ODBC" scherm. Als deze DSN's niet aanwezig waren, dan kon het geautomatiseerde implementatiescript niet werken. De tweede stap was kopiëren van de MS Acces database naar de nieuwe server zodat beiden databases op dezelfde server stonden. Indien ik het via het netwerk zou doen, dan zouden andere websites op de eerste server hiervan last krijgen omdat het implementatiescript veel CPU vergt. Vervolgens moest het geautomatiseerde implementatiescript worden uitgevoerd. Als vierde stap moest de overdracht worden nagekeken of daadwerkelijke alle data was getransfereerd. Als laatste stap moest de URL arbeidsmarkt.com verwijzen naar het ip adres van de nieuwe server.

De vier eerste stappen van het scenario gingen goed. Echter bij de laatste stap ging er iets niet goed. De URL arbeidsmarkt.com verwees nog steeds naar het ip adres van de oude server, terwijl de nieuwe server al "live" stond te draaien. Bij het overschakelen naar de nieuwe server kon er dus nog data in de oude database worden ingevoerd, terwijl de website al draaide op de nieuwe server. Volgens de opdrachtgever viel hier niks aan te doen omdat het wijzigen van de verwijzing van de URL minstens 24 uur zou gaan duren.

6.3 Pilot software agent

6.3.1 Het pilotontwerp

Voor de ontwikkeling van de pilot “Software Agent” heb ik een klein pilotontwikkelpun geschreven. In dit pilotontwikkelpun had ik aangegeven in welke pilotdelen ik de pilot “Software Agent” had opgedeeld. Dit waren de volgende pilotdelen: de tabellen, de configuratieschermen en de software agent. Als pilot ontwikkelstrategie had ik gekozen voor de strategie: “de buitenkant het eerst”. Deze strategie houdt in dat je zo snel mogelijk met concrete, tastbare resultaten aan komt zetten. Gezien mijn de situatie die mij te wachten stond, zouden de opdrachtgevers niet snel concrete resultaten zien, omdat het ontwikkelen van de software agent zelf alleen maar uit programmeren bestond. Daarom was het verstandiger om eerst een aantal visuele componenten te vervaardigen, zodat ze een in ieder geval een idee hadden hoe de software agent uiteindelijk zou gaan functioneren. Daarnaast was het voor mezelf ook beter om eerst de visuele componenten te ontwikkelen, omdat deze componenten vergeleken met de agent veel makkelijker waren te verwezenlijken. Daarna kon me volledig verdiepen in de nieuwe technologieën die kwamen kijken bij het maken van de software agent. Op grond van deze ontwikkelstrategie had ik een indeling gemaakt van de drie pilotdelen. Figuur 43 geeft u een overzicht van de gekozen rangschikking voor de pilotdelen.

Prioriteit	Pilotdeel
1	De tabellen
2	De configuratieschermen
3	De software agent

Figuur 43 – Pilotdelen van de pilot “Database” en hun prioriteiten

Omdat er juist bij deze pilot veel nieuwe technologische aspecten erbij komen, was de technische invulling van de software agent nog niet helemaal duidelijk. De gewenste functionaliteit had ik wel beschreven bij de systeemeisen. Echter, het was vooraf zeer moeilijk in kaart te brengen hoe de wat het uiteindelijke resultaat zou moeten worden. Daarom was ik voor deze pilot overgestapt van iteratiestrategie “incrementeel opleveren” naar een evolutionair iteratiestrategie. Als het nodig was, dan zou ik de definitiestudie voor deze pilot ook itereren. Een andere reden waarom ik voor deze pilot voor een evolutionaire aanpak heb gekozen, is omdat deze iteratiestrategie goed past bij het karakter van programmeren en valideren. Ik zal steeds stukje voor stukje moeten gaan programmeren en valideren, omdat het meeste programmeerwerk nog nieuw voor mij is. De software agent zal daarom ook op die manier stukje bij beetje evolueren naar zijn uiteindelijke vorm. Veel codes en tabellen zouden met het “trial&error” principe gevormd en gewijzigd moeten worden. Gezien de tijd die ik nog voor het ontwikkelen van de software agent over had, was deze keuze misschien niet de beste geweest. Echter, ik kon geen concessies doen betreffende de agent, omdat ik geen inschatting kon maken van hoe de agent er uiteindelijk uit zou gaan zien.

Met welke technologie ik de agent zou gaan ontwikkelen was nog een grote vraag. Bij het inventariseren van mogelijke protocollen en programmeertalen tijdens de definitiestudie vond ik termen als XML (Extended Markup Language), SOAP (Simple Object Access Protocol),

WDDX (Web Distributed Data Exchange) en screenscraping. Uit die inventarisatie kwam ik erachter dat WDDX al geïmplementeerd was in Coldfusion. Ik heb toen besloten om deze technologie te gaan gebruiken, omdat ik de agent in Coldfusion ging maken.

Iteratie 1

Na de tweede validatie bij het coderen in WDDX kwam ik erachter dat WDDX niet werkte voor de agent. Dit was omdat WDDX alleen kon werken bij sites waarbij de server WDDX ondersteunt. Aangezien de agent voornamelijk vacaturesites zou nagaan, was de kans niet groot dat de agent een vacature zou vinden, omdat de geen enkele vacaturesite op een server draait dat een WDDX ondersteuning heeft. Dit ben ik nagegaan door een lijst met vacaturesites op te stellen en in die lijst aan te geven of de site op een coldfusion server draait of niet. Dit heb ik kunnen achterhalen door na de extensie van de webpagina's te kijken. In figuur 44 ziet u een gedeelte van de inventarisatielijst.

Vacaturesite	WDDX ondersteuning
Vacaturebank.nl	nee
Jobnews.nl	Nee
Monsterboard.nl	Nee
Content.nl	Nee
Intermediar.nl	Nee
Vacatures-vacatures.nl	Nee
Jobtrack.nl	Nee
Werk.nl	Nee
Askjim.nl	Nee
Medischevacaturebank.nl	Nee
Denationalevacaturebank.nl	Nee
Vacatureplein.nl	Nee
Continu.nl	Nee
Werkplus.nl	Nee
Volkskrantbanen.volkskrant.nl	Nee
Stepstone.nl	Nee
arbeidsmarkt.jobbingmall.nl	Nee
allebouwwacatures.nl	Nee
Vacaturekrant.nl	Nee
jobzone.nl	Nee
avjob.nl	Nee
d-job.nl	Nee

Figuur 44 – Inventarisatie jobsites met een WDDX ondersteuning

Voor XML was eigenlijk dezelfde soort situatie als bij WDDX, omdat geen enkele site hun vacatures of cv's aanbod in XML formaat. De technologie SOAP was nog niet beschikbaar voor Coldfusion 5.0. Ik had toen besloten om de “screenscraping” techniek te gaan gebruiken. Deze techniek was net als WDDX ook beschikbaar in Coldfusion. Alleen was je met deze techniek niet afhankelijk van andere servers. Screenscraping kon bij iedere website gedaan worden. Dit komt omdat deze techniek de inhoud van het scherm schraapt zoals de term het zelf eigenlijk al aanduidt. Er zitten wel een aantal nadelen aan vast. In de eerste plaats je afhankelijk van de snelheid van de andere server. Indien die traag is, dan zal het “screenscraping” ook traag gaan. Indien de andere server uit staat, dan zal het “screenscraping” vastlopen.

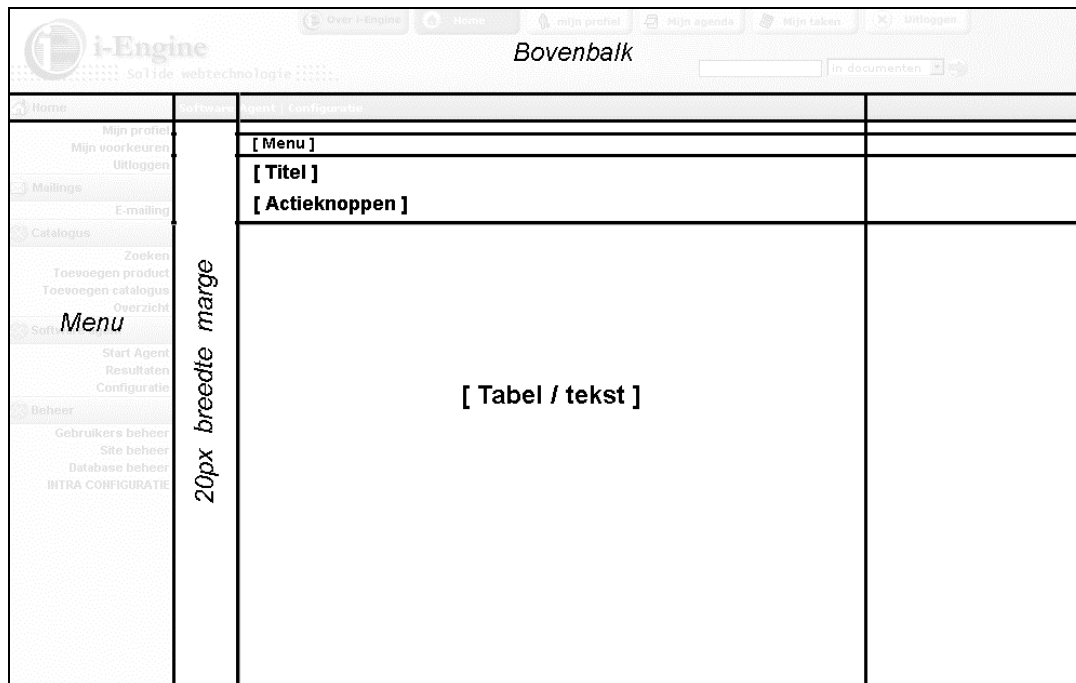
6.3.1.1 De configuratieschermen

In het pilotontwikkelpjan had ik ook de schermindelingen van de configuratieschermen van de software agent opgenomen. De bedoeling van deze configuratieschermen was om de software agent te configureren voor specifieke zoekgebieden. Voor meer informatie hierover verwijs ik u door naar paragraaf 6.3.1.2. Het intranet waarin de software agent uiteindelijk geïntegreerd zou worden, had al een huisstijl. Deze huisstijl was qua kleuren en lettertypen al helemaal vastgelegd in een zogenaamde “stylesheet”. Indien ik de schermen zou gaan ontwikkelen, hoefde ik in feite alleen de “stylesheet” in de pagina’s toe te voegen.

Ik hoefde me in feite alleen nog maar te richten op de indelingen van de schermen en de indelingen van de tabellen. Aangezien het intranet qua kleur en lettertype helemaal consistent was, wilde ik de vlakindelingen ook zoveel mogelijk laten overeen komen met de andere vlakindelingen van het intranet. In de figuren 45 en 46 ziet u twee schermindelingen. In het eerste figuur ziet u een schermindeling van een bestaande module in het intranet. In figuur 46 ziet u de schermindeling voor de te ontwikkelen configuratieschermen. Zoals u kunt zien verschilt de indeling nauwelijks. Het enige duidelijke verschil is de indeling van de tabellen. In de indeling van het configuratiescherm, wordt de breedte van de tabellen beperkt. Ik heb hiervoor gekozen omdat de data die wordt gepresenteerd in die tabellen geen lange regels nodig hebben. Indien ik de gewone breedte zou aanhouden, dan zou de tekst niet lekker lezen omdat je dan letterlijk van links naar rechts over het scherm moet kijken. Een ander punt dat opvalt, is dat bij het configuratiescherm een extra indeling is gemaakt. Deze indeling is gereserveerd voor actieknoppen: subbuttons behoren bij een bepaald vervolgscherm in het configuratiescherm.



Figuur 45 – Schermindeling een module van het huidige intranet



Figuur 46 – Schermindeling van de te ontwikkelen configuratieschermen

Om een totaalbeeld te krijgen van alle schermen heb ik een navigatieschema gemaakt. In dit navigatieschema heb ik alle schermen opgenomen en aangegeven hoe je kunt navigeren door het configuratiegedeelte.

6.3.1.2 Specificatie werking software agent

De globale werking van de software agent had ik al vastgelegd in een “activity diagram” in het systeemconcept. Zie paragraaf 5.6.1.2 voor meer informatie hierover. Dit was mijn eerste idee omtrent de interne werking van de agent. Kort samengevat kwam het idee hier op neer. De agent begint te zoeken vanaf een start url: een startdomein. Dit domein zou dan opgegeven moeten zijn in het configuratiescherm. Vervolgens gaat de agent de desbetreffende pagina compleet na op allerlei mogelijke URL's die zouden kunnen leiden tot een vacature. Indien hij denkt een geschikte URL gevonden te hebben, gaat hij die URL volgen en vervolgens controleren of de gevolgde URL een vacature is. Indien het een vacature is, probeert hij zoveel mogelijk informatie van deze pagina te verzamelen om dat vervolgens in de database op te slaan. Indien hij geen vacature vindt op die gevolgde URL, dan gaat hij weer verder zoeken naar andere URL's om vervolgens die eventueel weer te kunnen volgen. Dit proces blijft zich eendeloos herhalen totdat hij geen URL's meer heeft om te volgen.

6.3.2 Bouw

6.3.2.1 Pilotdeel “De tabellen”

De tabellen die nodig waren voor de software agent had ik tijdens de definitiestudie gemaakt en in het systeemconcept opgenomen. Echter, de kans was groot dat deze tabellen vaak gewijzigd moesten worden in verband met de manier van ontwikkeling van de software agent. Zoals ik al eerder heb besproken in paragraaf 6.3.1 zou het ontwikkelen voornamelijk een “trial&error” proces gaan worden. Desalniettemin kon het implementatiemodel dat was opgesteld tijdens de definitiestudie, gebruikt worden als startpunt. In figuur 47 ziet u het implementatiemodel van de tabel “sagagents”.

```
CREATE TABLE sagagents (  
    agent_ID INT(11) auto_increment NOT NULL,  
    name VARCHAR(50) NOT NULL,  
    maxtracinglinks INT(11) NOT NULL DEFAULT 100,  
    searcharea INT(11) NOT NULL DEFAULT 0,  
    matchpercentage DOUBLE NOT NULL DEFAULT 30,  
    PRIMARY KEY (Agent_ID),  
    KEY (searcharea)  
)
```

Figuur 47 – Het implementatiemodel van de tabel “sagagents”

In totaal had ik 6 tabellen opgesteld die ik ging nodig zou hebben voor zowel het opslaan van configuraties van verschillende agents en het opslaan van gevonden links en vacatures. Dit waren de volgende 6 tabellen:

- SAGagents
- SAGsearcharea
- SAGvacaturefound
- SAGfilterterm
- SAGsearchterms
- SAGlinksfound

Om de structuur van het intranet te handhaven waarin de software agent uiteindelijk geïmplementeerd zal gaan worden, heb ik voor de agent module de afkorting “SAG” gebruikt. In tabel “SAGagents” wordt informatie over verschillende agents opgeslagen. Over de agents wordt de naam, het maximaal te volgen links, het zoekgebied, het slagingspercentage en zijn status opgeslagen. Het maximaal te volgen links is er later bijgekomen. Eerder in dit hoofdstuk had ik aangegeven dat de agent continue zou gaan zoeken naar vacatures totdat hij geen links meer had om te onderzoeken. Echter, met het oog op de performance van de server wilde ik deze niet continue belasten aangezien het continue zoeken naar vacatures veel van de processor vergt. Daarom heb ik dit instelbaar gemaakt met de optie “maxtracinglinks”. Bij het onderzoeken van een pagina moet de agent ook weten wanneer hij een vacature mag beschouwen als een vacature volgens de normen van i-concept. Indien een vacature hoger uit komt dan het opgegeven slagingspercentage dan mag de gevonden vacature daadwerkelijk beschouwd worden als een vacature.

De normen worden opgeslagen in de tabel “SAGsearchterms”. Bij de tupel “term” wordt de gezochte term opgeslagen. Bij de tupel “factor” wordt aangegeven hoe belangrijk die term is. Ik heb de tupel “factor” erbij gedaan zodat het zoeken naar geschikte vacatures beter afgesteld kan worden.

Indien de agent een geschikte vacature heeft gevonden, is het de bedoeling dat de juiste data eruit wordt geëxtraheerd om dat vervolgens op te slaan in tabel “SAGvacaturefound”. Dat extraheren gebeurt aan de hand van termen ingevoerd in tabel “SAGsearchterm”. Indien de agent geen vacature op een pagina vindt, maar wel links dan worden die links opgeslagen in de tabel “SAGlinksfound”. Zodoende weet de agent waar hij is geweest en waar hij nog heen kan in zijn huidige zoekactie of in een toekomstige zoekactie. Wanneer de agent weer gaat starten dan kijkt hij eerst in de tabel “SAGlinksfound” over er nog onbezochte links over zijn. Als dat niet het geval is, dan neemt hij de starturl uit tabel “SAGsearcharea”.

Iteratie 1

Nadat ik een andere oplossing had gekozen voor het filteren van de tekst van een vacature, had ik de tabel “SAGfilterterm” niet meer nodig. Ik heb toen alles via script eruit kunnen filteren met behulp van mijn “relativeringstheorie”. Dit tabel heb ik toen verwijderd uit de database. Voor meer informatie over de theorie verwijs ik u door naar paragraaf 6.3.2.3.5.

6.3.2.2 Pilotdeel “De configuratieschermen”

Voor het maken van de configuratieschermen, heb ik gebruik gemaakt van Dreamweaver MX. Met dit programma in broncode modus kon ik snel de configuratieschermen ontwikkelen, omdat Dreamweaver een duidelijke kleurschema gebruikte voor de verschillende HTML tags. Daardoor was het programmeren voor mij heel erg overzichtelijk. I-Concept had voor het aanmaken van pagina’s bestemd voor het intranet een template geschreven. Zodoende bleef de codering overzichtelijk voor het geval iemand anders eraan moest werken. Figuur 48 geeft u een beeld van de template. In dit template ziet u dat ik mijn eigen geschreven code tussen de “BLOCKQUOTE” tags moest schrijven. Op die manier blijft de code gestructureerd en overzichtelijk. Verder ziet u dat ik ook de variabelen “navcategorie” en “pagtitel” moest invullen. Deze variabelen komen in het intranet terug in de vorm van twee koppen op de plaats van het kleine menu en de titel zoals in figuur 46.

```

<!-- ----->
<!--      Made by i-Concept InternetWorks www.i-concept.nl      -->
<!--      Auteur: (Vul hier je naam in)                          -->
<!-- ----->
<!--      Beschrijving: dit is een stramien voor alle functies    -->
<!-- ----->
<!--      Niet veranderen: OPEN-TAG (INGELOGDE USER CHECK)      -->
<cfif parameterexists(cookie.login)>
<!-- ----->
<!--      Stel ALTIJD het volgende in:                            -->
<!-- ----->
<!--      1. Memucategorie:                                       -->
<cfset navcategorie = "Vul hier een naam in!">
<!-- ----->
<!--      2. Titel                                                -->
<cfset pagtitel = "Vul hier een naam in!">
<!-- ----->
<!--      Niet veranderen: deze include verwijst naar klant afhankelijke variabelen -->
<!--      Kijk wel even voor de zekerheid of het pad klopt!      -->
<cfinclude template="../../config/variabelen.cfm">
<!-- ----->
<!--      Niet veranderen: deze include is voor het design (header) -->
<!--      Kijk wel even voor de zekerheid of het pad klopt!      -->
<cfinclude template="../../style/layout/header-default.cfm">
<!-- ----->
<!--      .....START coldfusioncode.....                        -->
<blockquote>

JOUW CODE HIER

</blockquote>
<!--      .....STOP coldfusioncode.....                        -->
<!-- ----->
<!--      Niet veranderen: deze include is voor het design (footer) -->
<!--      Kijk wel even voor de zekerheid of het pad klopt!      -->
<cfinclude template="../../style/layout/footer.cfm">
<!-- ----->
<!--      Niet veranderen: SLUIT-TAG (INGELOGDE USER CHECK)      -->
<cfelse>
    <cfinclude template="../../config/notloggedin.cfm">
</cfif>
<!-- ----->

```

Figuur 48 – De intranet template

Om de opmaak zo consistent mogelijk te houden, had ik een “include” bestand gemaakt met daarin het menu. Dit bestand heb ik vervolgens bij alle schermen van het configuratiegedeelte toegevoegd met de tag: “< CFINCLUDE >”. Zodoende blijft het menu altijd op dezelfde plaats gepositioneerd. Het kan dan niet per ongeluk verschoven worden.

In het hoofdmenu van het configuratiegedeelte zijn er in totaal drie knoppen aanwezig: “agent configuration”, “search words” en “search areas”. Bij alle drie de gebieden heb ik voor het invoerscherm dezelfde structuur genomen qua indeling maar ook qua codering. Voor de indeling bij alle invoerschermen heb ik bovenin in het scherm steeds het hoofdmenu van het configuratiegedeelte terug laten komen. Zodoende kan de gebruiker sneller en gemakkelijker navigeren door het configuratiegedeelte. In figuur 49 ziet u hoe ik het hoofdmenu heb terug laten komen in een invoerscherm.

Software Agent | Configuratie

[agent configuration](#) [search words](#) [search areas](#)

Agent Configuratie

Naam:

links volgen: Hoe hoger het getal, hoe zwaarder de applicatie moet gaan draaien

Zoekgebied:

Matchpercentage: Minimale percentage waaraan een vacature moet voldoen.

Actief: ☒

Figuur 49 – De schermindeling van het invoerscherm voor de tabel “sagents”

Onder het menu heb ik steeds de titel van de desbetreffende pagina laten terugkomen. Vervolgens heb ik daaronder de invoervelden geplaatst. Helemaal onderin heb ik de actieknoppen geplaatst. Bij alle drie de invoerschermen zijn dat er steeds drie te weten: “annuleer”, “verwijder” en “wijzig”. Door de invoerscherm steeds op deze manier op te bouwen, vermijd ik onduidelijkheden in de GUI en daarmee ook eventuele onzekerheden. De gebruiker raakt door deze consistentie sneller vertrouwd met de configuratieschermen.

Wat betreft de codering heb ik steeds een bepaalde structuur gehanteerd waarbij ik het mogelijk heb gemaakt om zo min mogelijk aantal pagina's te gebruiken. Op die manier heb ik het overzicht behouden. Zo heb ik voor het wijzig scherm steeds de volgende structuur gebruikt. In de eerste paar regels wordt gekeken of er een ID aan de pagina wordt doorgestuurd. Dit kan een waarde zijn dat verkregen wordt van zichzelf doormiddel van een formulier. Maar het kan ook zo zijn dat de waarde verkregen wordt via een andere pagina. Vervolgens wordt er gekeken of er een paramater “actie” ook is meegestuurd. Dit kan ook zowel vanaf de eigen pagina als vanaf een andere pagina. Vervolgens wordt er gecontroleerd wat die paramater voor waarde heeft. De paramater kan in totaal vier verschillende waardes aannemen: edit, delete en display. Aan de hand van één van die waardes worden er de bijbehorende acties en/of SQL queries uitgevoerd en keert het programma weer terug in het hoofdmenu. Zoals al eerder heb gezegd heb ik op deze manier het aantal pagina's weten te reduceren, omdat ik in één zo'n invoerscherm meerdere functies tegelijk laat zien. Ik heb dit gedaan met de bedoeling het gebruiksgemak van het programma te bevorderen, omdat de gebruiker nu alle functies bij de hand heeft. Het resultaat was dat ik per onderdeel slechts drie pagina's hoefde op te leveren en in totaal tien voor het gehele configuratiegedeelte 10. Indien ik dit niet op die manier had gecodeerd, moest ik voor het gehele configuratiegedeelte 16 pagina's coderen. Daarnaast probeerde ik door juist compacter te coderen, tijd terug te winnen die ik had verloren tijdens het itereren van het pilotdeel “Het overzetten van data”. Daarbij heb ik geen concessies gedaan, omdat ik nog wel alle functionaliteit erin geprogrammeerd had.

Naast het configuratiegedeelte had ik ook nog een resultatenoverzicht gemaakt. Dit onderdeel heeft met doorvoering van de consistentie precies de zelfde schermindeling en compacte codering als het configuratiegedeelte. Dit resultatenoverzicht had als functie de gevonden vacatures te laten zien, bij te werken, te verwijderen en eventueel te publiceren op de eigen website arbeidsmarkt.com

6.3.2.3 Pilotdeel “De software agent”

6.3.2.3.1 Het zoeken naar URL's

In het pilotontwikkelplan had ik besloten om de techniek WDDX (Web Distributed Data Exchange) te gaan gebruiken voor de ontwikkeling van de agent. In de eerste plaats heb ik hiervoor naar een tutorial gezocht waarmee ik mezelf de taal kan aanleren. Hiervoor heb ik via google.nl gezocht met de zoektermen “tutorial wddx coldfusion”. De volgende website leek mij geschikte tutorial te bevatten: “Webmonkey”⁸. Op deze site vond ik een stukje script met betrekking tot WDDX.. Figuur 50 geeft u een beeld van WDDX scripting.

```
<CFHTTP METHOD="GET" URL="http://www.forta.com/cf/tips/syndicate.cfm"
TIMEOUT="60">

<!--- Extract data --->
<CFWDDX ACTION="WDDX2CFML" INPUT="#CFHTTP.FileContent#"
OUTPUT="DailyTip">

<!--- Display output --->
<CFOUTPUT>
<TABLE WIDTH="300">
  <TR><TH><A HREF="#DailyTip.URL#">#DailyTip.title#</A></TH></TR>
  <TR><TD
ALIGN="center"><B>#HTMLEditFormat(DailyTip.header)#</B></TD></TR>
  <TR><TD><FONT SIZE="-1">#HTMLEditFormat(DailyTip.body)#</FONT></TD></TR>
</TABLE>
</CFOUTPUT>
```

Figuur 50 – WDDX scripting

Ik had toen dit script ingevoerd in een testpagina om het vervolgens snel te valideren met de “White-box” testmethode. Ik kreeg zonder problemen direct data op mijn scherm te zien. Daarna veranderde ik de URL in de “<CFHTTP>” tag in een andere site. Ook dit ging ik weer testen. Echter, nu verscheen er helemaal geen data op het scherm. Ik heb toen de gevonden tutorial nog een keer geraadpleegd. Op de laatste pagina van die tutorial las ik dat WDDX alleen werkt indien de server WDDX packets verstuurd. Ik had toen besloten om deze techniek niet te gaan gebruiken, omdat het erg afhankelijk was van de serverkant. Ik had toen besloten om het pilotontwikkelplan snel te itereren.

Als resultaat van die eerste iteratie van het pilotontwikkelplan, bleek dat de techniek “screenscraping” de enige techniek is het minst afhankelijk is van andere servers. Ik had toen besloten om de techniek “screenscraping” te gaan gebruiken. Met “screenscraping” haal je letterlijk de gehele inhoud van het scherm en dus ook de broncode. Dit was voor mij erg interessant omdat ik bij het extraheren van een vacature uit een pagina ook moest kijken naar de broncode. Ik had toen via google.nl gezocht op met de zoekterm “screenscraping coldfusion”. Het opgeleverde zoekresultaat leverde niks bruikbaar voor mij op. Ik had toen de zoekterm gewijzigd in “screenscraping coldfusion”. Ik kreeg toen veel meer resultaten. De website “CFHUB”⁹ leek mij een goede site met veel bruikbare informatie over “screenscraping” met Coldfusion. Op deze pagina vond ik informatie over de Colfusion tag “<CFHTTP>” waarmee je de complete inhoud van een andere pagina kon kopiëren. Bij het

⁸ http://hotwired.lycos.com/webmonkey/99/15/index3a_page5.html?tw=authoring

⁹ <http://cfhub.com/advanced/cfhttp/copy.cfm>

kopieren had ik twee keuzes. De eerste keuze was om de inhoud op te slaan in een bestand. De tweede keuze was om de inhoud tijdelijk te bewaren in een variabele. Juist de laatste keuze sprak mij erg aan, omdat ik er dan bewerkingen mee kon doen en binnen de waarde van die variabele kon zoeken naar URL's. Nadat het me was gelukt om de inhoud van een willekeurig webpagina te kopiëren, heb ik dit nog eens getest met een webpagina, "http://www.stepstone.nl/home_fs.cfm?lang=NL", van de website stepstone.nl. Ook hier bleek de tag "<CFHTTP>" de inhoud van de pagina te kopiëren.

6.3.2.3.2 Het filteren en sorteren van links

Ik ben toen vervolgens een script gaan schrijven waarmee je alle links uit een pagina kan extraheren. Het vinden van links is in de beginfase voor de agent van groot belang, omdat die anders geen aansluitpunten heeft om eventueel verder te zoeken naar vacatures. De URL's heb ik kunnen vinden met behulp van "Regular Expressions". Deze regular expressions, of kortweg regexps, kunnen tekst na gaan op bepaalde patronen. Zo kan je bijvoorbeeld gaan zoeken naar een woord dat minstens vier keer de letter K bevat. Het zoeken naar bepaalde patronen lijkt wel op "Fuzzy Logic", wat eigenlijk wel zo gepast is bij een agent dat iets moet vinden uit vreemde bronnen. Om alle URL's in een pagina te vinden heb ik de volgende regexp geschreven:

```
<a [^>]+href=( [^"] *) "[^>]*(<[*[^>]]*>*)(<\/+[Aa][^>]*>)
```

Hiermee geef ik aan dat ik een tekencombinatie wil vinden dat begint met "<a" wat de HTML tag is voor het opgeven van een link. Vervolgens mag die tag niet gelijk sluiten maar moet er nog een verwijzing komen met "href=". Als eindbaken zocht ik in die tekencombinatie de sluiting van de HTML tag: "". Door deze regexp in combinatie met de Coldfusion functie "REFindNoCase" uit te voeren, vond ik steeds een de startpositie van een link. Dit zoekproces had ik in een "WHILE"-loop geplaatst en dat herhaaldelijk laten uitvoeren totdat er geen URL's meer aanwezig waren. Figuur 51 geeft u inzicht in het script die ik heb geschreven voor het eruit filteren van links.

```
while(true)
{
    foundMatches = REFindNoCase('<a[>]+href=(["]*)"[^>]*(<[*[^>]]*>*)(<\/+[Aa][^>]*>)', searchResult,
    searchStartPosition, true);

    if(foundMatches.pos[1] EQ 0)
        break;
    else
    {
        foundString = Mid(searchResult, foundMatches.pos[1], foundMatches.len[1]);
        foundString = ReplaceNoCase(foundString, "_self", "_blank");
        foundString = ReplaceNoCase(foundString, ":80", "");
        loopCount = loopCount + 1;
        QueryAddRow(linksQuery);
        QuerySetCell(linksQuery, "links", foundString, loopCount);
        searchStartPosition = foundMatches.pos[1] + 1;
    }
}
```

Figuur 51 – Script voor het eruit filteren van URL's

Ik heb dit stukje script vervolgens getest. Daarbij heb ik de gevonden waarden afgedrukt op het scherm, zodat ik precies kon zien of de geëxtraheerde tekst ook daadwerkelijk een link was. Als resultaat kreeg ik alleen maar links op het scherm te zien. De regular expressions bleken een zeer krachtig hulpmiddel te zijn bij het zoeken naar bepaalde patronen. Ik heb daarom besloten om, waar het mogelijk is, zoveel mogelijk regular expressions te gebruiken. Op die manier ben ik minder invalide voor slecht gecodeerde sites waarbij het een onbegonnen taak is om tekst eruit te filteren met de gewone “string” functies van Coldfusion.

Nadat ik de links eruit had gefilterd, was het zaak om de gevonden links te sorteren op mogelijke vacatures en niet mogelijke vacatures. Voor het uitvoeren van deze taak had ik het volgende geredeneerd. Alle URL's die een vraagteken bevatten sturen vaak informatie door naar een volgende pagina. Deze informatie zou eventueel ook informatie kunnen bevatten over een vacature. Bijvoorbeeld het ID nummer van een vacature. Vraagtekens kunnen het vaakst voorkomen bij extensies als PHP, ASP, CFM, JSP, HTML, XHTML en FPL. Dit zijn namelijk de zogenaamde programmeertalen voor het web en bij programmeertalen is het vaak nodig om informatie door te sturen naar een andere pagina. Kortom, links die een webpagina bevatten met een van de zonet opgenoemde extensies en een vraagteken, kunnen mogelijke vacatures zijn. Alle links die dat niet hebben, kunnen worden niet beschouwd worden als mogelijke vacatures. Dit heb ik met de combinatie van de functie “REFindNoCase” en een regexp gecontroleerd en getest. Evenals bij de vorige “white-box” test heb ik ook nu weer de gevonden waarden op het scherm afgedrukt zodat ik kon controleren of het goed ging. Daarnaast heb ik ook het totale aantal links afgedrukt op het scherm, zodat ik kon controleren of het script alle links had gecontroleerd.

6.3.2.3.2 Bepalen welke link het meeste voorkomt in een pagina

De volgende stap die ik wilde gaan doen was het bepalen welke pagina het meeste voorkomt in een pagina. Dit wilde ik gaan doen op de gefilterde selectie van pagina's die mogelijk een vacature kunnen zijn. Op die manier kan ik met een beetje meer zekerheid zeggen dat een van die pagina's een vacature kan zijn. Het uitzoeken welke pagina het meeste voorkomt heb ik gedaan door de mogelijke vacatures stuk voor stuk na te lopen. Dit heb ik gedaan met een “FOR”-loop. In die loop heb ik bijgehouden hoe vaak een vacature voorkwam in. Daarbij heb ik de URL variabelen, de waarden achter de vraagteken, verwijderd om ervoor te zorgen dat de agent altijd alleen de pagina's zelf met elkaar vergelijkt. De webpagina's “default.asp” en “default.asp?id=334” zijn wel dezelfde pagina's alleen de URL is anders. Dus alles wat na de extensie van een webpagina komt moet verwijderd worden om ervoor te zorgen dat de agent geen appels met peren vergelijkt. Dit heb ik kunnen verwezenlijken in met regexps. Echter niet in 1 keer. De eerste keer gebruikte ik de volgende regexp:

```
+[^?]+[A-Za-z0-9_-]+[ ]+
```

Deze regexp vond helemaal niets. Ik heb toen met de “trial&error” techniek het plus-teken veranderd in een ster. Een plus-teken houdt in dat een karakter 1 of meer kan mag voorkomen.

Een ster houdt in dat een karakter 0 of meer keer mag voorkomen. Echter, dit werkte ook niet. Ik heb toen de volgende regexp gebruikt:

```
href=("[^"]")+["\?"]+
```

Hierbij ging ik ervan uit dat de lettercombinatie “href” altijd bestaat in een link. Zodoende had ik in ieder geval een houvast. Vervolgens heb ik deze regexp gecombineerd met een paar standaard Coldfusion functies als “MID” en “RIGHT”. Met “MID” zoek je ergens in het midden van een string en wijk je uit naar de rechterkant. Met “RIGHT” zoek vanaf de rechterkant naar links. Met behulp van deze twee functies heb ik de URL zonder parameters eruit kunnen filteren. Dit heb ik getest door de links naar mogelijke vacatures op het scherm af te drukken met daarnaast de gefilterde webpagina. Naast alle links kreeg ik rechts de losse webpagina’s uit die links. Nu kon de agent de webpagina’s goed vergelijken en bij elkaar optellen.

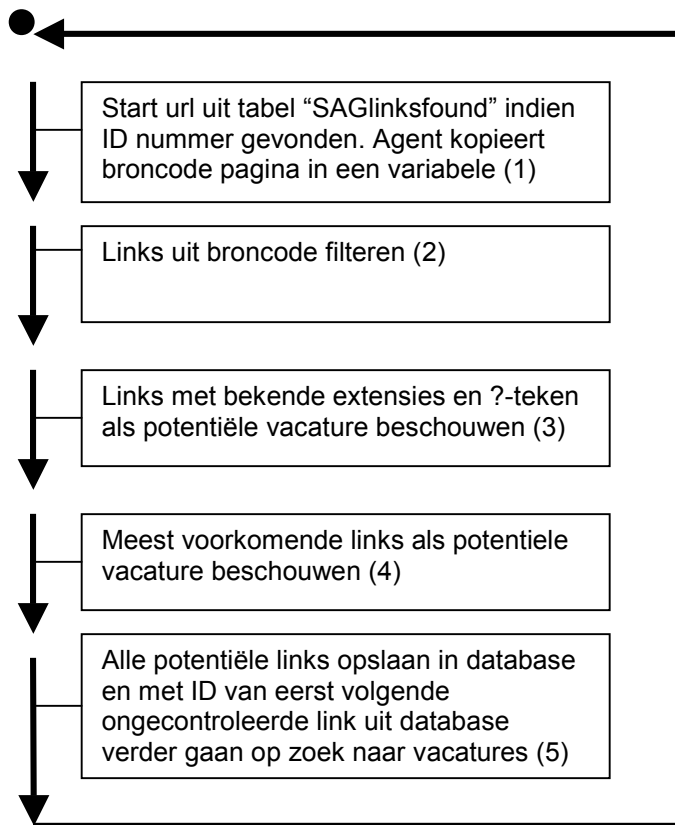
De pagina die het meest voorkomt bij de telling, is hoogstwaarschijnlijk een detailpagina. Met andere woorden, de kans dat die pagina een vacature is, is groter geworden. Dit heb ik uitgedrukt in procenten, omdat er bij de volgende stap word gekeken naar het aantal procent dat een pagina voorkomt. Ik heb dit in procenten gedaan, omdat je dan de verhouding weet. Stel je hebt 100 links op een pagina en 20 links op een andere pagina. Dan kan je niet zeggen dat een pagina minstens 30 keer moet voorkomen. Dan zou er van de tweede pagina geen enkele link een mogelijke vacature meer kunnen zijn. Indien je het in procenten meet, dan meet je in verhoudingen. En dan kan ik voor alle pagina’s een minimale waarde opstellen ongeacht het aantal. Voor de agent heb ik een minimale waarde van 35% opgesteld. Indien het aantal procent dat een link voorkomt in een pagina lager is dan 35%, dan wordt deze link niet beschouwd als een detailpagina. Ik ben aan deze grens gekomen door een aantal keer een kleine “white-box” testjes te doen om te kijken of er links niet werden doorgelaten terwijl ze wel leiden naar een vacaturepagina. Echter hun procentuele deel op die pagina was lager dan de grens. Zodoende heb ik steeds de grens bijgesteld totdat de hiervoor genoemde situatie niet meer voorkwam.

6.3.2.3.3 Het volgen van een link

Nu ik de meest potentiële links uit een pagina heb kunnen filteren, was het zaak om dat die links te volgen om te kijken of die link een vacaturepagina opleverde. Om die link te volgen heb ik met behulp van javascript een “x” aantal nieuwe windows geopend met dezelfde pagina van de agent waarin de meegestuurde link werd gevolgd. Het getal “x” staat voor het aantal potentiële links. In het geval van 5 potentiële links zou de agent 5 nieuwe windows openen met een replica van zichzelf alleen met een andere link, namelijk de link die vanuit het eerste scherm is meegestuurd. Vanuit hier begint het proces eigenlijk op zoals. De agent gaat de pagina weer af op links. Vervolgens filtert, sorteert en telt hij ze waarna hij weer een “x” aantal nieuwe windows opent met een nieuwe link. Het oude scherm wordt overigens automatisch gesloten, wanneer het nieuwe scherm is geladen. Dit heb ik getest met de “white-box” testmethode. Nu bleek dat bij een klein aantal te volgen links de server het nog aan kon. Met klein bedoel ik 5 tot 10 keer. Echter, de agent deed al een aardig beroep op de processor. Het verbruik van de CPU schommelde tussen de 40% en 80%. Indien ik het aantal links volgen zou verhogen, is de kans groter dat 1 pagina meer nieuwe windows

tegelijkertijd opent. Deze windows zouden op hun beurt ook weer veel windows kunnen openen. Kortom, het aantal windows stijgt exponentieel. Aangezien er al een aardig beroep werd gedaan op de processor is een exponentiële stijging in het verbruik van de processor een slechte zaak. Ik heb hiervoor dus een andere oplossing moeten vinden.

Ik heb toen besloten om niet meer gebruik te maken van het javascript waarmee de agent steeds nieuwe windows opende. In plaats daarvan heb ik besloten dat de agent de link naar zichzelf stuurt. Met andere woorden, hij herlaadt zichzelf met die nieuwe link. Aangezien de agent nu slechts 1 link per keer kan behandelen, was ik genooddaakt om de gevonden links op te slaan in de database. De gevonden links heb ik toen opgeslagen in de tabel "SAGlinksfound". In deze tabel kreeg elk gevonden unieke link een uniek ID nummer. Daarnaast werd er per link bijgehouden of deze al was gecontroleerd of niet. Door na de tellingen van links gelijk de meest potentiële links op te slaan in de database, kon ik daarna weer met een query uit diezelfde tabel de eerst volgende link selecteren die nog niet gecontroleerd was. Van die link gebruikte ik het ID nummer om aan de agent duidelijk te maken welke link hij moest volgen. Vervolgens herlaadt de agent zich met een meegepaste parameter: het uniek ID nummer van een link. Het begin van het script heb ik moeten aanpassen, omdat ik nu in het begin moest controleren of er een nummer was meegegeven of niet. Indien er een ID nummer is gevonden, dan wordt de link behorende bij dit nummer uit de database gehaald. De pagina behorende bij deze link wordt geopend en het hele proces van filter, sorteren en tellen begint weer opnieuw. In figuur 52 ziet u een schematische weergave van het nieuw geschreven script.



Figuur 52

Schematische weergave v/d werkzaamheden v/d agent

6.3.2.3.4 Controleren op vacature

Ik was nu zover dat ik de robot eindeloos links kon laten volgen. Nu was het zaak om de de pagina's te controleren of ze een vacature zijn of niet. Dit ga ik doen met behulp van de tabel "SAGsearchterms". Zoals ik al eerder had beschreven wordt hier de agent eigenlijk geconfigureerd. In dit tabel worden de termen met bijbehorende factor ingevoerd die i-Concept belangrijk achten voor een goede vacatureopmaak en omschrijving. Bij de bepaling of een webpagina wel of geen vacature is, laat ik de agent rekenen. In de eerste plaats moet hij een aantal termen vinden in de gekopieerde webpagina. Stel dat de er in de tabel "SAGsearchterms" slechts vier termen stonden: vacature met een factor 8, bedrijfsomschrijving met een factor 6, CV met een factor 8 en vacatureomschrijving met een factor 4. Nu heeft de agent een webpagina waarop hij alleen de termen cv en vacatureomschrijving heeft gevonden. Dan zou het slagingspercentage voor die vacature conform de normen gesteld in de tabel het volgende zijn: $(8+4 / 8+4+6+8) * 100\% \approx 46\%$. Stel dat er bij de agent een minimaal slagingspercentage van 35% is ingesteld, dan zou deze webpagina door de agent beschouwd worden als een vacature. In figuur 53 ziet u hoe ik dit proces heb gecodeerd.

```
count = 0;
while (count lt getTerms.RecordCount)
{
    count = count + 1;
    ratingtotal = ratingtotal + val(getTerms.factor[count]);
    if (REFindNoCase(getTerms.term[count],searchResult, 1) neq 0)
    {
        totalscore = totalscore + getTerms.factor[count];
    }
}
matchscore = (totalscore / ratingtotal) * 100;
```

Figuur 53 – Script voor het uitrekenen van het slagingspercentage van een webpagina

Dit script heb ik getest met de "white-box" testmethode. Ik heb hierbij alle waarden op het scherm afgedrukt, zodat ik het zelf nog kon narekenen. Figuur 54 geeft u een resultaat die ik had verkregen tijdens het "white-box" testen.

```
gevonden termen:
- 10 ----- wegingsfactor: functie
- 10 ----- wegingsfactor: vacature
- 8 ----- wegingsfactor: branche
- 8 ----- wegingsfactor: cv
- 8 ----- wegingsfactor: eisen
- 8 ----- wegingsfactor: job
- 8 ----- wegingsfactor: omschrijving
- 3 ----- wegingsfactor: ervaring
- 3 ----- wegingsfactor: functie-eisen
- 2 ----- wegingsfactor: opleiding

totaal mogelijke score: 212
totaal behaalde score: 68
gescoorde matchingspercentage: 32.0754716981%
```

Figuur 54 – Een resultaat tijdens het "white-box" testen

6.3.2.3.5 Vacaturegegevens extraheren

Nu de de webpagina was bepaald als een vacature, kon ik beginnen met het schrijven van scripts waarmee ik de benodigde vacature gegevens kon extraheren. Maar voordat ik de vacaturetekst kon extraheren, wilde ik eerst zoveel mogelijk onnodige HTML tags en andere vormen van tags verwijderen. Het verwijderen van alle onnodige HTML tags heb ik verwezenlijkt met de volgende regexps in:

1. `<img[^>]+src=("[^"]*"')[">"]*>`
2. `[#chr(13)##chr(10)#]`
3. `(<td></td>)`
4. `(<tr></tr>)`
5. `<map[^>]*?>.*?</map>`
6. `<head[^>]*?>.*?</head>`

Door deze regexps in combinatie met de functie `REReplaceNoCase` uit te voeren heb ik alle HTML tags die aan één van de hier zes genoemde regexp voldoet kunnen vervangen met niks. Ik zal nu even in het kort uitleggen wat die zes regexps doen. De eerste regexp selecteert alle plaatjes uit de tekst. De tweede regexp selecteert alle enters en nieuwe regels. De derde regexp selecteert alle lege tabelkolommen. De vierde selecteert alle lege tabelrijen. De vijfde regexp selecteert alles wat tussen de tags “<map>” en “</map>” staat. De laatste regexp selecteert alles wat tussen de tags “<head>” en “</head>” staat. Door dit allemaal te verwijderen heb ik al een veel “schonere” broncode en tevens duurt het extraheren van data minder lang, omdat de volgende regexps niet meer door in onnodig code hoeft te zoeken.

Nu kwam ik voor een grote vraag te staan. Hoe weet ik nou wat vacaturetekst is en wat niet? Mijn eerste theorie ging als volgt. Ik ging ervan uit dat een webpagina was opgesteld volgens een bepaalde opmaakstructuur. Aangezien een vacature meestal een aparte webpagina is, zullen er in de webpagina's niet gelinked worden naar stylesheets. Hooguit zal de stylesheet zelf in de webpagina voorkomen. Echter, voor de opmaak wordt er dus gebruik gemaakt van verschillende HTML tags voor dikke letters, schuine letters, paragrafen, kopteksten etc. Met de standaard HTML hiërarchie voor de opmaak, heb ik getracht te achterhalen wat nou een vacaturetitel is en wat tekst. Zo zal een “<H1>” tag hoogstwaarschijnlijk gebruikt worden voor een kop en “<p>” voor een paragraaf.

Op die gedachte in mijn achterhoofd ben ik gaan programmeren. In eerste instantie heb ik de agent laten zoeken naar “<H1>” of andere header tags. Voor de situatie dat de agent een “<H1>” zou vinden, had ik een aantal “rules” opgesteld. Indien er maar 1 “<H1>” tag aanwezig is in de gekopieerde pagina, dan zal dit hoogstwaarschijnlijk de titel zijn van de vacature. Indien er meerder header tags in de pagina voorkwamen, dan moest de agent gaan uitzoeken welke type header tags er werd gebruikt. De grootste header werd door de agent beschouwd als de titel. De kleinere header werd beschouwd als een subtitel. Indien er slechts twee headers waren en ze waren allebei van dezelfde grootte dan werd er gekeken naar bijzonderheden zoals toegevoegde classes en het expliciet aangeven van kleuren. De header die meer toegevoegde dingen bevatte was werd door de agent uiteindelijk als de titel van de vacature beschouwd. Op deze manier werd er ook vergelijkingen gedaan tussen “<H1>” en “<p>”, tussen “<H1>” en “

” en tussen “<p>” en “

”. Verder had ik

de agent zo geprogrammeerd dat hij indien hij iets had gevonden dat gelijk op het scherm liet zien, zodat ik precies kon zien wat hij aan het doen was. Naast de controle op de tags werd er ook nog een controle gedaan met behulp van de tabel “SAGfindterm” waarin een aantal termen stonden die werden gebruikt voor het nagaan van de tekst tussen de tags. Op die manier wilde ik kunnen achterhalen of dat stukje tekst hoort bij een bepaalde titel zoals die in de tabel “SAGfindterm” is aangegeven.

Ik heb dit toen meerdere malen getest met de “white-box” testmethode. De testsite was “http://www.stepstone.nl”. In veel gevallen kon hij onderscheid maken tussen wat een titel was en wat tekst. Het gebruik van “SAGfindterm” leverde telkens helemaal geen resultaat op. Echter, op een gegeven moment ging stepstone.nl hun site vernieuwen en kon de agent niks meer beginnen op hun site, omdat hun server voor een tijdje offline stond. Ik heb dus een andere vacaturesite moeten aannemen als testsite. Echter, bij veel sites kon de agent geen links vinden. Die hadden de makers van die sites verborgen. In de adresbalk zag je ook allen maar de originele url en geen webpagina’s erachter. Ik ben toen eerst gaan zoeken naar een vacaturesite die wel hun links lieten zien en bovendien een redelijk normale opmaak hebben voor hun vacatures. Door het systematisch afgaan van de inventarisatielijst met verschillende vacaturesites die ik tijdens de 1^e iteratie van het pilotontwerp van deze pilot had opgesteld, kwam ik uiteindelijk bij de site vacaturekrant.nl terecht. Ik heb toen op deze site een “whitebox” test gedaan. Als resultaat kreeg ik eindelijk wel weer koppen en tekst. Echter, de site had niet de standaard hiërarchie aangenomen voor de opmaak. Dit hield in dat sommige titels als tekst werd beschouwd en sommige stukken tekst als titel.

Gezien het tijdverlies die ik nu weer heb opgelopen en de achterstand die ik al had, heb ik besloten om deze theorie te laten vallen en een andere theorie te bedenken. Gezien het laatst behaalde resultaat, kon ik nu niet meer zeker zijn dat de standaard hiërarchie ook daadwerkelijk op die manier gebruikt werd door iedereen. Ik heb toen een andere theorie bedacht. In plaats van naar de hiërarchie te kijken, wilde ik de agent nu laten kijken tussen de onderlinge verhoudingen bij de gebruikte tags. Zodoende zou de agent de doelen van die gebruikte kunnen relativeren. Zoals voorheen wordt de broncode van de gekopieerde pagina eerst zoveel mogelijk opgeruimd van onnodige tags. Vervolgens heb ik de agent alle resterende tags laten onderzoeken op het type tag, of het een titel of tekst is en de tekst tussen tags. De laatste twee controles kunnen alleen uitgevoerd worden indien het type tag is toegestaan. De tags die zijn toegestaan zijn: font, b, h1 t/m h5, br, li, lu, en p. Alle andere tags worden alsnog verwijderd uit de broncode. Alle toegestane tags worden opgeslagen in een array. Ook wordt opgeslagen of het een titel of tekst is en de tekst zelf wordt ook opgeslagen. Indien de toegestane tag de allereerste tag is, dan gaat de agent automatisch ervan uit dat dit een titel is. Indien blijkt dat de volgende toegestane tag van hetzelfde type tag is, bijvoorbeeld een “<p>” tag, dan blijft de eerste tag een kop en wordt de tweede tag beschouwd als tekst. Indien de tweede tag een “<h1>” tag zou zijn en de eerste tag is een “<p>” tag, dan zou het raar zijn dat “<p>” als header wordt beschouwd en “<h1>” als tekst. De agent verandert de eerste tag en markeert het als tekst. De tweede tag wordt door de agent gemarkeerd als header. Figuur 55 geeft u een beeld van de “relativeringregels” die ik heb aangemaakt om de tags te kunnen relativeren met elkaar.

```

if ((TagTypeSN[checktagIndex - 1] eq "font") AND
(TagTypeSN[checktagIndex] eq "font"))
    {TagHeaderText[checkTagIndex] = "h";}
else if ((TagTypeSN[checktagIndex - 1] eq "font") AND
(TagTypeSN[checktagIndex] eq "b"))
    {TagHeaderText[checkTagIndex] = "t";}
else if ((TagTypeSN[checktagIndex - 1] eq "font") AND
(TagTypeSN[checktagIndex] eq "h1"))
    {TagHeaderText[checkTagIndex] = "h";
    TagTypeSN[checktagIndex - 1] = "t";}
else if ((TagTypeSN[checktagIndex - 1] eq "font") AND
(TagTypeSN[checktagIndex] eq "br"))
    {TagHeaderText[checkTagIndex] = "t";}
else if ((TagTypeSN[checktagIndex - 1] eq "font") AND
(TagTypeSN[checktagIndex] eq "li"))
    {TagHeaderText[checkTagIndex] = "t";}

```

Figuur 55 – “Relativeringsregels” voor het onderscheiden van titel en tekst

Bij het opstellen van deze relativeringsregels heb ik steeds de “white-box” testmethode toegepast om te kijken of de codering de juiste resultaat geeft. De waardes die de agent heeft gevonden heb ik op het scherm weergegeven, zodat ik precies kan zien wat er allemaal gebeurt. Een aantal keren heb ik de regels moeten bijstellen. De eerste keer was dat twee “
” tags achter elkaar de eerste “
” tag een kop werd. Maar dat hoefde niet altijd zo te zijn. Ik heb voor deze situatie de agent nog een derde tag erbij laten halen. De tag voor de eerste tag. Zodoende kan de agent eruit opmaken of de middelste tag nu echt een kop is of niet. De tweede keer merkte ik een fout op bij de twee “” tags. Ook voor deze situatie heb ik de agent een derde tag erbij laten halen om te bepalen of de middelste tag nu daadwerkelijk een titel is. Toen ik het weer ging testen, kreeg ik geen foute indelingen meer. De agent had alle gevonden vacature tekst verdeeld in kop en tekst. Ik heb toen geprogrammeerd dat de agent de titels en de teksten bij elkaar doet en er weer een vacature van maakt, alleen dit keer wordt de vacature opgeslagen in de database.

Voor wat betreft de benodigde data over een vacature dat geëxtraheerd moest worden en in een apart veld moest worden opgeslagen, heb ik regexps gebruikt. Dit was nodig voor de velden “title”, “bedrijf”, “contactpersoon”, “email”, “telefoon” en “fax”. Dit waren respectievelijk de volgende 6 regexps:

1. `(( (titel:) | (<h1[^>]*>[a-zA-Z0-9][^>]*>) | (<b [class]*[^>]*>[a-zA-Z0-9][^>]*>))`
2. `(( ([n|b][\.]?v[\.]?) | ([GmbH | [V[\.]?O[\.]?F[\.]?`
3. `(( [Mm]evrouw | [Mm]evr[\.]? | [Mm]w[\.]? | [Mm]r[\.]? | ([Mm]eneer) | [Dd]hr[\.]? | ([Dd]e)?[ ]? [Hh]eer) ([ [A-Z][a-z]+)? (( [A-Z][a-z]+) + | ([ ]?[A-Z][\.]?)+ [ ]? ) (de(n|r)? [ ] | van[ ](de)?(n|r)? [ ] | v[/]?d)?([A-Z][a-z\-\.]?)+`
4. `(( ([[:alnum:]] [-a-zA-Z0-9_%\.\-]*)? [[:alnum:]] @ [[:alnum:]] [-a-zA-Z0-9_%\.\-]* \. [[:alpha:]] {2,})`
5. `(tel[\.]?*\: ) | (telnr[\.]?*\: ) | (telefoon[\.]?*\: ) | telefoonnummer[\.]?*\: [ ]?`
6. `(fax[\.]?*\: ) | (faxnr[\.]?*\: ) | (faxnummer[\.]?*\: )`

Tijdens het testen bleek de regexp voor contactpersoon het minste juiste resultaten te geven. Dit is deels te verklaren doordat er zoveel notatiemogelijkheden zijn. Daarnaast kunnen afkortingen ook overeenkomen met andere afkortingen. Bijvoorbeeld B.V. kan worden gezien als voorletters, terwijl het ook de afkorting kan zijn van Besloten Vennootschap. In figuur 56 ziet u het resultaat van het testen van de gehele software agentscript. In figuur 57 ziet u de details van de vacature die zijn geëxtraheerd door de agent.

titel	bedrijf	contactpersoon	datum	matchscore
ACCOUNT MANAGER ENDUSER MARKET		Mevrouw I. Pattijn	09/06/2004	34.91 %
MICROSOFT PROGRAMMEUR	Egency B.V.	EsyRemote	09/06/2004	35.38 %
SOFTWARE TESTERS	Centric Locus BV	L. Het	09/06/2004	36.79 %
SENIOR SOFTWARE ENGINEERS	Centric Locus BV		09/06/2004	35.38 %
SUPPORTMEDEWERKER			09/06/2004	36.32 %
COMMERCIEEL MEDEWERKER BINNENDIENST/VERKOOPLEIDER	De Infotheek Groep N.V.		09/06/2004	44.81 %
WEBDESIGNER M/V			09/06/2004	30.66 %
HELPDESK MEDEWERKER		OmniHis	09/06/2004	35.38 %
CUSTOMER & TECHNICAL SUPPORT MWR ICT	Callic BV	T. Leermomenten	09/06/2004	33.49 %
ACCOUNT MANAGER	Egency B.V.	De heer H. de Boer	09/06/2004	41.51 %
CHIEF TECHNOLOGY OFFICER	Cyborg Webdesign VOF		09/06/2004	37.74 %
ELECTRONICUS OP HBO-NIVEAU M/V	Addware BV		09/06/2004	41.98 %
STARTENDE IT-HELD			09/06/2004	32.08 %
SECRETARIEEL MEDEWERKER M/V			09/06/2004	39.15 %
ACCOUNT MANAGERS (M/V)	ATPS B.V.		04/06/2004	36.79 %
SECRETARESSE			04/06/2004	36.79 %
TELESALES MEDEWERKER M/V			04/06/2004	33.49 %
INTERNET / WEB ONTWIKKELAAR	Pharmon BV	JavaScript	04/06/2004	38.21 %
SYSTEEM-/NETWERKBEHEERDER M/V		B. Bij	04/06/2004	34.43 %
SOFTWARE ONTWIKKELAAR M/V		P.J. Oudweg	04/06/2004	33.49 %
FINANCIEEL ADMINISTRATEUR	Astra Benelux BV	de heer T. Meulenberg	04/06/2004	41.51 %
ACCOUNTMANAGER LOKALE OVERHEID		dhr. S. Stellingwerff	04/06/2004	40.09 %
ACCOUNTMANAGER (M/V)	Plusine B.V.	dhr. H.M. Pomper	04/06/2004	44.81 %
STARTENDE IT-HELD			04/06/2004	33.49 %
CALL CENTER AGENTS			04/06/2004	38.21 %
		StorageMinds	04/06/2004	38.68 %
HELPDESKMEDEWERKERS			04/06/2004	35.38 %
CALL CENTER MEDEWERKERS			04/06/2004	35.38 %
IN-HOUSE SALES (M/V)		StorageMinds	04/06/2004	38.68 %
SOFTWARE ONTWIKKELAAR/ANALIST (M/V)			04/06/2004	37.74 %

Figuur 56 – Lijst met gevonden vacatures

titel:	ACCOUNT MANAGER
Bedrijf:	Agency B.V.
Vacature:	ACCOUNT MANAGER
Agency B.V.

Standplaats
Maarn

Functieomschrijving
Wegens groei zijn wij op zoek naar een Account Manager. Als Account Manager ben je zelf verantwoordelijk om nieuwe klanten in de markt te signaleren en deze binnen te halen. Het sluiten van een order is hier het kernwoord. Je bent zelf verantwoordelijk voor de marktbewerking.

Functie-eisen
Jouw profiel • HBO/WO werk- en denkniveau) • Ervaring in ICT direct sales • Resultaatgericht • Doorzettingsvermogen en stressbestendig • Uitstekende communicatieve vaardigheden • Gedreven, enthousiaste en representatieve instelling • Hoge mate van flexibiliteit en verantwoordelijkheid • Zelfstandig kunnen werken en kennis hebben van het detacheren van ICT professionals .

Gevraagde opleidingen
HBO/WO

Arbeidsvoorwaarden
Goede primaire en secundaire arbeidsvoorwaarden.Telefoon en PD&LeaseautoUitstekende bonusregeling

Solliciteren?
De
Contactpersoon:	De heer H. de Boer
E-mailadres:	h.deboer@agency.nl
Telefoonnummer:	0343444626

Figuur 57 – Vacaturedetails geëxtraheerd door de agent

Ik had echter geen tijd meer om het project af te maken. Mijn projecttijd zat erop. Ik heb de software agent in zoverre af kunnen maken dat mijn bedrijfsmentor en de opdrachtgever de geëxtraheerde data wel kunnen gebruiken omdat het al in de database is opgeslagen. Het enige wat er nog mist aan de pilot is dat het geen publiceermogelijkheden heeft. Doordat de pilot “software agent” nog niet af was, heb de pilot ook niet in kunnen voeren. Wat betreft de systeemdokumentatie over de agent, heb ik weinig op papier achtergelaten. Wel heb ik veel kleine documenten digitaal over de software agent. Daarnaast heb ik tijdens het ontwikkelen van de pilot zoveel mogelijk ingewikkelde blokken script van commentaar voorzien.

Ik heb gelukkig wel een afspraak kunnen maken betreffende het afmaken van de Software Agent. Indien ik zou slagen voor het afstuderen, dan kan ik in de toekomst de Software Agent afmaken.

7. Procesevaluatie

In dit hoofdstuk komt mijn terugblik over het proces van het project aanbod. Laat ik beginnen bij het begin. Bij de opdrachtformulering merkte ik al snel op dat dit niet een “gewone” opdracht zou zijn. De complete opdracht was voor mij in het begin nog niet helemaal duidelijk, maar de kern van de opdracht lag wel op een hoger niveau. En dat laatste sprak mij enorm aan. Ik heb voor deze opdracht zelfs een andere opdracht laten vallen. Zo enthousiast en geïnteresseerd was ik in de opdracht.

De fase waarin ik de definitiestudie ging maken, was voor mij een zeer belangrijke fase. Niet alleen vanuit de optiek van het project, maar ook in intellectueel opzicht. Het merendeels van het project bestond uit het installeren, configureren en programmeren van producten en programmeertalen waarvan ik nauwelijks iets van af wist. Gelukkig waren het juist die aspecten waarvan ik graag heel veel over wilde leren. Ik denk ook dat ik juist door deze leergierigheid, meer en meer bedreven raakte met het project. Zo kan ik voor mezelf zeggen dat ik de inventarisatie van de voor en nadelen van MySQL en MSSQL goed heb gedaan. Ik heb hierover veel kennis opgedaan en deze kennis heb ik goed kunnen overbrengen aan mijn collega's. Met talen als XML, SOAP, WDDX en WSDL heb ik nog nooit geprogrammeerd, maar ze interesseerde me wel enorm. Ik denk dat ik daarom de inventarisatie van de eventueel te gebruiken programmeertalen goed heb gedaan. Hoe gedreven je ook bent, het analyseren en inventariseren kan je niet goed genoeg doen. Dit heb ik ook zelf ook gemerkt. Hoe beter je de eisen en wensen in het begin in kaart brengt des te gemakkelijker je het bij het ontwikkelen ervan af brengt. Dit was een zeer goed leerpunt voor mij geweest. Zeker gezien het feit dat ik bij de ontwikkeling van de pilot “software agent” moest overschakelen naar een evolutionair iteratiestrategie. Als ik de eisen strakker en nog duidelijker had geformuleerd, dan had ik dat kunnen voorkomen en wellicht de software agent sneller kunnen ontwikkelen. Immers, door het steeds weer itereren van de definitiestudie bevorderde dat de snelheid waarmee de agent ontwikkeld werd niet en nam de zekerheid betreffende het eindresultaat ook niet toe.

Tevens realiseer ik me dat de niet strak geformuleerde eisen en wensen ook te maken heeft met het feit dat ik mijn interview met de opdrachtgever niet goed heb voorbereid. Die had ik veel beter moeten voorbereiden, aangezien ik ervoor had gekozen om geen workshops te houden. Het interview was mijn enige formele vorm van informatie vergaren. Daarnaast heb ik ervaren dat informatie via informele wegen niet altijd de beste methode is. Informatie dat even tussendoor wordt verteld, kan je ook weer snel vergeten. Echter, bij het houden van een interview ben je er ook echt toegespitst. De kans dat je het snel vergeet is veel kleiner. Ik heb van deze fout geleerd.

Op het gebied van implementatiemodellen heb ik het achteraf gezien ook beter kunnen doen. Ik had nog beter moeten nadenken over de gevolgen bij het overzetten van een database naar een andere database. Immers, een verandering in het fysieke implementatiemodel kan grote gevolgen hebben en uiteindelijk ook heel veel tijd kosten. En dat heb ik geweten ook. De fout die ik hier heb gemaakt heeft me uiteindelijk de kop gekost met betrekking tot het gehele project. Wat ik hieruit heb geleerd is, dat je pas moet gaan ontwikkelen indien je zeker weet dat het fysiek implementatiemodel voor honderd procent goed opgezet is. Daarnaast heb ik ook geleerd dat je beter ook nog eerst kan navragen of er

in het verleden nog problemen zijn geweest met bepaalde datatypen. Dan kan je hiermee ook rekening houden bij het ontwikkelen van de database.

Wat betreft de andere modellen en diagrammen, heb ik achteraf gezien beter ook geen Statechart diagram kunnen maken, omdat dit diagram mij en mijn opdrachtgever uiteindelijk niet de duidelijkheid verschaftte zoals het zou moeten. Ik heb hiervan geleerd dat je alleen de modellen en diagrammen moet maken, waarbij jezelf maar vooral ook je opdrachtgever er wijzer van wordt.

Met een terugblik op het pilotplan denk ik dat ik de juiste keuzes heb gemaakt betreffende de indeling van de pilots. Ook vind ik dat ik de pilots met de juiste redenen goed heb ingedeeld. Ik heb hier uiteindelijk veel profijt van gehad, omdat ik geen enkel moment het overzicht uit het oog verloor. Juist doordat de meeste pilots compact waren ingedeeld, kon de verloren tijd enigszins weer inhalen omdat ik de pilotdelen door die indeling sneller kon installeren, configureren en programmeren.

Ik vond het incrementele opleverkarakter die de eerste twee pilots hadden ook erg prettig, omdat de testen en de tussentijdse validatie gerichter input gaven voor een iteratie. Toch vond ik het evolutionair ontwikkelen ook heel prettig. De “whitebox” testmethode kwam in deze ontwikkelstrategie namelijk zeer goed van pas.

Het leren van Coldfusion is mij gelukkig gemakkelijk afgegaan. Ik had deze taal al snel onder de knie en heb zelfs bij het ontwikkelen van de software agent in de niet standaard methode geprogrammeerd. Het bedenken van eigen theorieën en die vervolgens toepassen bij het evolutionair ontwikkelen beviel mij erg goed. Redeneren en nog eens redeneren. En dan uiteindelijk je bedachte theorie met programmeren verwezenlijken. De vrijheid die ik hierin had gekregen vond ik daarom ook zeer prettig.

Uiteindelijk heb ik de pilot software agent niet helemaal kunnen afronden in verband met tijdgebrek. Daarnaast heb ik ook geen Dit had ik voornamelijk te danken aan mezelf doordat ik de eisen en wensen niet strak genoeg had gedefinieerd. Daarnaast hebben de aanpassingen in het fysiek implementatiemodel mij ook veel tijd gekost. Ik baal hier enorm van en ik heb hier gelukkig van geleerd.

Tot slot moet ik zeggen dat ik dit zeer leerzaam en geslaagd project vind. Ondanks dat de pilot software agent niet helemaal af is, heb ik wel twee met succes pilots geïnstalleerd, geconfigureerd en geprogrammeerd en uiteindelijk geïmplementeerd. En door het testen zijn deze twee producten ook kwalitatief gezien goed. Als ik terug kijk op het gehele afstudeerproces vind ik dat ik het er goed vanaf heb gebracht. Ik heb goede keuzes gemaakt op punten die van groot belang waren voor het proces zoals de ontwikkelstrategie. Ik heb uiteraard ook minder goede keuzes gemaakt. Echter, ik weet wel hoe ik het anders had kunnen aanpakken wat mij zelfverzekerd en positief stemt voor toekomstige projecten.

8. Productevaluatie

8.1 Plan van aanpak

Het plan van aanpak voor het gehele project was voor mij een goede houvast. De vastlegging van cruciale zaken in dit document hebben mij geholpen tijdens het project. Ik ben dan ook tevreden met dit product

8.2 Definitiestudie Software Agent

De definitiestudie was voor mij een goed naslag werk. Ook al was het niet groot van omvang, ik heb dit document herhaaldelijk moeten raadplegen. Dat wil dus wel zeggen dat het goed geschreven is. Echter, de volgende keer neem ik alleen modellen en diagrammen op in de definitiestudie die ik echt acht nodig te hebben.

8.3 Testplan/Testrapport

Ik ben uitermate tevreden over het testrapport. Met dit rapport heb ik de server goed kunnen configureren. Daarnaast zal ik in de toekomst dit rapport ook gebruiken als voorbeeld bij het opstellen van nieuwe testrapporten.

8.4 De server

De server is een goed geconfigureerde server geworden van hoge kwaliteit. Dit valt ook weer af te leiden aan het feit dat de server nog geen een keer is gecrashed na de implementatie. De server is snel en is nog goed schaalbaar ook. Wat ik echter wel anders had willen doen, is om de hardeschijf minimaal in te delen in twee partities, zodat data en programma's gescheiden kunnen blijven.

8.5 Software Agent

De Software Agent heb ik helaas net niet kunnen afronden. De technologie daarachter en de codering vind ik uitermate van hoge kwaliteit. Dit komt mede door het veelvuldig testen en valideren van de code. Een punt dat ik wel zou willen veranderen aan de agent is het duidelijk weergeven van zijn status tijdens het zoeken en zijn status van werkzaamheden. De output dat nu op het scherm verschijnt was eigenlijk alleen bedoeld voor het “whitebox” testen.

8.7 CMS Software Agent

Het content management gedeelte van de software agent vind ik prima passen binnen het intranet van i-concept. Zowel qua kleur als indeling is het helemaal in overeenstemming met het intranet. Ook de navigatie is duidelijk en consistent. Wat ik wel zou willen veranderen zijn de formulierknoppen. Ik zou deze liever in links willen veranderen, omdat je dan de opmaak ook nog kunt verfraaien.

Literatuur- en internetverwijzingen

Literatuur

IAD – het evolutionair ontwikkelen van informatiesystemen
R.J.H. Tolido, 1996

Databasesystemen voor de praktijk
Prof. Dr. J.A. Van den Bulcke, 1997

Coldfusion 5.0 Web Applicatoion Construction Kit
Ben Forta, 2001

Internetverwijzingen

MySQL.com	http://www.mysql.com
SQL Reference	http://www.cfdev.com/resources/discussion/.cfm/id/7
CfHub	http://cfhub.com/advanced/cfhttp/copy.cfm
Webmonkey	http://hotwired.lycos.com/webmonkey/99/15/index3a_page5.html?tw=authoring
Regexlib.com	http://regexlib.com/
IIS Tips&Resources	http://www.iis-resources.com/modules/news/print.php?storyid=19
MSDN	http://msdn.microsoft.nl

Bijlagen



Software Agent

Definitiestudie

Philip Boekholt

Inhoudsopgave

1 Inventarisatie Website	5
2 Eisen Software Agent.....	9
3 State Diagram	10
4 Activity Diagram	11
5 Taakdiagrammen.....	12
6 EER Model.....	14
7 Relatieel model.....	15
Tabel SagAgent	15
Tabel SagFilterTerm.....	15
Tabel SagSearchTerms.....	15
Tabel SagSearchArea.....	15
Tabel SagSearchHistory.....	15
Tabel SagVacatureFound.....	15
8 Implementatiemodel Software Agent	16
Tabel SagAgents.....	16
Tabel SagFilterTerm.....	16
Tabel SagSearchTerms.....	16
Tabel SagSearchArea.....	17
Tabel SagSearchHistory.....	17
Tabel SagVacatureFound.....	17
9 Pilotplan	18
Pilots	18
Pilotdelen	18
Pilotstrategie	19

1 Inventarisatie Website

Table	Field	Wijziging
Cv	Geboortedatum	Date/time -> date
Cv	Datum_invoer	Date/time -> date
Opleiding	Begin_opleiding	Date/time -> date
Opleiding	Einde_opleiding	Date/time -> date
Ervaring	Begin_dienstbetrekking	Date/time -> date
Ervaring	Einde_dienstbetrekking	Date/time -> date

Alter Pagina's	Wijziging in database
Alter_tblcv.cfm	Tbl Cv -> add kolom gebdatum, kolom datumvoer
Alter_tblnevenactiviteit.cfm	Tbl nevenactiviteit -> add kolom begindienstbetrekking, einddienstbetrekking
Alter_tblopleiding.cfm	Tbl opleiding -> add kolom beginopleiding, eindeopleiding
Alter_tblwerkervaring.cfm	Tbl ervaring -> add kolom begindienstbetrekking, einddienstbetrekking
Alter_tblvacature.cfm	Tbl vacature -> add kolom reagerenvoor, tonenvanaf, tonentm
Alter_tbluser.cfm	
Alter_tblvacs.cfm	LET OP! Allebei 2 DATETIME gebruiken!
Alter_tblvacviews.cfm	

Pagina	Wijziging
Bannerinsert.htm	DSN internetworks verwijderd
Quickcv\index.cfm	
Aanmeldform.cfm	
Form_cv_werkervaring_submit.cfm	
Form_cv_werkervaring.cfm	Nieuwe controle erbij/ kleiner dan 8 getallen is niet goed, het aantal dagen groter dan 31 niet goed, het aantal maanden groter dan 12 is ook niet goed
Form_cv_opleiding.cfm	Datum -> gewijzigd van echte datum naar numeriek veld -> datum gecreeerd uit een numeriek veld!
Form_cv_opleiding_submit.cfm	Datum -> gewijzigd van echte datum naar numeriek veld -> datum gecreeerd uit een numeriek veld!
Form_cv_werkervaring.cfm	Idem dito
Form_cv_werkervaring_submit.cfm	Idem dito
Nevenactiviteiten.cfm	Idem dito
Nevenactiviteiten_submit.cfm	Idem dito
Form_cv2.cfm	Idem dito
Form_cv_submit.cfm	Idem dito
Quickcv\Index.cfm	Opslaan van data -> numeriek!
Checkdate.cfm	Idem dito
Company_remove_confirm.cfm	Idem dito
Company_result.cfm	
Company_result_admin.cfm	

Overzicht_cv_print.cfm	Outfile: verwijzing e:\ --> c:\ Datum -> numeriek!
Overzicht_cv_loop.cfm	
Overzicht_cv_recruiter.cfm	
Overzicht_cv.cfm	Datum -> gewijzigd van echte datum naar numeriek veld -> datum gecreëerd uit een numeriek veld!
New-submit_recruit.cfm	
Update_recruiters.cfm	<pre> <cfset tempDate = val(dateformat("#now()- 1#", "yyymmdd"))> <cfquery name="update" datasource="arbeidsmarkt"> UPDATE users SET toeglevel = 'grd' WHERE usersince in (SELECT usersince FROM Users WHERE toeglevel = 'gr' AND Usersince <= #tempDate#) </cfquery> </pre>
zoek_cv_submit3.cfm	
Vacature.cfm	
Vacature_add.cfm	
Vacature_submit.cfm	
Vacature2.cfm	
Vacature_bekijk_detail.cfm	
Vacature_bekijken.cfm	
Vacature_delete.cfm	

Vacature_publish.cfm	Reagerenvoor, reagerentm aangepast op numerieke datum. Publish naar e -> c
Vacature_publish2.cfm	Idem dito
Vacature_publish_loop.cfm	Publish naar e -> c <pre> <cfif result.reagerenvoor neq 0> <cfset inputDatum = '#result.reagerenvoor#'> <cfset tmpjaar = mid(#inputDatum#,1,4)> <cfset tmpmaand = mid(#inputDatum#,5,2)> <cfset tmpdag = mid(#inputDatum#,7,2)> <cfset showOutput = tmpdag & "/" & tmpmaand & "/" & tmpjaar> <cfelse> <cfset showOutput = "> </cfif> <cfif datecompare(#showOutput#,now()) is -1> </pre>
Vacature_statistiek	Let op datum is hier datum+tijd. Maar aangezien er geen userinput is, kan er weinig fout gaan met datum. Dus hier gebruik maken van datetime
Vacature_aanpassen.cfm	
Vacature_bekijken.cfm	
Vacature_bekijken_all.cfm	Let op: meerder malen het volgende: <pre> (tonentm >= { d '#dateformat("#now()#","yyyymmdd")#'} }) </pre>
Vacature_bedrijf.cfm	<pre> <cfquery name="vacatures" datasource="arbeidsmarkt"> select * from vacature where bedrijf_id = #url.b# AND (tonentm >= { d '#dateformat("#now()#","yyyymmdd")#'} }) ORDER BY tonen_vanaf DESC </cfquery> tonen_vanaf -> tonenvanaf </pre>
Vacatures_view100.cfm	Let op andere db: acomstats <pre> <cfset totalcount='0'> <cfloop query="selvac"> <cfquery name="countviews" datasource="acomstats"> SELECT * from vacviews where vacid=#selvac.vacid# <cfif isdefined('datumselectie')> AND datum BETWEEN #createodbcdate(form.sdatum)# AND #createodbcdate(form.edatum)# </cfif> </cfquery> createodbc -> timestamp </pre>
Zoek_cd_detail-old.cfm	Alle nieuwe datumattributen aanroepen
Zoek_cd_detail.cfm	Idem dito
Zoek_cv_bericht_submit.cfm	E:\ -> c:\

Zoek_cv_submit2.cfm	Datum_invoer -> datuminvoer
Zoek_cv_submit.cfm	Idem dito
Zoek_cv_top10.cfm	Idem dito
Zoek_cv_trefwoord_result	Idem dito
Zoek_cv_trefwoord_result2.cfm	Idem dito
Zoek_cv_trefwoord_result_doethet.cfm	
User_delete_htm.cfm	E:/ -> c:/
User_delete_fromemail_ok.cfm	Idem dito
User_delete	Datum_invoer -> datuminvoer Tonen_vanaf -> tonenvanaf

2 Eisen Software Agent

Functionele eisen

- De software agent moet vacatures van het Internet kunnen opslaan in een formaat dat gebruikt kan worden op arbeidsmarkt.com
- De software agent moet onderscheid kunnen maken tussen goede en slechte vacatures. Met een goede vacature wordt bedoeld dat een vacature geschikt is om op arbeidsmarkt.com te plaatsen.
- De software agent moet niet al te moeilijk te configureren zijn
- Gevonden vacatures moeten opgeslagen kunnen worden in de database
- De gevonden vacatures moeten snel benaderbaar zijn voor de medewerkers
- Plaatsing van gevonden vacatures door de software agent moet niet moeilijk zijn voor medewerkers
- Er moet automatisch back-ups gemaakt kunnen worden van de vacatures
- Er moet automatisch back-ups gemaakt kunnen worden van de database

Interface eisen

- Het configuratiescherm van de agent moet overzichtelijk zijn
- Het configuratiescherm van de agent moet gemakkelijk te bedienen zijn
- De software agent moet qua kleurgebruik passen binnen de huidige intranet
- De schermindelingen en structuur van het configuratiescherm moeten bekend voorkomen bij de gebruiker

Performance eisen

- De software agent mag de server niet te zwaar belasten. Het cpu verbruik mag niet langer zijn dan 10 seconden.
- De software agent mag andere websites niet negatief beïnvloeden qua performance
- De data die de software agent gevonden heeft, moet snel uit de database gehaald kunnen worden. De database moet een snellere dataoverdracht hebben dan MS Acces. De overdracht mag niet langer dan 5 seconden duren.
- De database moet een grotere opslagcapaciteit aan kunnen zonder dat daarbij de response tijd exponentieel achteruit gaat.
- De server waarop de database draait moet optimaal geconfigureerd zijn voor grote bezoekersaantallen. De response tijd mag niet exponentieel achteruit gaan.

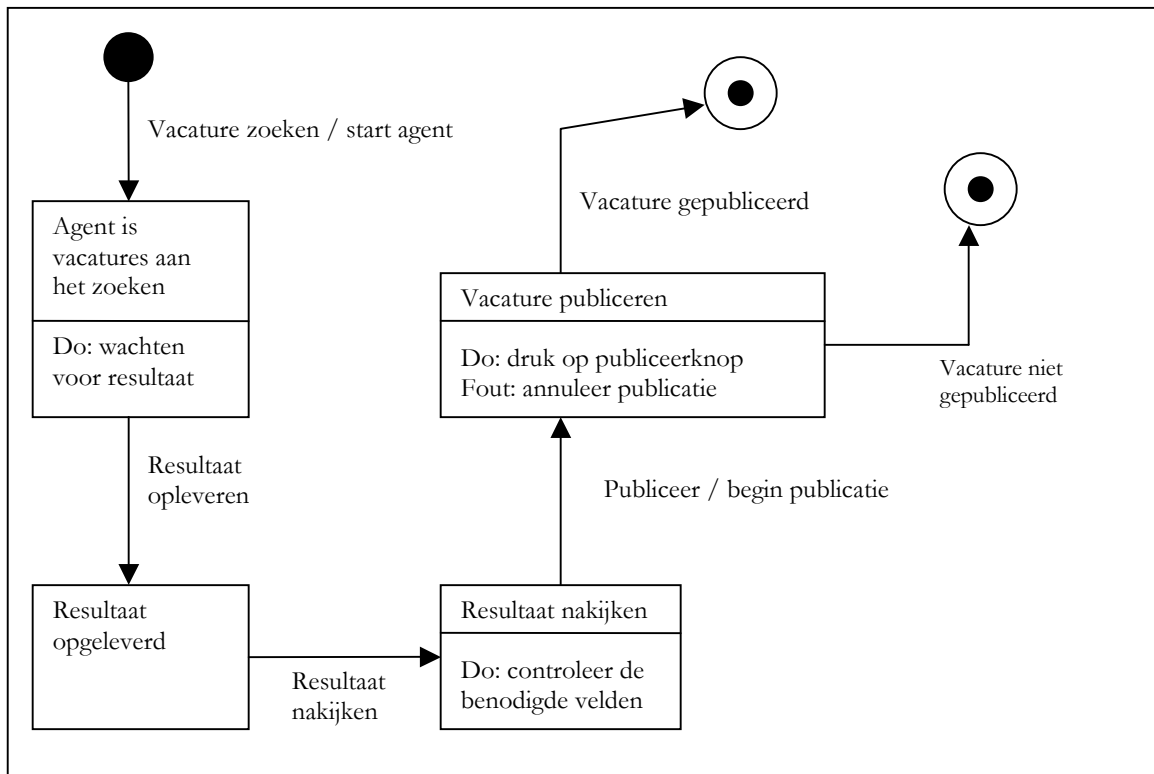
Operationele eisen

- De software agent dient herbruikbaar te zijn voor andere zoekacties

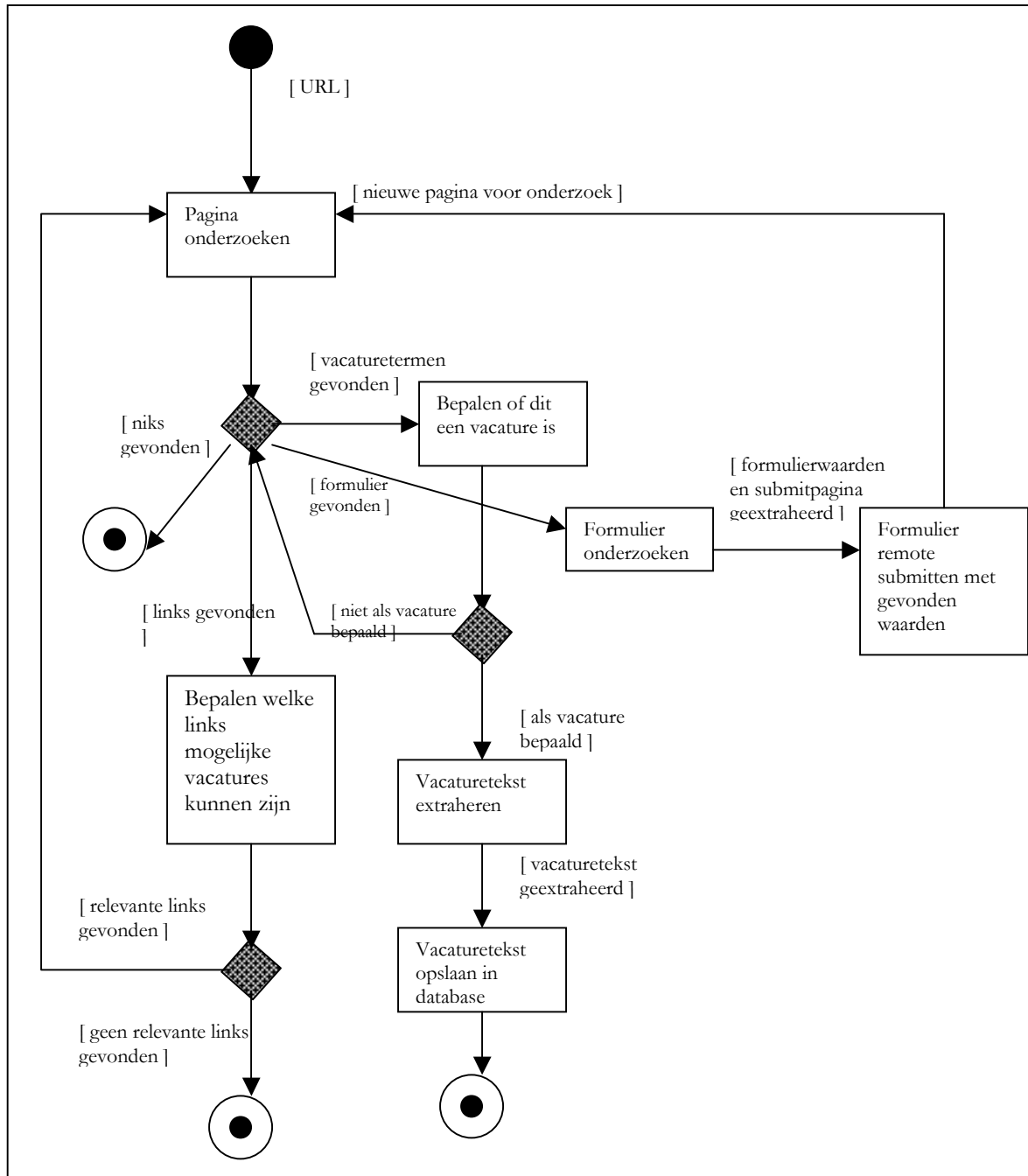
Beveiligingseisen

- De database dient beveiligd te zijn en mag alleen via interne kanalen gemodificeerd worden zijn
- De server dient beveiligd te zijn tegen buitenstaanders en virussen
- De software agent dient beveiligd te zijn en mag alleen door eigen medewerkers aangestuurd en gemodificeerd worden

3 State Diagram

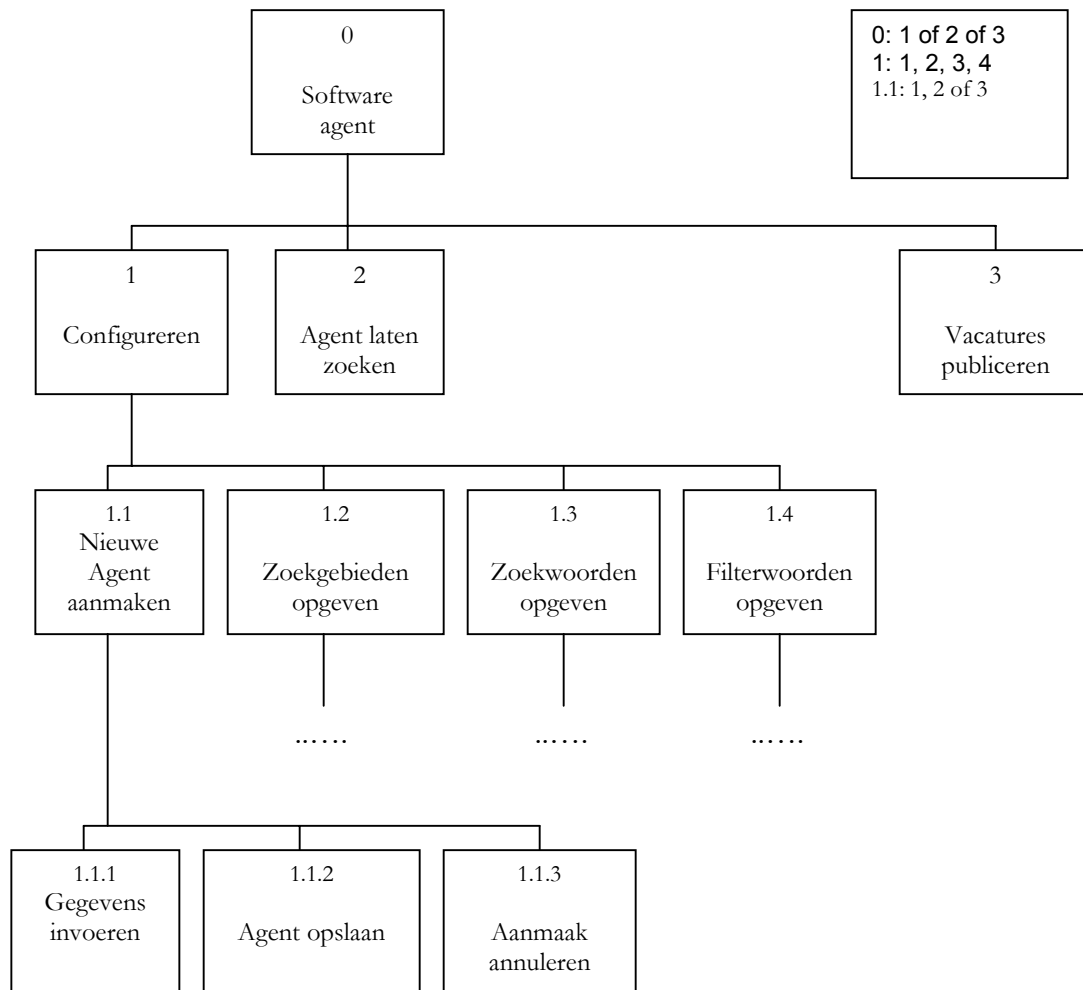


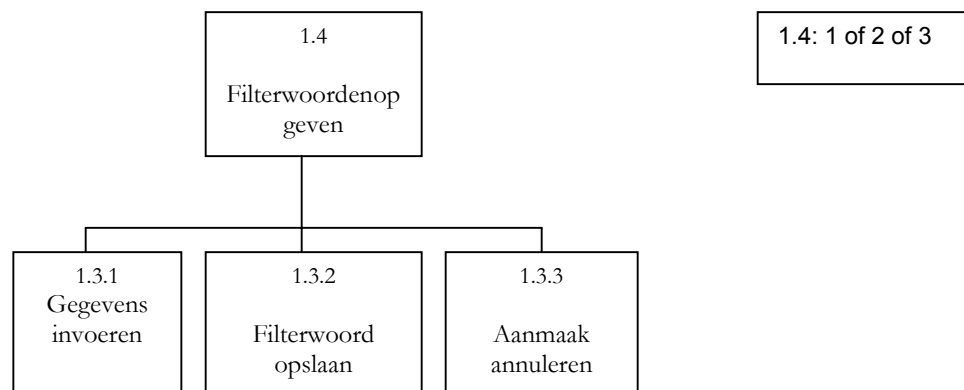
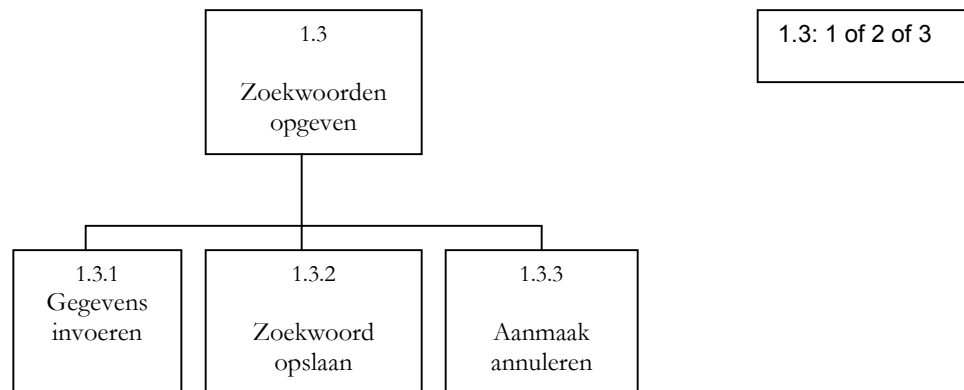
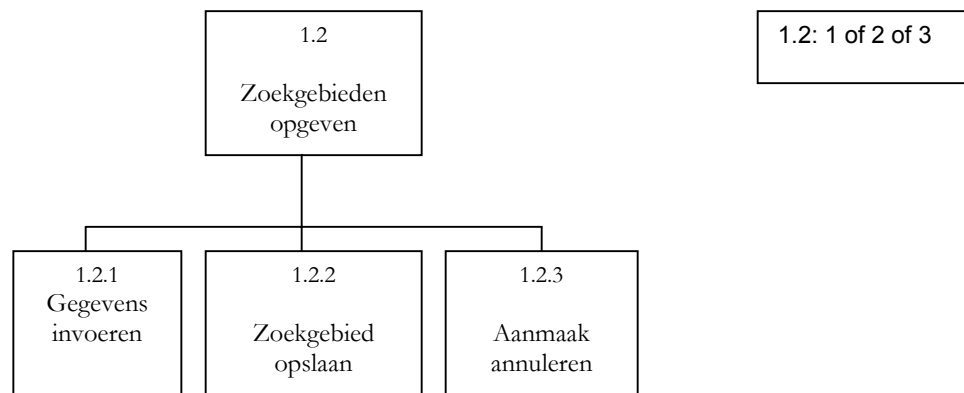
4 Activity Diagram



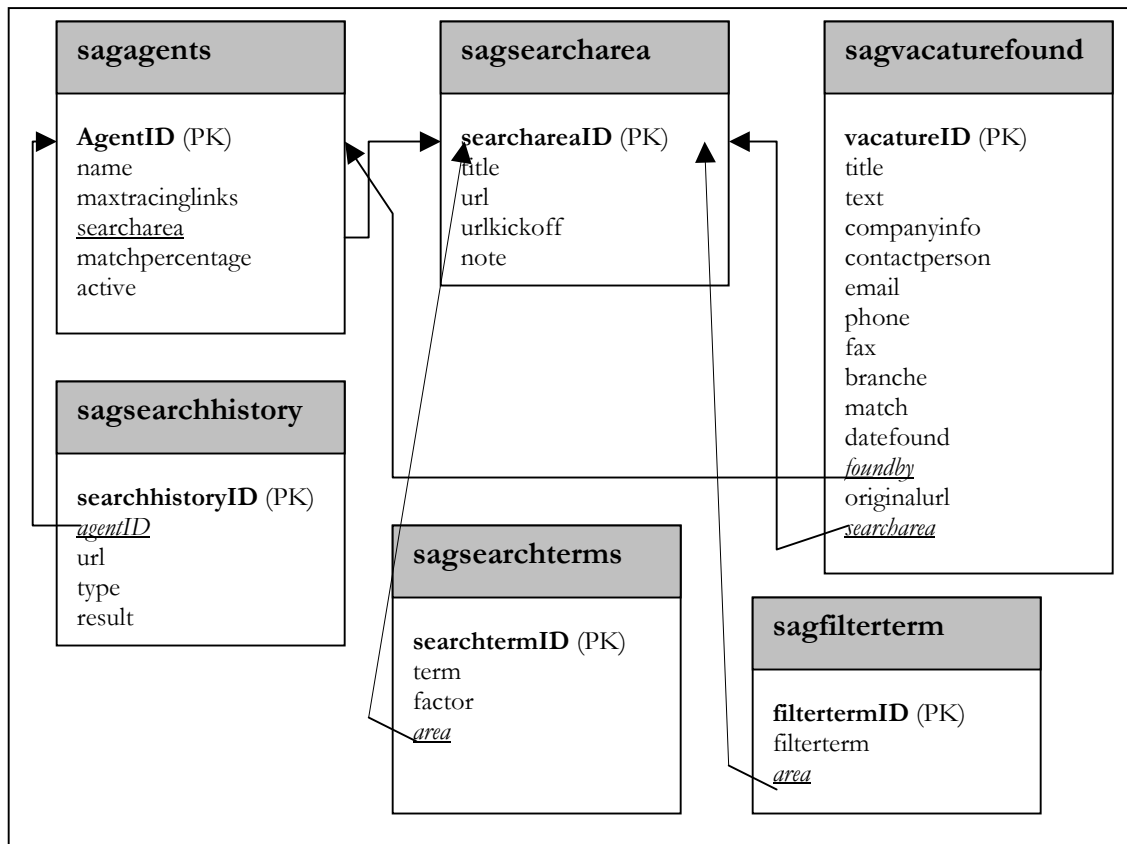
5 Taakdiagrammen

De software agent configureren





6 EER Model



7 Relationeel model

Tabel SagAgent

SagAgents (**agentID**, name, maxtracinglinks, matchpercentage, active, searcharea)

- Primaire sleutel is agentID
- Vreemde sleutel is searcharea refereert aan tabel sagsearcharea

Tabel SagFilterTerm

SagFilterTerm (**filtertermID**, filterterm, area)

- Primaire sleutel is filterterm
- Vreemde sleutel is area refereert aan tabel SagSearchArea

Tabel SagSearchTerms

SagSearchTerms (**searchtermID**, term, factor, area)

- Primaire sleutel is searchtermID
- Vreemde sleutel is area refereert aan tabel SagSearchArea

Tabel SagSearchArea

SagSearchArea (**searchareaID**, title, url, urlkickoff, note)

- Primaire sleutel is searchareaID

Tabel SagSearchHistory

SagSearchHistory (**searchHistoryID**, agentID, url, type, result)

- Primaire sleutel is searchhistoryID
- Vreemde sleutel is agentID refereert aan tabel sagAgents

Tabel SagVacatureFound

SagVacatureFound (**vacatureID**, title, text, bedrijf, contactperson, email, telefoon, fax, datefound)

- Primaire sleutel is vacatureID
- Vreemde sleutel foundby refereert aan tabel sagAgents
- Vreemde sleutel searcharea refereert aan tabel sagSearchArea

8 Implementatiemodel Software Agent

Tabel SagAgents

```
DROP TABLE IF EXISTS sagagents
```

```
CREATE TABLE sagagents (  
    agentID SMALLINT(6) auto_increment NOT NULL,  
    name VARCHAR(50) NULL,  
    maxtracinglinks INT(11) NOT NULL  
    searcharea INT(11) NOT NULL,  
    matchpercentage DOUBLE NOT NULL,  
    active smallint(6) NOT NULL  
    PRIMARY KEY (agentID)  
)
```

Tabel SagFilterTerm

```
DROP TABLE IF EXISTS sagfilterterm
```

```
CREATE TABLE sagfilterterm (  
    agentID SMALLINT(6) auto_increment NOT NULL,  
    name VARCHAR(50) NULL,  
    area SMALLINT(11) NOT NULL  
    PRIMARY KEY (agentID)  
    KEY (area)  
)
```

Tabel SagSearchTerms

```
DROP TABLE IF EXISTS sagsearchterms
```

```
CREATE TABLE sagsearchterms (  
    searchtermID SMALLINT(6) auto_increment NOT NULL,  
    term VARCHAR(100) NULL,  
    factor INT(11) NOT NULL,  
    area SMALLINT(11) NOT NULL  
    PRIMARY KEY (agentID)  
    KEY (area)  
)
```

Tabel SagSearchArea

```
DROP TABLE IF EXISTS sagsearcharea
```

```
CREATE TABLE sagsearcharea (  
    searchareaID SMALLINT(6) auto_increment NOT NULL,  
    title VARCHAR(100) NULL,  
    url VARCHAR(100) NULL,  
    urlkickoff VARCHAR(100) NULL,  
    note LONGTEXT NULL  
    PRIMARY KEY (searchareaID)  
)
```

Tabel SagSearchHistory

```
DROP TABLE IF EXISTS sagsearchhistory
```

```
CREATE TABLE sagsearchhistory (  
    searchhistoryID SMALLINT(6) auto_increment NOT NULL,  
    agentID SMALLINT(6) NOT NULL,  
    url VARCHAR(100) NULL,  
    type INT(11) NULL,  
    result VARCHAR(100)  
    PRIMARY KEY (searchhistoryID)  
    KEY (agentID)  
)
```

Tabel SagVacatureFound

```
DROP TABLE IF EXISTS sagvacaturefound
```

```
CREATE TABLE sagvacaturefound (  
    vacatureID BIGINT(11) auto_increment NOT NULL,  
    title VARCHAR(100) NULL,  
    bedrijf VARCHAR(100) NULL,  
    text LONGTEXT NULL,  
    contactpersoon VARCHAR(100) NULL,  
    email VARCHAR(100) NULL,  
    telefoon VARCHAR(100) NULL,  
    fax VARCHAR(100) NULL,  
    datefound INT(8) NOT NULL,  
    PRIMARY KEY (vacatureID)  
)
```

9 Pilotplan

Pilots

<i>Pilot 1</i>	De server
<i>Pilot 2</i>	De database
<i>Pilot 3</i>	Software Agent

Pilotdelen

Pilot 1	De server
Deel #1	De Windows 2000 server
Deel #2	De webserver
Deel #3	De Coldfusion server
Deel #4	AVG virusscanner
Deel #5	Double image
Deel #6	Remote Desktop

Pilot 2	De database
Deel #1	Installatie van MySQL
Deel #2	Configuratie van MySQL
Deel #3	Het overzetten van data
Deel #4	De website arbeidsmarkt.com gereed maken voor MySQL
Deel #5	De tabellen voor de software agent toevoegen aan de MySQL database

Pilot 3	Software Agent
Deel #1	De tabellen
Deel #2	De configuratieschermen
Deel #3	De software agent

Pilotstrategie

<i>Pilot 1</i>	Incrementeel opleveren
<i>Pilot 2</i>	Incrementeel opleveren
<i>Pilot 3</i>	Incrementeel opleveren Wijziging: Evolutionair ontwikkelen

Incrementeel opleveren:



Evolutionair ontwikkelen:

