

# Ontwerp van een Group Decision Support System

Master Software Engineering  
Universiteit van Amsterdam

Jakob Ooms  
0600156

31 augustus 2006

Afstudeerdocent	: Drs. Hans Dekkers
Stagebegeleider	: Dr. Dirk G.A. Kenis
Opdrachtgever	: Memori
Publicatiestatus	: Openbaar



## Voorwoord

Dit proefschrift is het resultaat van het masterproject van Jakob Ooms voor de éénjarige opleiding Master Software Engineering aan de Universiteit van Amsterdam. Gedurende het onderzoekstraject en het uitwerken van dit onderzoeksrapport heb ik mogen genieten van professionele ondersteuning van een aantal personen, dewelke ik via deze weg graag wil bedanken.

Hans Dekkers, voor de snelle en opbouwende kritiek die u hebt geven bij tussentijdse evaluaties. Dirk Kenis, voor de levendige discussies en de vrijheid om dingen te mogen ontdekken en onderzoeken. Het hele Memori team, voor de ondersteuning, de sprankelende ideeën en de ontspannende humor. Bedankt.

## Samenvatting

KnoSoS (Knowledge Sharing over Social Software) is een onderzoeksproject dat wil nagaan hoe sociale software [Bijlage 1, TheRiseOfSocialSoftware] samenwerking en kennisdeling kan bevorderen, ondersteunen en stimuleren tussen zelfstandig opererende (deel)organisaties [KnoSoS Home]. Om het onderzoeksproject te ondersteunen wordt een systeem opgezet waarin de eindgebruikers hun kennis kunnen delen.

Tijdens het delen van deze kennis moet het mogelijk zijn om op een gestructureerde manier besluiten te kunnen nemen in groepsverband. Omdat de betrokken onderzoekscentra een uitgebreid arsenaal van besluitvormingsprocessen wensen te ondersteunen, is besloten om een apart platform op te zetten dat hiervoor de gepaste ondersteuning kan bieden. Dit platform moet onafhankelijk kunnen fungeren van het KnoSoS-systeem, maar dient uiteraard wel een koppeling te voorzien tussen beide systemen.

Dit proefschrift bekijkt hoe dat dit platform opgezet moet worden. Hiervoor wordt gekeken naar de gemeenschappelijke kenmerken van besluitvormingsprocessen en hoe dat deze gemodeleerd kunnen worden in het platform. Een bestaand systeem, dat reeds gebruikt wordt door één van de onderzoekscentra, wordt onderzocht op sterke en zwakke punten, zodat dezelfde fouten in het platform vermeden kunnen worden.

Om de exacte eisen en wensen vast te stellen ten aanzien van het platform, wordt een requirements document opgesteld. Omdat het platform naast asynchrone besluitvormingsprocessen [Bijlage 1] ook synchrone besluitvormingsprocessen zal ondersteunen, wordt gekeken naar een techniek om de data overdracht tussen de eindgebruikers en de webserver te laten verlopen zonder dat de eindgebruiker wordt geconfronteerd met een webpagina die in zijn geheel moet herladen.

Aan de hand van al deze resultaten is het mogelijk om een software architectuur op te zetten. Deze software architectuur wordt vervolgens geïmplementeerd in een prototype, waarna een validatie plaats vindt van de opgestelde software architectuur.

# Inhoudsopgave

<b>1. ONDERZOEKSVRAAG.....</b>	<b>5</b>
<b>2. CONTEXT VAN HET PROBLEEM .....</b>	<b>6</b>
2.1. GESTRUCTUREERDE BESLUITVORMINGSOMGEVING VOOR HET KNOSES-SYSTEEM .....	7
2.2. ONDERZOEK NAAR BESLUITVORMINGSPROCESSEN IN FUNCTIE VAN KENNISDELING .....	7
2.3. ONDERZOEK NAAR SYNCHRONE BESLUITVORMINGSPROCESSEN .....	7
<b>3. KADERING VAN ONDERZOEKSONDERDELEN .....</b>	<b>8</b>
<b>4. OPSTELLEN VAN EEN GENERIEK GROEPSBESLUITVORMINGSPROCES.....</b>	<b>9</b>
4.1. ONDERZOEKSAANPAK.....	9
4.2. UITVOERING .....	10
4.3. RESULTATEN.....	11
4.4. VASTGESTELDE EISEN .....	15
<b>5. ANALYSEREN VAN BESTAAND DELPHI SYSTEEM.....</b>	<b>16</b>
5.1. ONDERZOEKSAANPAK.....	16
5.2. UITVOERING .....	16
5.3. RESULTATEN.....	17
5.4. VASTGESTELDE EISEN .....	18
<b>6. ACHTERHALEN VAN DE EISEN VOOR HET PLATFORM.....</b>	<b>20</b>
6.1. ONDERZOEKSAANPAK.....	20
6.2. UITVOERING .....	20
6.3. RESULTATEN.....	22
<b>7. ONDERZOEKEN VAN HET POTENTIEEL VAN AJAX .....</b>	<b>24</b>
7.1. ONDERZOEKSAANPAK.....	24
7.2. UITVOERING .....	24
7.3. RESULTATEN.....	25
<b>8. OPSTELLEN VAN DE SOFTWARE ARCHITECTUUR .....</b>	<b>29</b>
8.1. ONDERZOEKSAANPAK.....	29
8.2. UITWERKING.....	29
8.3. RESULTATEN.....	30
<b>9. UITWERKEN VAN HET PROTOTYPE.....</b>	<b>37</b>
9.1. ONDERZOEKSAANPAK.....	37
9.2. UITVOERING .....	37
9.3. RESULTATEN.....	37
<b>10. VALIDATIE EN EVALUATIE ONDERZOEKSTRAJECT .....</b>	<b>39</b>
10.1. VALIDATIE VAN ONDERZOEKSTRAJECT .....	39
10.2. WAT IS ER BEREIKT .....	40
10.3. WAT GING ER MIS.....	40
10.4. BRUIKBAARHEID VOOR DE OPDRACHTGEVER.....	40
10.5. REFLECTIE OP ONDERZOEKSTRAJECT .....	40
10.6. NIEUWE WEGEN BEWANDELEN.....	41
<b>11. REFERENTIES.....</b>	<b>42</b>
<b>BIJLAGE 1: WOORDVERKLARINGEN EN DEFINITIES.....</b>	<b>44</b>
<b>BIJLAGE 2: OPSTELLEN VAN GENERIEK GROEPSBESLUITVORMINGSPROCES.....</b>	<b>46</b>
<b>BIJLAGE 3: ANALYSEREN VAN BESTAAND DELPHI SYSTEEM.....</b>	<b>51</b>
<b>BIJLAGE 4: PERSONA'S.....</b>	<b>54</b>

<b>BIJLAGE 5: OVERZICHTEN VAN OPGESTELDE EISEN.....</b>	<b>59</b>
<b>BIJLAGE 6: BELANGRIJKSTE EISEN AAN HET PLATFORM.....</b>	<b>64</b>
<b>BIJLAGE 7: HET POTENTIEEL VAN AJAX.....</b>	<b>79</b>
<b>BIJLAGE 8: SCREENSHOTS PROTOTYPE.....</b>	<b>85</b>
<b>BIJLAGE 9: TOELICHTING ONTWIKKELING VIEW.....</b>	<b>93</b>
<b>BIJLAGE 10: VALIDATIE ARCHITECTUUR.....</b>	<b>94</b>

# 1. Onderzoeksvraag

Dit onderzoek heeft als doel een ontwerp te maken voor een Group Decision Support System dat een breed scala van besluitvormingsprocessen ondersteund. Om dit platform te kunnen opzetten is het eerst noodzakelijk om domeinkennis op te doen omtrent besluitvormingsprocessen in groepsverband en hun gemeenschappelijke kenmerken (Onderzoeksvraag 1).

Nadat de eisen en wensen ten aanzien van het platform zijn vast gesteld (Onderzoeksvraag 3a) kan, met behulp van een abstract generiek besluitvormingsproces (Onderzoeksvraag 2), een software architectuur worden gebouwd (Onderzoeksvraag 3c) die de generieke kenmerken van besluitvormingsprocessen ondersteund (Onderzoeksvraag 3).

Om de opgestelde software architectuur te valideren wordt een prototype opgezet waaruit de flexibiliteit van het platform wordt aangetoond (Onderzoeksvraag 3d). Dit prototype dient ook als basis om te onderzoeken in hoeverre dat AJAX (Asynchronous JavaScript and XML) kan bijdrage om synchrone besluitvormingsprocessen te implementeren in het Group Decision Support System (Onderzoeksvraag 3b).

1. Onderzoek wat de gemeenschappelijke kenmerken zijn van verschillende besluitvormingsprocessen in groepsverband en hoe kwaliteit in deze processen bevorderd kan worden.
2. Creeër aan de hand van deze bevindingen een generiek besluitvormingsmodel.
3. Bouw met dit generiek besluitvormingsmodel een Group Decision Support System dat een rijk scala van besluitvormingsprocessen kan ondersteunen, en waarin deze besluitvormingsprocessen kunnen worden aangepast.
  - a. Stel de eisen en wensen vast die aan dit platform worden gesteld.
  - b. Onderzoek hoe dat AJAX kan bijdragen om synchrone besluitvormingsprocessen in dit platform te implementeren.
  - c. Creeër een software architectuur conform de opgestelde eisen en de opgedane kennis van AJAX.
  - d. Ontwerp een prototype dat de opgestelde architectuur implementeert, waarin de flexibiliteit van het platform wordt aangetoond.

## 2. Context van het probleem

KnoSoS (Knowledge Sharing over Social Software) is een onderzoeksproject dat op 1 september 2005 is opgestart en gedurende twee jaar wordt uitgevoerd door Memori, het onderzoekscentrum van de Katholieke Hogeschool Mechelen, in samenwerking met de Vrije Universiteit van Brussel, vakgroep MOSI (Mathematics, Operational research, Statistics and Information systems, applied in human sciences).

Het KnoSoS project wil nagaan hoe sociale software [Bijlage 1, TheRiseOfSocialSoftware] samenwerking en kennisdeling kan bevorderen, ondersteunen en stimuleren tussen zelfstandig opererende (deel)organisaties [KnoSoS Home]. Om dit onderzoek te leiden, wordt een systeem opgezet waarin kennisdeling centraal staat.

Om de eindgebruikers van dit systeem de mogelijkheid te geven om gestructureerd besluiten te laten nemen in groepsverband, is besloten een Group Decision Support System (GDSS) [Bijlage 1] platform op te zetten. Het platform is een webapplicatie dat ondersteuning moet bieden aan participanten [Bijlage 1] om op een interactieve manier ideeën te genereren en besluiten te nemen in groepsverband.

Het GDSS platform zal de onderlinge communicatie en interactie tussen participanten ondersteunen, waardoor deze gedurende het besluitvormingsproces hun gedachten kunnen wisselen of nieuwe ideeën kunnen genereren. Het zijn net deze vormen van interactie tussen de participanten die kennisdeling bevorderen. Het is dan ook belangrijk dat de facilitator [Bijlage 1], de beheerder van een besluitvorming, in staat wordt gesteld om het procesmatige van deze interactie te beheren.

In tegenstelling tot systemen zoals Microsoft Netmeeting, zal het GDSS platform geen gebruik maken van audio- en videokanalen tussen de participanten. Het platform zal gebaseerd zijn op tekstuele interactie, waarbij de participanten hun antwoorden moeten ingeven via webformulieren (zie [Bijlage 1] voor een exacte definitie van "antwoord" in deze context). De nadruk van het GDSS platform ligt ook op het proces dat doorlopen moet worden in een besluitvorming. Bij Microsoft Netmeeting worden de gebruikers in staat gesteld om synchroon te communiceren (met audio en video), maar het systeem heeft geen (expliciete) voorziening in het begeleiden van het proces bij besluitvormingen.

Het platform zal van nul af aan opgebouwd moeten worden (achterhalen van eisen, ontwikkelen architectuur, implementatie,...). Enkele ideeën en concepten kunnen worden overgenomen van een bestaand systeem dat gebaseerd is op het Delphi besluitvormingsproces [Kenis ImprovingGroupDecisions]. Dit systeem werd enkele jaren geleden ontwikkeld als webapplicatie in VisualWorks en werd in 2005 voor een deel opnieuw geïmplementeerd in ASP.NET.

Het GDSS platform zal, in tegenstelling tot de vorige systemen, ondersteuning bieden voor meerdere communicatie processen die erop gericht zijn om in een groep tot efficiëntere besluiten te komen (zoals het "Delphi" proces, de "Nominal Group Technique", de "Method 6-3-5"). Hierbij krijgt de facilitator de mogelijkheid om een bepaald proces te kiezen of om een proces samen te stellen. Bij het opzetten van dit proces bepaalt de facilitator welke vragen en wat voor soort vragen (multiple choice vragen, open vragen) hij wenst te stellen. Het kunnen configureren van deze vragen is een belangrijke meerwaarde van het GDSS platform ten aanzien van de vorige ontwikkelde systemen, omdat zo de facilitator de volledige controle over het proces heeft.

Een ander belangrijk verschil is dat het platform ondersteuning biedt voor zowel asynchrone als synchrone besluitvormingsprocessen. Bij deze synchrone besluitvormingsprocessen [Bijlage 1] wordt realtime informatie uitgewisseld tussen de participanten en de facilitator en tussen de participanten onderling. Op deze manier kan een interactiever besluitvormingsproces worden opgebouwd. Bij asynchrone besluitvormingsprocessen krijgt



de participant de antwoorden van de andere participanten te zien nadat de antwoorden zijn verwerkt in een analyse door de facilitator.

Het GDSS platform wordt opgezet met drie verschillende redenen, dewelke hieronder kort worden beschreven. (Een exacte beschrijving van deze redenen zijn terug te vinden in hoofdstuk 6.3 waar dat de bedrijfsdoelen worden getoond)

## **2.1. Gestructureerde besluitvormingsomgeving voor het KnoSoS-systeem**

In de eerste plaats wordt het platform ontwikkeld om besluitvormingsprocessen te ondersteunen voor de personen die met het KnoSoS-systeem [Bijlage 1] werken. Deze personen moeten de mogelijkheid hebben om op een gestructureerde manier besluiten te nemen, en de resultaten van deze besluiten te integreren in de KnoSoS omgeving. Hiervoor zal het GDSS platform gekoppeld moeten worden aan het KnoSoS-systeem.

Het platform moet echter ook als standalone applicatie kunnen fungeren, zonder dat gebruik wordt gemaakt van de KnoSoS omgeving.

## **2.2. Onderzoek naar besluitvormingsprocessen in functie van kennisdeling**

Aangezien de twee betrokken onderzoekscentra werken rond kennismanagement, willen ze het nut van groepsbesluitvormingsondersteunende systemen onderzoeken bij het structureren van kennisdeling. Hiervoor willen ze een platform (laten) ontwikkelen waarin ze voldoende vrijheid hebben om de ondersteunende processen zelf, naar behoeven, samen te stellen.

Het platform zal dus gebruikt worden om verder onderzoek te begeleiden naar groepsbesluitvorming en kennisdeling.

## **2.3. Onderzoek naar synchrone besluitvormingsprocessen**

Het GDSS platform zal, in tegenstelling tot de vorig ontwikkelde systemen, toelaten om synchrone besluitvormingsprocessen op te zetten. Bij zo een besluitvormingsproces kunnen gegevens worden verstuurd tussen de eindgebruikers (tussen participanten en de facilitator en tussen participanten onderling) en tussen de eindgebruikers en het platform, zonder dat de webpagina van de eindgebruiker moet worden herladen. Het voordeel van deze synchrone manier van werken is dat de workflow van de eindgebruiker niet onderbroken wordt wanneer deze informatie wenst te versturen naar andere gebruikers of naar het platform.

Deze manier van werken brengt echter ook nieuwe complexe problemen met zich mee:

- Hoe wordt deze nieuwe informatie duidelijk zichtbaar gemaakt voor de eindgebruikers,
- Hoe gaat de participant om met deze nieuwe informatie / informatie-overload,
- Wat is de performantie van deze communicatie

Deze nieuwe problematiek zal, met behulp van het platform, in een later stadium verder onderzocht kunnen worden door de betrokken onderzoekscentra.

### 3. Kadering van onderzoeksonderdelen

Het onderzoek dat wordt uitgevoerd is opgedeeld in een aantal deelonderzoeken. Dit hoofdstuk heeft als doel om deze verschillende deelonderzoeken te kaderen en om aan te geven waarom de deelonderzoeken in deze specifieke volgorde zijn uitgevoerd.

Om het onderzoek op te starten is het noodzakelijk om domeinkennis op te doen omtrent groepsbesluitvormingsprocessen. Met behulp van deze domeinkennis kan een generiek besluitvormingsproces worden gemaakt, waarin de gemeenschappelijke kenmerken van besluitvormingsprocessen worden gemodeleerd (Zie hoofdstuk 4: “Opstellen van een generiek groepsbesluitvormingsproces”). Dit generiek besluitvormingsmodel ligt in een later stadium mee aan de basis voor het ontwikkelen van de software architectuur. De architectuur moet deze generieke kenmerken goed kunnen ondersteunen en moet ook ruimte bieden voor enkele processpecifieke kenmerken die niet in het generieke besluitvormingsmodel zijn opgenomen.

Om naast deze theoretische inzichten over besluitvormingsprocessen ook praktijkervaringen mee te nemen bij het opzetten van het platform, wordt een analyse uitgevoerd op een bestaand besluitvormingssysteem dat het Delphi besluitvormingsproces implementeert (Zie hoofdstuk 5: “Analyseren van bestaand Delphi systeem”). Aan de hand van deze analyse wordt het ook eenvoudiger om de exacte eisen van het platform vast te stellen, aangezien voor een aantal onderdelen een vergelijking kan worden gemaakt met het bestaande Delphi systeem.

Met deze theoretische en praktische kennis is het mogelijk om de eisen en de wensen ten aanzien van het platform te achterhalen (Zie hoofdstuk 6: “Achterhalen van de eisen voor het platform”). Het is belangrijk om deze eisen correct op te stellen, aangezien deze een belangrijke input zijn bij het opzetten van de software architectuur. Het is immers onmogelijk een architectuur te bouwen zonder te weten wat de verwachtingen ten aanzien van die architectuur zijn.

In een laatste stap voor het opzetten van de architectuur wordt gekeken naar welke meerwaarde AJAX kan hebben in het platform (Zie hoofdstuk 7: “Onderzoeken van het potentieel van AJAX”). Omdat AJAX nog een nieuwe technologie is waar nog heel wat onduidelijkheid rond bestaat, wordt eerst een afbakening gemaakt van wat AJAX inhoudt en wanneer de omstandigheden optimaal zijn om van AJAX gebruik te maken. Aan de hand van deze bevindingen wordt gekeken waar in het platform AJAX kan worden ingezet.

Met de vastgestelde eisen, de inzichten over AJAX en het uitgewerkte generieke besluitvormingsproces, is het mogelijk om een software architectuur op te stellen (Zie hoofdstuk 8: “Opstellen van de software architectuur”). Om deze software architectuur te valideren en om aan te tonen dat de opgestelde eisen kunnen worden uitgewerkt in de architectuur, wordt een prototype ontwikkeld waarin de belangrijkste onderdelen worden uitgewerkt (Zie hoofdstuk 9: “Uitwerken van het prototype”).

## 4. Opstellen van een generiek groepsbesluitvormingsproces

### 4.1. Onderzoeksaanpak

Op de eerste plaats wordt een literatuuronderzoek uitgevoerd om inzicht te krijgen in de typische kenmerken van GDSS systemen. Dit moet gezien worden in de breedst mogelijke context:

- wat zijn typische kenmerken van een GDSS systeem,
- wat zijn de terugkomende problemen bij het ontwerpen van een GDSS systeem,

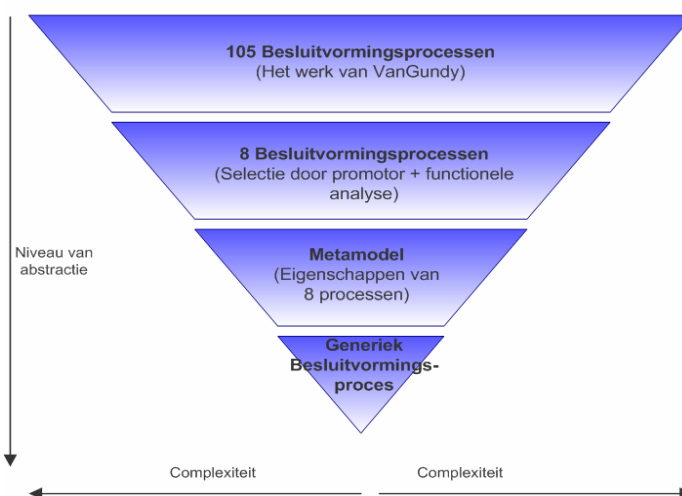
Om deze systemen te kunnen begrijpen en doorgronden is ook domeinkennis nodig, in het bijzonder kennis over besluitvormingsprocessen in groepsverband (Onderzoeksvraag 1).

Een belangrijke informatiebron hiervoor is het werk van [VanGundy], waarin 105 besluitvormingsprocessen worden beschreven. Dit boek wordt door [CreativityAtWork] omschreven als “[...] *This book is considered by many to be the “bible” of problem solving techniques. It was the first comprehensive book on techniques and still is used as a resource book by practitioners in marketing research, new product development, Research & Development, training and many other fields. It also has served as the primary reference resource for many idea generation books currently on the market. [...]*”

Omdat een verzameling 105 besluitvormingstechnieken te complex is om te kunnen overzien, worden enkele abstractieniveaus toegepast op de verzameling. Een eerste abstractie filtert de belangrijkste processen uit de verzameling. Op basis van deze processen wordt een metamodel ontwikkeld dat de kenmerken van besluitvormingsprocessen kan herbergen. Een laatste stap is om van deze kenmerken de gemeenschappelijke eigenschappen te modelleren in een generiek besluitvormingsproces (Onderzoeksvraag 2). Dit generiek besluitvormingsproces zal in een later stadium (Zie Hoofdstuk 8: “Opstellen van de software architectuur”) gebruikt worden om de software architectuur mee op te zetten.

Een schematisch overzicht van de verschillende abstractieniveaus staat in Figuur 1.

De domeinspecifieke kennis die noodzakelijk is voor het opzetten van dit generiek besluitvormingsproces wordt nadien ook gebruikt om de eisen van het platform te kunnen plaatsen in een juiste context. (Onderzoeksvraag 3a)



**Figuur 1: Abstraheren van 105 besluitvormingsprocessen naar een generiek besluitvormingsproces**

## 4.2. Uitvoering

### 4.2.1. Selecteren van besluitvormingsprocessen

Het werk [VanGundy] geeft van 105 besluitvormingsprocessen een procesmatige beschrijving, met de voor- en nadelen (en wetenschappelijk gerelateerde studies). [ParticipatoryToolkit] geeft aan wanneer een bepaalde techniek het best gebruikt kan worden en welke voorbereidingen genomen moeten worden voor de aanvang van het proces. Verder geven ze veel praktische tips voor het gebruik van de besluitvormingsprocessen.

Omdat het analyseren van 105 besluitvormingsprocessen teveel tijd in neemt, en om de complexiteit van de analyse te reduceren, zijn een 13-tal processen geselecteerd door de interne promotor. Deze geselecteerde besluitvormingsprocessen dekken de meest uiteenlopende vormen van besluitvormingsprocessen af en vormen bijgevolg een representatie van wat besluitvormingsprocessen inhouden.

Om te kijken welke van deze 13 processen, op functioneel niveau, overlappingen bevatten, wordt een korte analyse uitgevoerd. De processen onderscheiden zich door de verschillende soort vragen die gebruikt worden (open vragen, meerkeuze vragen,...) en naar wat voor soort interactie tussen de participanten onderling plaats vindt. (geen onderlinge communicatie, veel onderlinge communicatie,...). Indien 2 of meer processen op deze gebieden teveel overeenkomsten vertonen, wordt hiervan het belangrijkste proces behouden. Na deze analyse bleven de volgende 8 processen over, waarop het generiek besluitvormingsmodel zich baseert:

1. Brainwriting Pool
2. Classical Brainstorming
3. Method 6-3-5
4. Story Boards
5. Creative Problem Solving
6. Kepner-Tregoe
7. Nominal Group Technique
8. Delphi

Deze technieken worden ook als de meest belangrijk beschouwd binnen de onderzoeksgroep. Deze processen worden in een later stadium ook als eerste geïmplementeerd in het GDSS platform, waardoor het van extra groot belang is dat deze processen goed worden ondersteund in de architectuur.

### 4.2.2. Opstellen van een metamodel

Uit al deze gemeenschappelijk eigenschappen wordt een metamodel opgesteld. Dit metamodel laat toe om de eigenschappen van besluitvormingsprocessen structureel te noteren.

Het metamodel bevat de volgende eigenschappen:

- Aantal personen dat deel neemt aan het proces (minimum / maximum)
- Geschatte duur voor het doorlopen van het proces (uitgedrukt in minuten)
- Is het proces anoniem/ kan anonimiteit worden afgedwongen
- Te doorlopen cyclus:
  - Stappenplan bij het doorlopen van het proces
  - Wie onderneemt welke acties
  - Welke informatie stroomt naar welke persoon
- Specifieke kenmerken van het besluitvormingsproces
  - Wat zijn de voordelen
  - Wat zijn de nadelen

### **4.2.3. Factoren die een invloed uitoefenen op de kwaliteit van besluiten**

Met het metamodel (Zie 4.2.2) kan het generiek besluitvormingsmodel worden ontwikkeld. Dit generiek besluitvormingsmodel moet echter zo gebouwd worden dat er kwalitatief goede besluiten mee kunnen worden genomen. Daarom wordt in dit deelonderzoek onderzocht welke factoren de kwaliteit van besluiten positief of negatief kunnen beïnvloeden, zodat een systeem kan worden ontwikkeld dat de kwaliteit van de besluitvormingsprocessen kan bevorderen.

De twee belangrijkste factoren die een invloed kunnen uitoefenen zijn in principe twee basiscomponenten van een besluitvormingsproces: de participanten moeten onderling communiceren (factor 1) om zo hun kennis te kunnen delen (factor 2). Een derde factor is de wisselwerking van de tevredenheid van de participanten over een bepaald besluit. Indien de participanten een tevreden gevoel hebben na het vormen van een besluit, zullen ze volgende keer vol vertrouwen aan een nieuw besluit werken. Indien de participanten echter ontevreden zijn over de genomen besluiten, bestaat de kans dat in latere besluiten de participanten ongemotiveerd gaan zijn om een kwalitatief besluit neer te zetten.

Andere factoren, die ook een rol spelen bij de kwaliteit van besluiten (zoals bijvoorbeeld de beschikbare expertise bij de participanten, de typesnelheid van de participanten), worden niet onderzocht, aangezien het platform hier geen (directe) invloed op kan uitoefenen.

Omdat de meeste besluiten geen objectief “beste” besluit hebben, is het erg lastig om de kwaliteit van een besluit vast te kunnen stellen. Het is namelijk bijna nooit zeker wat het juiste besluit is, en dus of de beslissing kwalitatief goed is.

Een uitgebreide analyse van wat in de literatuur werd gevonden over kwaliteit in besluitvormingsprocessen is te vinden in [Bijlage 2]. Deze analyse vormt tevens de basis voor de resultaten die in hoofdstuk 4.3 worden besproken.

### **4.2.4. Schaalbaarheid van het generieke besluitvormingsproces**

Om te onderzoeken wat voor schaalbaarheid het generieke besluitvormingsproces moet bezitten, wordt expliciet aandacht besteed aan de grootte van een groep bij besluitvormingen. Om een impressie te krijgen van de gemiddelde grootte van een groep, worden de processen uit [VanGundy] bestudeerd. Omdat groeps grootte in een digitale omgeving, bij een gepaste ondersteuning, minder kritisch is, is besloten om ideeën van systemen van exorbitante grootte [SDSS, TheWisdomOfCrowds] mee te nemen in de analyse. Hierdoor kunnen de kenmerken van de extreem grote besluitvormingsprocessen ook meegenomen worden in het generieke besluitvormingsproces.

Een uitgebreide analyse van wat in de literatuur werd gevonden over de gemiddelde grootte en over extreem grote groepen in besluitvormingsprocessen is te vinden in [Bijlage 2]. Deze analyse vormt tevens de basis voor de resultaten die in hoofdstuk 4.3 worden besproken.

## **4.3. Resultaten**

Dit hoofdstuk bevat de resultaten die zijn verkregen tijdens het vormen van het generiek besluitvormingsproces. De eisen aan het platform die resulteerde uit dit deelonderzoek zijn opgenomen in hoofdstuk 4.4 “Vastgestelde eisen”.

### **4.3.1. Factoren die een invloed uitoefenen op de kwaliteit van besluitvormingsprocessen**

Voor elke factor dat een invloed kan hebben op de kwaliteit van een besluitvormingsproces wordt hier een kort overzicht gegeven.

### **Factor 1: Communicatie tussen participanten**

De voordelen van GDSS communicatie ten opzichte van face-to-face communicatie zijn:

- De participant kan anoniem zijn mening formuleren
- De participanten moeten niet op dezelfde fysieke plek aanwezig zijn.
- De participanten kunnen de aangeleverde informatie verwerken op het moment dat ze het zelf wensen. Ze verwerken de informatie door middel van een “pull”-mechanisme, in plaats van het “push”-mechanisme bij verbale communicatie

De voordelen van een (traditionele) face-to-face communicatie zijn:

- De communicatie bevat veel non-verbale gegevens (intonatie, lichaamstaal en ondersteunende gebaren)
- De participant vertrouwt de aangeleverde informatie mogelijk beter als bij GDSS communicatie, aangezien de informatie niet anoniem wordt aangeboden.

### **Factor 2: Kennisuitwisseling tussen participanten**

Kennisuitwisseling tussen participanten is niet iets vanzelfsprekends. Participanten delen namelijk (bewust of onbewust) niet al hun kennis en participanten nemen niet alle beschikbare informatie op [StimulatingThinking]. Hierdoor gaat een groot deel van de geleverde kennis verloren.

De factoren die hierbij een rol spelen zijn vooral :

- Onzichtbaarheid van de nieuwe aangeleverde informatie (onbewust niet alle informatie verwerken)
- Motivatieproblemen om de nieuwe informatie te verwerken (onbewust en/of bewust niet alle informatie verwerken)
- Niet alle kennis beschikbaar stellen aan andere participanten om de besluitvorming te manipuleren (bewust niet alle informatie delen)
- Overvloedig aanleveren van informatie, waardoor de participanten door het bos de bomen niet meer ziet (onbewust teveel informatie delen)

Mogelijke oplossingen voor deze problemen zijn:

- Duidelijk signaleren waar en wanneer nieuwe informatie wordt aangeboden aan de participanten
- De nieuwe informatie aanbieden in een aantal vastgelegde categorieën
- de participant de mogelijkheid geven om de nieuwe informatie zelf te laten rangschikken of categoriseren

### **Factor 3: Tevredenheid van participanten**

[FeelGood] geeft aan dat de tevredenheid van de besluiten afhankelijk is van de grootte van de groep. Dit is toe te schrijven aan het feit dat kleine groepen te lijden hebben onder drie negatieve neveneffecten van een GDSS systeem, met name:

- Het intypen van antwoorden is trager als verbale face-to-face communicatie
- Het overbrengen van een boodschap via tekst heeft slechts een beperkte expressiviteit: er kunnen geen/slecht emoties en nuances in worden opgenomen.
- De participant weet niet altijd met wie hij/zij in een besluitvormingsproces zit, waardoor de participant zich “afstandelijk” kan voelen.

Grote groepen kunnen echter meer profijt halen uit de voordelen van een GDSS systeem:

- Participanten kunnen zich anoniem opstellen om zo hun ware identiteit verborgen te houden.

- Parallel verlopen van het proces: er kunnen meerdere participanten tegelijk deelnemen aan éénzelfde discussie en dus virtueel tegelijk spreken (in plaats van het seriële communiceren in face-to-face situaties).

Om de negatieve effecten van een GDSS systeem te beperken kunnen de volgende maatregelen worden genomen in het GDSS platform:

- Het toevoegen van “richt text controls”, waardoor de participanten door middel van tekstopmaak beter nuances kunnen aanbrengen.
- De participanten in staat stellen om de profielen van andere participanten op te vragen, zodat een grotere groepscohesie ontstaat (en de participanten zich niet meer “afstandelijk” voelen).

#### **4.3.2. Schaalbaarheid van het generieke besluitvormingsproces**

- Een gemiddelde van het aantal personen dat aan een besluitvormingsproces deel neemt is ongeveer 5 à 10 personen
- Sommige beperkingen van het aantal personen zijn echter eigen aan face-to-face besluitvormingsprocessen, waardoor deze beperkingen bij een GDSS systeem niet meer gelden.
- Bij extreem grote groepen (enkele honderden tot duizenden) moet het proces een zelfregulerend systeem hebben, waarbij geen gebruik wordt gemaakt van facilitators die het proces beheren.
- Een grote groep waarin een hoge mate van diversiviteit is, kan onder bepaalde voorwaarden gelijkwaardige (of zelfs betere) besluiten nemen dan een groep van experts.

Uit de analyse van besluitvormingsprocessen die (extreem) grote groepen ondersteunen, blijkt dat deze processen op een aantal fundamentele eigenschappen verschillen vertonen. Zo zijn er bij deze besluitvormingsprocessen geen facilitators aanwezig, en is er dus ook geen eindverantwoordelijke voor het proces. Deze processen baseren zich op zelfregulerende systemen om zo het besluitvormingsproces in goede banen te leiden.

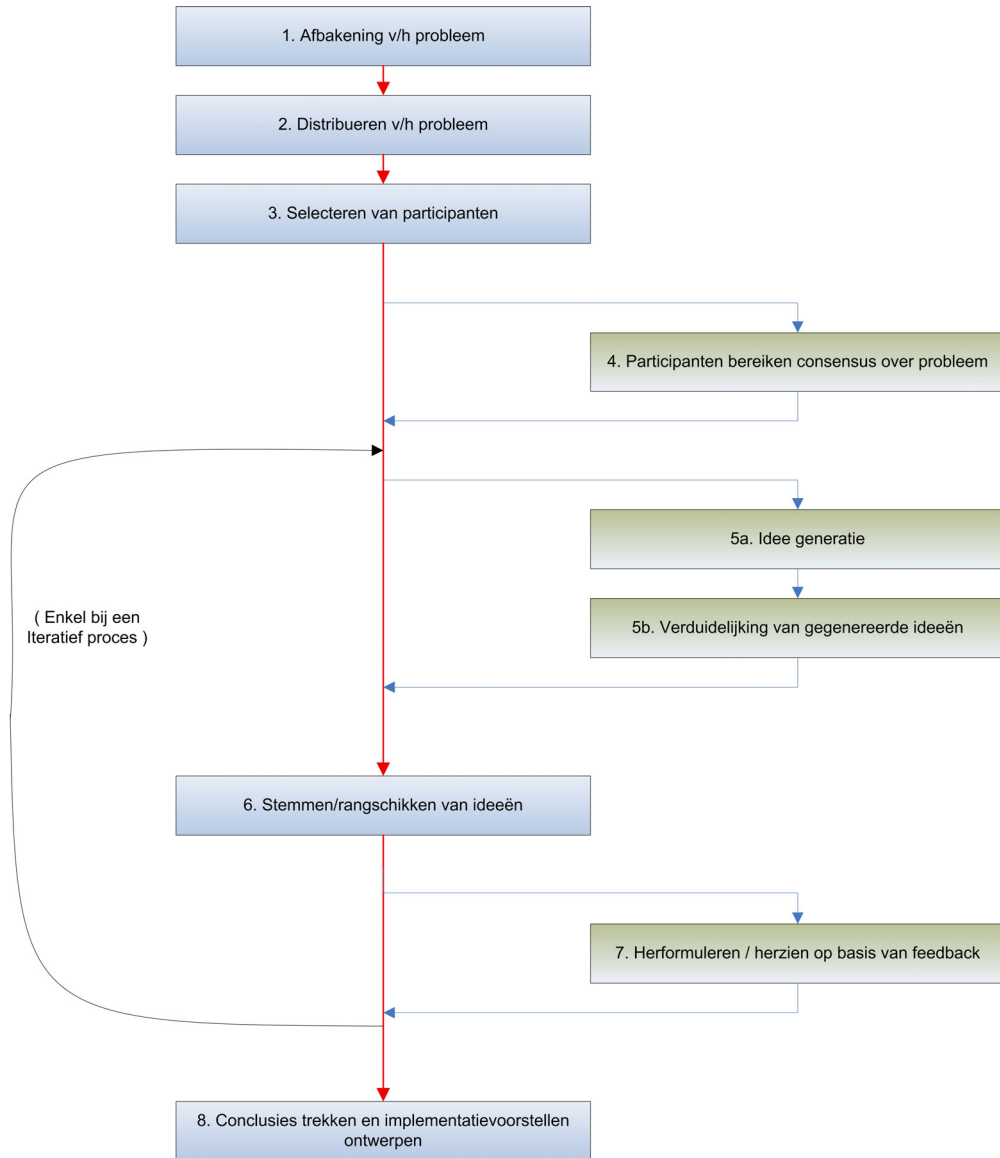
Omdat deze manier van werken dus radicaal anders is als die van processen met een normale groeps grootte, is het praktisch onmogelijk om een generiek besluitvormingsproces op te zetten dat beide manieren van werken ondersteunt. Aangezien de onderzoekscentra zich voornamelijk wensen te richten op de normale besluitvormingsprocessen is beslist om de schaalbaarheid van het generiek besluitvormingsproces niet radicaal te veranderen. Het generiek besluitvormingsproces zal in 4.3.4 “Opstellen van het generieke besluitvormingsproces” bijgevolg geen rekening houden met besluitvormingsprocessen die extreem grote groepen ondersteunen.

#### **4.3.3. Opstellen van het generieke besluitvormingsproces**

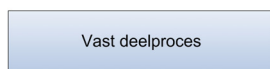
De geanalyseerde groepsbesluitvormingsprocessen uit [VanGundy] vertonen een aantal grote gemeenschappelijke delers: alle besluiten worden genomen door gestructureerd mensen aan het woord te laten op vastgelegde tijdstippen, of juist omgekeerd, door iedereen vrij te laten spreken. Tevens is er steeds een facilitator aanwezig, iemand die de discussie en het algemene besluitvormingsproces in goede banen leidt en beheert (uitgezonderd bij de extreem grote besluitvormingsprocessen, waarbij een zelfregulerend systeem wordt voorgesteld. Zie 4.3.3 “Schaalbaarheid van het generieke besluitvormingsproces”).

Een ander typerend kenmerk dat voor de meeste besluitvormingsprocessen terug komt, en wat ook in [VanGundy] wordt beschreven, is dat de bestudeerde processen ook gelijkaardige “fasen” doorlopen. In de eerste fase worden zoveel mogelijk ideeën gegenereerd (divergeren), bijvoorbeeld door middel van brainstorming of hitchhiking. In een tweede fase wordt naar een consensus toe gewerkt (voting en eliminatie van ideeën). Deze fasen worden in een aantal besluitvormingsprocessen iteratief doorlopen.

Figuur 2 geeft het generiek besluitvormingsproces weer. Aangezien niet alle deelprocessen noodzakelijk zijn om tot een besluitvorming te komen, zijn sommige deelprocessen optioneel. Deze optionele deelprocessen worden weergegeven doordat ze als aftakking van de chronologische lijn van het generiek besluitvormingsproces optreden. De deelprocessen worden ook tekstueel beschreven in [Bijlage 2] (hoofdstuk 3.3 “Opstellen van generiek besluitvormingsproces”.)



### Legenda



Chronologisch verloop van besluitvormingsproces



Aftakking naar een optioneel deelproces



Sprong in de chronologische volgorde

**Figuur 2: Het generieke besluitvormingsproces**



#### **4.4. Vastgestelde eisen**

Tijdens het opstellen van dit generieke besluitvormingsproces zijn enkele aandachtspunten naar boven gekomen waarmee het platform rekening moet houden. Deze aandachtspunten geven aanleiding tot het opstellen van een nieuwe eis aan het platform. [Bijlage 5] laat zien welke aandachtspunten hebben geleid tot welke eisen.

Om kennisuitwisseling tussen participanten zo optimaal mogelijk te laten verlopen moet het platform:

- Duidelijk te signaleren waar en wanneer nieuwe informatie wordt aangeboden aan de participanten (Aandachtspunt 1),
- Deze nieuwe informatie aan te bieden in een aantal vastgelegde categoriën (Aandachtspunt 2),
- De participant de mogelijkheid te geven om de nieuwe informatie zelf te rangschikken of te categoriseren. (Aandachtspunt 3).

Om de negatieve effecten van communicatie via een GDSS systeem te beperken moeten de volgende maatregelen worden genomen:

- Het toevoegen van “richt text controls”, waardoor de participanten door middel van tekstopmaak beter nuances kunnen aanbrengen in hun antwoorden. (Aandachtspunt 4)
- De participanten in staat stellen om de profielen van andere participanten op te vragen. (Aandachtspunt 5)

## 5. Analyseren van bestaand Delphi systeem

### 5.1. Onderzoeksaanpak

Omdat het platform enkele concepten en ideeën kan overnemen van een bestaand besluitvormingssysteem, worden op dit systeem twee analyses uitgevoerd. Deze analyses moeten een duidelijk beeld schetsen van de sterke en zwakke punten van het bestaande systeem.

Hierbij wordt gekeken naar wat het systeem kan (analyse van de functionaliteit) en wat de gebruikersproblemen zijn bij dit systeem (analyse van gebruikersproblemen). Een puur technische analyse wordt niet uitgevoerd, aangezien het GDSS platform totaal andere technieken zal gebruiken dan het bestaande Delphi systeem, waardoor een technische analyse geen meerwaarde zou bieden.

Het doel van deze analyses is om gebruik te maken van de ervaringen van het vorige systeem, zodat de beste ideeën uit het systeem overgenomen kunnen worden en zodat dezelfde fouten vermeden kunnen worden tijdens het opstellen van de software architectuur en (Onderzoeksvraag 3c) de prototype-implementatie hiervan. (Onderzoeksvraag 3d)

### 5.2. Uitvoering

#### 5.2.1. Analyse gebruikersproblemen

In 2004 werd op de VUB een Delphi besluitvormingsproces uitgevoerd met behulp van de Delphi applicatie. Dit besluitvormingsproces bevatte twee open bevraging rondes en werd uitgevoerd met in totaal 159 participanten.

Gedurende het traject van dit besluitvormingsproces werden verscheidene e-mails verstuurd naar de beheerder van het besluitvormingsproces om vragen te stellen in verband met de Delphi applicatie. Deze mails worden hier geanalyseerd en opgedeeld in een aantal categoriën.

De volgende categoriën worden opgesteld om de mails te categoriseren:

- Geen vertrouwen en / of geen inzicht in het werken van de applicatie
- Tijdsgebrek voor deelname aan het besluitvormingsproces
- Gebruikersnaam en / of wachtwoord vergeten
- Website van het besluitvormingsproces is onbereikbaar / slecht bereikbaar

Omdat het eerste besluitvormingsproces uit een licht gewijzigde versie van het Delphi proces bestond (er werd enkel gewerkt met open vragen), is (in overleg met de interne promotor) besloten om een tweede bijkomend besluitvormingsproces te analyseren.

Na het analyseren van de support-emails is, samen met de facilitator, de beheerssectie onderzocht. Hierbij gaf de facilitator een "task demonstration" [SoftwareRequirements]. Alle functionaliteit met betrekking tot het beheren van het besluitvormingsproces is onderzocht op de volgende punten:

- Wordt de functie doorgaans gebruikt tijdens een analyse van de gegeven antwoorden.
- Is de functie handig of omslachtig in gebruik.
- Wat zijn de verbeterpunten van deze functie.

## 5.2.2. Analyse functionaliteit systeem

Om de functionaliteit van het systeem te testen zijn gegevens ingeladen van een afgerond besluitvormingsproces. Deze gegevens bevatten al de antwoorden (argumenten en ideeën) die zijn gegeven door de participanten tijdens één bepaalde bevragingronde. Hierna zijn scenario's opgesteld die situaties beschrijven die kunnen voorkomen in het systeem. Samen met de facilitator zijn deze scenario's doorlopen, om zo de sterke en zwakke kenmerken van het systeem te achterhalen. De volgende scenario's zijn doorlopen:

- Het beheren van gebruikers (toevoegen, aanpassen, verwijderen).
- Het beheren van vragen die voorgelegd worden aan de participanten.
- Het openen/sluiten van een bevragingronde.
- Het invullen van de vragen door een participant.
- Het opslaan van de vragen door een participant.
- Het analyseren van de antwoorden in de beheerssectie.

## 5.3. Resultaten

Deze sectie vat de resultaten samen van de problemen die zijn gevonden bij het analyseren van het bestaande Delphi systeem. De volledige analyse staat uitgewerkt in [Bijlage 3]. De problemen die in het platform moeten worden voorkomen zijn als aandachtspunten uitgewerkt in hoofdstuk 5.4 "Vastgestelde eisen".

### 5.3.1. Analyse gebruikersproblemen

#### Overzicht resultaten van support emails

Bij het analyseren van de support emails is een onderscheid gemaakt tussen verschillende soorten van problemen. Om een overzicht te krijgen van de verschillende soorten van problemen, zijn een aantal categoriën opgesteld (zie 5.2.1).

De resultaten van deze analyse kunnen als volgt worden samengevat:

- Er bestaat veel onduidelijkheid bij de participanten of de antwoorden zijn opgeslagen in het systeem of niet.
- Een groot aantal participanten geeft aan geen tijd te hebben voor een deelname. Het is echter onduidelijk of de eindgebruikers effectief teweinig tijd hebben of dat een andere oorzaak gezocht moet worden. Aangezien veel participanten afhaken tussen de rondes door kan het namelijk zijn dat participanten zich niet meer kunnen motiveren om nog een keer te werken met het systeem.
- De beschikbaarheid en performantie van het systeem was te laag.

#### Overzicht gebruikersproblemen beheerssectie

De problemen die terug te vinden zijn in de beheerssectie van het Delphi systeem zijn talrijk. Alhoewel dat de expert die het systeem al enkele jaren gebruikt weinig klachten heeft, kunnen toch enkele duidelijke tekortkomingen worden vastgesteld. Het is duidelijk dat de expert na al die jaren werken zich bij de tekortkomingen heeft neergelegd en als het ware "blind" is geworden voor deze constructiefouten (zie ook [AlanCooperTheInmates]).

De belangrijkste fouten die zijn gevonden:

- De beheerssectie vergt teveel manuele handelingen en is te arbeidsintensief. Zaken die perfect geautomatiseerd kunnen worden, moeten toch handmatig worden uitgevoerd, wat tot onnodig tijdverlies en ergernis kan leiden.
- Bij de analyse van de antwoorden ontstaat een enorme informatieoverload. De gegevens kunnen niet gefilterd worden wat leidt tot ellenlange pagina's met tekst.
- De facilitator is niet in staat om te achterhalen welke participant volledig klaar is met het ingeven van antwoorden.

### 5.3.2. Analyse functionaliteit

Om het GDSS platform verder te laten bouwen op de beste praktijken van het bestaand Delphi systeem, werd gekeken welke sterke eigenschappen het systeem heeft. Ook de negatieve elementen werden bekeken, zodat deze functionaliteit verbeterd en/of uitgebreid kan worden in het te ontwikkelen GDSS platform.

#### Beheerssectie

Deze beheerskant van het systeem heeft twee onderdelen: het configureren van de “Delphi webserver” (standalone VisualWorks applicatie) en het beheren van de data (webapplicatie).

De belangrijkste (of opmerkelijkste) eigenschappen / hidden features van het webgedeelte zijn:

- De beheerssectie stelt de facilitator in staat om het Delphi proces te beheren door:
  - Gebruikers aan te maken en rechten te geven op een ronde
  - Analyses uit te voeren op de gegeven antwoorden en zo een nieuwe ronde op te starten
- De beheerssectie kent geen inlogprocedure. Kennis van de URL geeft toegang tot de beheerssectie.
- Er is geen mogelijkheid om een andere structuur te hanteren dan de voorgestelde Delphi methode. Wel is het mogelijk om meerdere rondes op het zelfde moment te hebben open staan (dit is een hidden feature, want het staat nergens vermeld).
- Gebruikersnamen en paswoorden worden opgezet volgens een vast patroon.

De beheerssectie die zich bezig houdt met het opzetten van de “Delphi webserver” bevat de volgende eigenschappen:

- Er is fysieke toegang nodig tot de webserver om de Delphi webserver te configureren, of er dient een remote desktop applicatie te worden geïnstalleerd
- De webserver ondersteund slechts 1 instantie van het Delphi proces. Indien meerdere processen simultaan moeten worden uitgevoerd dient een nieuwe webserver te worden opgestart.

#### Gebruikersgedeelte

Het gebruikersgedeelte, het gedeelte dat de participanten gebruiken om te antwoorden, bevat de volgende eigenschappen:

- Alle vragen worden bovenaan het scherm weergegeven. Per vraag wordt een link getoond die verwijst naar een subgedeelte in de pagina waar de vraag herhaald wordt en waar de vraag daadwerkelijk beantwoord kan worden. Hierdoor is de pagina onnodig lang.
- Er bevindt zich slechts 1 “opslaan” knop op de hele pagina (onderaan de pagina), waardoor de participant steeds naar beneden moet scrollen om zijn/haar antwoord(en) te bewaren. Wanneer de pagina opnieuw wordt opgebouwd moet de participant handmatig zoeken wat zijn laatst ingevuld antwoord was.
- De participant heeft geen overzicht op het aantal vragen dat hij/zij reeds heeft beantwoord.

## 5.4. Vastgestelde eisen

Doorheen de analyse van het Delphi systeem blijkt dat een tal van gebruikersproblemen optreden tijdens het uitvoeren van een besluitvormingsproces (zowel bij de facilitator als bij de participanten). Aan deze problemen is echter nooit veel aandacht geschonken en de gebruikers leerden leven met de gebreken van de applicatie zonder zich echt vragen te stellen bij deze problemen.

Een aantal van deze problemen zijn dermate belangrijk dat ze als aandachtspunten worden opgenomen. Deze aandachtspunten geven aanleiding tot concrete eisen die gesteld worden aan het platform. Een overzicht van welke aandachtspunten tot welke eisen hebben geleid, kan worden gevonden in [Bijlage 5]. De belangrijkste aandachtspunten waarin het platform een verbetering moet zijn voor de participant:

- Bij de participanten bestaat veel onduidelijkheid of de antwoorden zijn opgeslagen in het systeem of niet. (Aandachtspunt 6)
- Er bevindt zich slechts 1 “opslaan” knop op de hele pagina (Aandachtspunt 7)
- De participant heeft geen overzicht op het aantal vragen dat hij/zij reeds heeft beantwoord. (Aandachtspunt 8)

De belangrijkste problemen die opgelost moeten worden voor de facilitator zijn:

- De beheerssectie vergt teveel manuele handelingen en is te arbeidsintensief. (Geen direct aandachtspunt, aangezien dit een algemene opmerking is, en niet specifiek kan worden opgenomen als eis)
- Bij de analyse van de antwoorden ontstaat een enorme informatieoverload. (Aandachtspunt 9)
- De facilitator is niet in staat om te achterhalen welke participant volledig klaar is met antwoorden. (Aandachtspunt 10)
- De beheerssectie kent geen inlogprocedure. (Aandachtspunt 11)
- Gebruikersnamen en paswoorden worden opgezet volgens een vast patroon. (Aandachtspunt 12)
- Er is fysieke toegang nodig tot de webserver om de Delphi webserver te configureren, of er dient een remote desktop applicatie te worden geïnstalleerd (Ook hier is geen direct aandachtspunt van te maken, aangezien het een technische restrictie is dat het platform wordt uitgewerkt via een webbased systeem)
- De webserver ondersteund slechts 1 instantie van het Delphi proces. (Aandachtspunt 13)

## 6. Achterhalen van de eisen voor het platform

### 6.1. Onderzoeksaanpak

Om de software architectuur van het GDSS platform te kunnen opstellen moest eerst worden achterhaald wat de exacte eisen en wensen aan het platform zijn. (Onderzoeksvraag 3a). Deze eisen kunnen dan later in een prototype worden geïmplementeerd om zo te valideren dat deze correct zijn vastgelegd (Onderzoeksvraag 3d).

Omdat de eisen en wensen uiteenlopend zijn (wat moet het systeem kunnen, hoe goed moet het systeem het kunnen onder bepaalde voorwaarden,...) is het belangrijk om een goed overzicht te behouden van alle eisen. Hiervoor is gebruik gemaakt van de categorisatie die voorgesteld wordt in [SoftwareRequirements], met name:

- Functionele eisen (welke taken moet het platform kunnen vervullen)
- Niet-functionele eisen (hoe snel, veilig, uitbreidbaar moet het platform deze taken kunnen vervullen)
- Data eisen (welke gegevens moeten er worden bijgehouden in het platform)
- Integratie eisen (welke koppelingen met externe applicaties moeten het platform ondersteunen)

Al deze eisen dienen om bepaalde diepliggende doelen te verwezenlijken. Deze doelen moesten worden vastgelegd als bedrijfsdoelen. Aan de hand van deze bedrijfsdoelen konden vervolgens koppelingen worden gemaakt met de opgestelde eisen, zodat duidelijk werd welke eisen welke doelen vervulden en welke eisen mogelijk overbodig/niet-prioritair waren.

Door deze eisen goed af te bakenen en deze bedrijfsdoelen goed voor ogen te houden kon een scope worden gemaakt van wat wel en wat niet gerealiseerd moest worden in het GDSS platform.

### 6.2. Uitvoering

Voor het achterhalen van de eisen voor het platform werd voornamelijk gebruik gemaakt van documentaties over het bestaande Delphi systeem, documenten over een eerder geschetst generieke besluitvormingsproces en interviews met de interne promotor. Aanvullend op deze bestaande documentatie werd gebruik gemaakt van de verworven inzichten van het generieke besluitvormingsproces en de eigenschappen die de kwaliteit van de besluiten kunnen beïnvloeden.

Tijdens het achterhalen en opstellen van de eisen kon ook rekening worden gehouden met de gekende gebreken en sterke kanten van het vorige systeem. De scenario's die gebruikt werden bij die analyse konden ook (gedeeltelijk) herbruikt worden om de eisen vast te stellen. Een deel van de scenario's uit het vorige systeem bleven namelijk (gedeeltelijk) hetzelfde, waardoor het eenvoudiger was om een overzicht te krijgen van welke onderdelen van het platform nog moesten worden geanalyseerd.

De documentatie van het eerder geschetste generieke besluitvormingsproces bestond uit een functionele analyse die werd opgesteld voor een implementie in .NET, die slechts gedeeltelijk werd uitgewerkt. Deze functionele analyse bevatte een aantal kernprincipes van het GDSS platform, maar hield onder meer geen rekening met de ondersteuning van zowel synchrone als asynchrone processen.

Verder bestond de bestudeerde documentatie uit een verzameling van change requests die waren opgemaakt voor de Delphi applicatie, maar die nooit werden uitgevoerd wegens tijd en geld gebrek.

Om al deze documentatie te corrigeren, te actualiseren en uit te breiden naar de huidige behoeften werden verschillende interviews en brainstormsessies [SoftwareRequirements] gehouden met de medewerkers van de betrokken onderzoekscentra. Deze discussies dienden voornamelijk om duidelijkheid te scheppen en om conceptuele problemen uit te praten en te bediscussiëren. Tijdens deze discussies werden ook de prioriteiten vastgelegd van bepaalde features, rekening houdende met de typische kenmerken van de doorsnee gebruikers (Zie Persona's in [Bijlage 4]). Aangezien niet alle eisen als even belangrijk werden ervaren, was het belangrijk om een onderscheid te maken van prioritaire en niet-prioritaire functionaliteit.

De eisen aan het platform spreiden zich over meerdere deelgebieden. Daarom is gekozen om de eisen op te delen in een aantal categoriën, zoals in [SoftwareRequirements] wordt voorgesteld.

### **6.2.1. Functionele Eisen**

Aangezien de meeste van de documentatie specificaties waren van bepaalde features, en dus het meest uitgebreid waren beschreven, werd de meeste aandacht besteed aan het opstellen van de functionele eisen voor het platform. Over de functionele eisen bestond ook de meeste duidelijkheid, omdat hier het meeste over gediscussieerd werd tijdens interview en brainstorm sessies (zoals beschreven in [SoftwareRequirements]). Tijdens deze sessies werd gebruik gemaakt van whiteboards waarop high-level tasks werden genoteerd en (niet-gestructureerde) task descriptions [SoftwareRequirements].

### **6.2.2. Niet functionele eisen**

Tijdens de bestudering van de documenten [SoftwareRequirements] werd duidelijk dat bij het opstellen van al de aanbevelingen, change requests en documentatie niet veel aandacht is besteed aan niet-functionele eisen. Uit interviews bleek het ook erg lastig te zijn om concrete cijfers en criteria op te stellen waarop dat de functionaliteit getest kon worden.

Omdat een groot deel van het platform uit nieuwe features bestond, was het lastig om hierover reeds afspraken te kunnen maken over snelheid, betrouwbaarheid en uitbreidbaarheid, etc. Om toch richtlijnen vast te kunnen stellen werd gedeeltelijk gebruik gemaakt van open metrieken [SoftwareRequirements], en deels van "wat als" scenario's, zoals beschreven in [SoftwareArchitecture]

### **6.2.3. Data eisen**

De datadefinitie van de .NET herimplementatie gaf een goede aanwijzing van welke gegevens opgeslagen moesten worden in het platform. Dit overzicht werd aangevuld met de verworven kennis van het generieke besluitvormingsproces en werd gestructureerd in een data dictionary [SoftwareRequirements].

### **6.2.4. Integratie eisen**

Een koppeling met het KnoSoS-systeem is een belangrijke eis aan het GDSS platform. Omdat het KnoSoS-systeem zelf nog volop in ontwikkeling is, was het lastig om concrete afspraken te maken in verband met data uitwisseling tussen de platformen. Omtrent deze kwestie werd veel gediscussieerd met de ontwikkelaars van het KnoSoS-systeem.

De grote alternatieven die moesten worden afgewogen waren: hoe ver integreren we de systemen en hoe ver moeten ze onafhankelijk van elkaar kunnen werken. Wie neemt de initiatieven om data uit te wisselen tussen de systemen en vooral : welke data wordt uitgewisseld tussen de systemen.

Om afspraken te maken omtrent de data uitwisseling is een technische interface [SoftwareRequirements] opgezet waarin duidelijk staat uitgelegd wie welke data aanlevert.



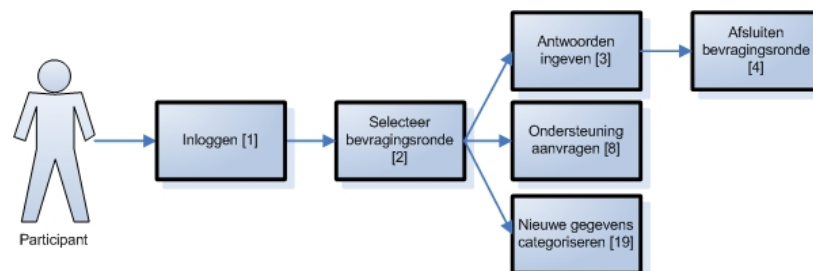


Een overzicht van hoe dat deze eisen tot de bedrijfsdoelen zijn gerelateerd, is terug te vinden in [Bijlage 5]. Figuur 4 is een kort fragment uit dit overzicht ter illustratie van de opgestelde matrix. De opzet van deze matrix is om een overzicht te krijgen van welke eisen tot welke bedrijfsdoelen bijdragen. Zo kunnen eventuele “overbodige” eisen in vraag worden gesteld en moet worden bedacht of deze eisen wel relevant voor het platform.

	Ondersteuning bieden voor onderzoek naar groepsbesluitvorming	Onderzoek naar synchrone intermenselijke communicatie	Onderzoek naar synchrone data transfer	Het groepsbesluitvormingsproces ondersteunen	Ruimte bieden voor uitbreidingen en aanpassingen	Koppeling met KnoSoS platform
<b>Functionele eisen</b>						
1 Inloggen (participant & facilitator)				x		
2 Selecteer een bevringsronde				x		
3 Antwoorden ingeven tijdens een bevringsronde			x	x		
4 Afsluiten van een bevringsronde				x		
5 Autosaving van antwoorden	x		x			
6 Beheren van eindgebruikers				x		
7 Toekennen van rechten aan eindgebruikers				x		
8 Ondersteuning synchrone intermenselijke communicatie	x	x				
9 Beheren van een besluitvormingsproces				x		
10 Beheren van een bevringsronde				x		
11 Importeren en exporteren van een besluitvormingsproces				x		
12 Beheer van groepen						

**Figuur 4: Relatie van eisen aan bedrijfsdoelen**

Om een idee te hebben van hoe dat de opgestelde eisen daadwerkelijk gebruikt dienen te worden door de eindgebruikers van het platform, is een workflow overzicht gemaakt van de eindgebruikers (zie [Bijlage 5]). Hierin staat beschreven welke eindgebruiker welke actie kan ondernemen. Figuur 5 is een voorbeeld van deze workflow, en beschrijft de acties die een participant kan ondernemen tijdens een bevringsronde.



**Figuur 5: Acties die een participant kan uitvoeren tijdens een bevringsronde**

Deze workflow overzichten helpen om een overzicht te krijgen van welke eisen relateren aan de andere eisen. Hierbij kunnen eventuele overbodige eisen nog achterhaald worden (Indien niet één eindgebruiker een eis daadwerkelijk nodig heeft, is de opgestelde eis overbodig). Ook kan een workflow overzicht ontbrekende eisen aan het licht brengen. Indien bij het opstellen blijkt dat een eindgebruiker een handeling moet uitvoeren die niet is beschreven, kan het zijn dat een ontbrekende eis is gevonden. Hierdoor was het overzicht van deze workflow uitermate geschikt om de compleetheid van de eisen te toetsen.

## 7. Onderzoeken van het potentieel van AJAX

### 7.1. Onderzoeksaanpak

Aangezien het hele GDSS platform ontwikkeld zal worden met behulp van webtechnologieën, is gekeken of de nieuwe techniek AJAX (Asynchronous Javascript and XML) [Garrett Ajax] gebruikt kon worden om het platform te voorzien van een (relatief) eenvoudige manier om data uit te wisselen tussen de eindgebruikers onderling en tussen de eindgebruikers en het platform, zonder dat de hele webpagina hiervoor moeten herladen (Onderzoeksvraag 3b).

Dit deelonderzoek beperkt zich enkel tot AJAX. Alternatieven van AJAX, voor zover die mogelijk zijn, zijn niet geanalyseerd. De opzet was om te achterhalen hoe en waar dat AJAX kon helpen in het opzetten van het GDSS platform.

### 7.2. Uitvoering

#### 7.2.1. Opstellen van een definitie

Om een goed beeld te krijgen van wat AJAX exact inhoudt en wat de functionaliteit hiervan is, is beroep gedaan op [FoundationsOfAjax, AjaxInAction]. Om de context van het ontstaan van AJAX en het ontstaan van de behoefte naar AJAX te onderzoeken is [Web20, BuildingRichWebApps, WebOS] bestudeerd.

#### 7.2.2. Vergelijken van data-overdracht technieken

Omdat AJAX toelaat om op verschillende manieren de data tussen de eindgebruiker en de webserver uit te wisselen, is gekeken naar welke techniek welke voor- en nadelen bevat.

#### 7.2.3. Analyseren van frameworks

Aansluitend is gekeken of een AJAX framework gebruikt kon worden bij de ontwikkeling van het GDSS platform, rekening houdende met de huidige en toekomstige eisen en wensen op het gebied van AJAX. Bij deze analyse zijn de voor- en nadelen van enkele AJAX frameworks tegen elkaar afgewogen.

#### 7.2.4. Integreeren van AJAX in GDSS Platform

Verder is onderzocht in welke onderdelen van het GDSS platform dat AJAX gebruikt kan worden (Onderzoeksvraag 3b). De criteria die bepalen of een bepaald onderdeel in aanmerking kwam om onderzocht te worden zijn eenvoudig. Indien een webpagina meervoudig gegevens moet controleren of versturen naar de webserver zonder dat de functionaliteit van de webpagina effectief moet veranderen, is het mogelijk interessant om gebruik te maken van AJAX. Dit zijn twee eenvoudige voorbeelden wanneer het aangewezen is om het gebruik van AJAX te overwegen:

1. De eindgebruiker ververs de webpagina enkel en alleen om te controleren of nieuwe gegevens beschikbaar zijn. Een mooi alledaags voorbeeld hiervan is de gebruiker die zijn webmail applicatie ververs om te kijken of er nieuwe emails beschikbaar zijn in zijn inbox.
2. De eindgebruiker verstuurt gegevens naar de webserver, waarop de webserver de oorspronkelijke webpagina terug weergeeft. Aangezien dezelfde webpagina wordt geladen, zijn er voor de eindgebruiker geen fundamentele veranderingen in functionaliteit te vinden. De workflow van de eindgebruiker is onderbroken, maar de nieuwe webpagina heeft geen toegevoegde functionaliteit te bieden.

## 7.3.Resultaten

### 7.3.1. Opstellen van een definitie en afbakenen van AJAX

AJAX, Asynchronous Javascript and XML, is een term die door J.J. Garrett werd bedacht [Garrett 2005]. In principe bestaat AJAX uit een verzameling van reeds bestaande technologieën [FoundationsOfAjax, AjaxInAction] :

- (X)HTML
- CSS (Cascading Style Sheets)
- Javascript
- DOM (Document Object Model)
- XML (of JSON)

Alhoewel deze verzameling van bestaande technieken op zich dus geen nieuwe technologie biedt, is het concept van AJAX echter wel ingrijpend. J.J. Garrett bedacht de term namelijk om aan te geven dat langzaam aan een nieuw soort van webapplicaties ontstaat: Webapplicaties waarbij dat de interactie tussen de gebruiker en de webpagina (bijna) even natuurlijk verloopt als in traditionele (desktop)applicaties.

Deze paradigmaverschuiving, van desktopapplicaties naar webapplicaties, wordt ook aangeduid als Web2.0 [Web20] en WebOS [WebOS], waarbij men de browser ziet als een platform waarop diverse applicaties kunnen draaien.

Eén van de meest gekende functies die helpt bij dit verhogen van interactie tussen de gebruiker en de applicatie is het XMLHttpRequest object dat teruggevonden kan worden in de standaard Javascript library van alle moderne webbrowsers. Aangezien het XMLHttpRequest object echter géén erkend W3C [W3C Home] standaard is, verschilt de implementatie van browser tot browser. Dit geeft echter geen aanleiding tot grote problemen bij het werken met dit object [FoundationsOfAjax].

In dit onderzoek hanteren we als definitie voor AJAX de definitie van [FoundationsOfAjax]: *"[...] AJAX is used to encompass all the technologies that allow a browser to communicate with the server without refreshing the page [...]"*.

De ontwikkeling van nieuwe invoerelementen waarmee de gebruiker kan interageren, zoals bijvoorbeeld een kalender control of een boomstructuur control, valt dus buiten dit gebied. Alhoewel deze nieuwe elementen toelaten de gebruiker op een efficiëntere manier te communiceren met de webapplicatie, verhinderen ze op zich niet dat de webpagina opnieuw geladen moet worden.

Tijdens discussies over Web2.0 wordt vaak ook gediscussieerd over AJAX. Het is echter belangrijk om een duidelijk onderscheid te maken tussen deze twee termen. Web2.0 is een verzameling van technieken en concepten die de gebruiker terug centraal stellen. Onder de term Web2.0 vallen ook vele sociale aspecten, zoals bijvoorbeeld het contact kunnen opnemen met andere gebruikers en het uitwisselen van favoriete links [Delicious Home] of foto's [Flickr Home]. Om deze concepten te kunnen verwezelijken kan gebruik worden gemaakt van AJAX, om de interactiviteit tussen de gebruiker en de webpagina te verhogen. Web2.0 is dus meer dan AJAX alleen.

### 7.3.2. Vergelijken van data-overdracht technieken

Omdat de gegevens tussen de webserver en de client op verschillende manieren kunnen worden uitgewisseld is het belangrijk om een bewuste keuze voor een bepaalde techniek te kunnen maken. De voor- en nadelen van een bepaalde keuze worden in volgende tabel op een rij gezet:

	Afhankelijkheid	Snelheid verwerking	Bandbreedte
XML	++	-- / -	-
JSON	0 / +	+	- / 0
Javascript functies	--	++	+

In [Bijlage 7] staat uitgelegd hoe dat de bovenstaande resultaten zijn bekomen.

### 7.3.3. Analyseren van frameworks

Het gebruik van een AJAX framework kan zinvol zijn om de volgende redenen:

- De implementatie van sommige Javascript functies is anders van browser tot browser. Een framework kan een uniforme manier van werken aanbieden
- Het kan vermijden dat het spreekwoordelijk wiel opnieuw wordt uitgevonden, door bestaande oplossingen aan te bieden.
- Het framework is opgebouwd uit code die zich reeds heeft bewezen (en dus al uitgebreid is getest)

Uit de literatuur [AjaxInAction, FoundationsOfAjax] blijkt al snel dat deze frameworks als paddenstoelen uit de grond schieten. Na een korte literatuur voorstudie uit [AjaxInAction, FoundationsOfAjax] bleven 3 frameworks over. De criteria die in deze voorstudie werden gebruikt zijn:

- Is het framework open source / gratis te verkrijgen
- Is het mogelijk om .NET te gebruiken in samenwerking met het framework (.NET is de ontwikkelingstaal die gekozen is voor het prototype.)
- Heeft het framework meer te bieden dan enkel een XMLHttpRequest wrapper

De AJAX frameworks die overbleven na deze eerste selectie zijn:

- Open Rico [Open Rico Home]
- Dojo [Dojo Home]
- Atlas [Atlas Home]

Om een gefundeerde keuze te maken uit deze grote verzameling van frameworks, is het belangrijk om de criteria voor de selectie goed vast te stellen.

- Om de gebruiksvriendelijkheid voor de participanten te verhogen zouden nieuwe invoerelementen beschikbaar moeten worden gemaakt. Deze invoerelementen dienen om op een eenvoudigere manier te interageren met het platform. De volgende invoerelementen zijn gekozen, omdat deze een meerwaarde kunnen bieden bij het beantwoorden van vragen:
  - Een kalender control
  - Een rich textbox control (Zie "Aandachtspunt 4" in Hoofdstuk 4)
  - Drag and drop support (Zie "Eis 19" in [Bijlage 6])
- Mogelijkheid tot het configureren van welke nieuwe invoerelementen geladen moet worden per webpagina. Hierdoor kan de laadtijd van de webpagina beperkt worden tot wat noodzakelijk is (Zie Niet-Functionele eis 3 in [Bijlage 6])
- Het beschikbaar stellen van de XMLHttpRequest via een eenvoudige interface (Het XMLHttpRequest is het hart van AJAX. Een goede ondersteuning van dit object vereenvoudigt het programmeren en verhoogt de onderhoudbaarheid van de code.)

Naast deze functionele criteria wordt ook gekeken naar hoe dat het framework wordt aangeboden. Deze criteria zijn:

- **Maturiteit** : in hoeverre is het framework volwassen en stabiel
- **Documentatie** : wordt er veel/ goede documentatie aangeboden
- **Extra features** : welke extra (unieke) features worden aangeboden die potentieel interessant zijn

Indien een bepaald criteria ontbrak, werd in de documentatie gekeken of er plannen waren om de functionaliteit in de (nabije) toekomst te implementeren.

	Dojo	Open Rico	Atlas
XMLHttpRequest	V	V	V
Configureren van het laden van controls	V		V
Kalender control	V		
Rich Textbox control	V		
Drag and Drop support	V	V	V
Ontstaan / Maturiteit	2004 -	2005 +	2006 ?
Documentatie	--	--	+
Extra features	-Javascript collections -Event system -Backbutton support	-Cinematic effects -Livegrid control	-ModalPopup control - Always Visible control
Nadelen	Groot in omvang		Enkel ondersteuning voor ASP.NET

Na deze analyse was het selecteren van het framework vereenvoudigd tot het afwegen van de positieve en negatieve kanten van de frameworks. Aangezien het Dojo framework beschikte over alle vereisten, was het dan ook een logische keuze om de ontwikkeling van het platform te verwezelijken met behulp van het Dojo framework.

### 7.3.4. Integreeren van AJAX in GDSS Platform

Aan de hand van de opgestelde criteria in 7.2.4 konden 3 onderdelen worden gemarkeerd waarin dat AJAX mogelijk een toegevoegde waarde kan bieden. Deze componenten worden verder beschreven in [Bijlage 7], waarbij een rationale wordt gegeven waarom en voor wie dat AJAX een meerwaarde kan betekenen (voor de ontwikkelaars / de facilitator / de participanten).

Dit is een beknopt overzicht van de gevonden resultaten:

#### **Opslaan en uitwisselen van antwoorden tijdens een bevringsronde**

Ten opzichte van Delphi implementatie biedt AJAX hier:

- Een gebruiksvriendelijkere interface voor participanten
- Meer ontwikkeltijd voor ontwikkelaars

Ten opzichte van de .NET herimplementatie biedt AJAX:

- Beter onderhoudbare code, omdat geen gebruik meer wordt gemaakt van hidden frames
- Dezelfde functionaliteit voor participant

#### **Communiceren via een op tekstgebaseerd kanaal**

Het communiceren via een op tekstgebaseerd kanaal is een nieuwe feature, waardoor het onmogelijk is het te vergelijken met eerdere implementaties. Wel kunnen de volgende conclusies worden gemaakt over het ontwikkelen van het communicatiekanaal:

- Praktisch zeer omslachtig zonder gebruik te maken van Flash / Java.
- AJAX vereenvoudigd de data-uitwisseling tussen de participanten onderling en tussen de participanten en de facilitator

#### **Opstellen van een bevragingronde**

Het opstellen van een bevragingronde is een feature die sterk uitgebreid is. Aangezien het platform zich richt op het dynamisch kunnen opstellen van de besluitvormingsprocessen, spreekt het voor zich dat het beheren van de bevragingrondes een complexe zaak is. Welke invloed dat AJAX heeft op deze complexiteit wordt hier kort toegelicht:

- AJAX versnelt het versturen en verwerken van de gegevens ten opzichte van traditionele webpagina's voor de eindgebruiker, aangezien de hele webpagina niet opnieuw moet worden geladen.
- AJAX zorgt voor een langere ontwikkeltijd voor ontwikkelaars (aangezien de code een stuk complexer is dan die van een traditionele webpagina)

## 8. Opstellen van de software architectuur

### 8.1. Onderzoeksaanpak

Een software architectuur wordt in [SoftwareArchitecture] omschreven als “[...] *the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them.*[...]”. Naast de beschrijving van deze structuur/ structuren, dient een software architectuur ook als communicatiemiddel tussen de verschillende betrokkenen tijdens het opzetten en het onderhouden van een software product.

Omdat deze betrokkenen hun eigen visie en belangen hebben ten aanzien van het te ontwikkelen product, worden verschillende invalshoeken gemaakt. Deze invalshoeken worden in [SoftwareArchitecture] “views” genoemd. Per view wordt eveneens een viewpoint gecreeërd, waarin wordt aangeduid welke conventies de view gebruikt ter notatie / illustratie.

Deze views (en hun corresponderende viewpoint) vormen samen met de beschrijvingen van de genomen architectuurbeslissingen de software architectuur. Dit deelonderzoek bestaat erin om de software architectuur te ontwerpen van het GDSS platform (Onderzoeksvraag 3c) op basis van de vastgestelde eisen (zie hoofdstuk 6: “Achterhalen van de eisen voor het platform”) en met de kennis van AJAX (zie hoofdstuk 7: “Onderzoeken van het potentieel van AJAX”). De daadwerkelijke implementatie details en ervaringen worden afgehandeld in hoofdstuk 9 (“Uitwerken van het prototype”).

### 8.2. Uitwerking

De onderstaande views zijn ontwikkeld in functie van de opgestelde persona's, die beschreven worden in [Bijlage 4]. Deze persona's geven de typische kenmerken weer van de doorsnee gebruikers / beheerders van het platform. Uitgezonderd van de overkoepelende view, is één view gerelateerd aan één persona.

#### 8.2.1. Overkoepelende / Facilitator view

Om een indruk te krijgen van het procesmatige van besluitvormingstechnieken is een view ontwikkeld die een beeld geeft van het proces dat typisch doorlopen wordt tijdens een besluitvorming. De view is de primaire view voor de facilitator, aangezien deze het meest bezig is met het procesmatige van de besluitvormingen.

Deze view dient tevens als overkoepelende view voor alle betrokkenen, aangezien de view schematisch laat zien hoe dat besluitvormingen in groep tot stand worden gebracht. De view zorgt ook voor een éénduidige terminologie voor de betrokkenen, zodat geen misverstanden meer kunnen optreden om bepaalde concepten aan te duiden.

Viewpoint: Facilitator	
Belanghebbenden:	Filip Verhagen (Facilitator, zie Persona's Bijlage 3)
Belangen:	-Procesmatige ondersteuning van besluitvormingsproces (++) De facilitator zijn grootste belang is dat hij/zij het procesmatige van het besluitvormingsproces kan ondersteunen  -Gebruikersvriendelijkheid (+) Aangezien de facilitator veel tijd zal spenderen bij het opzetten en analyseren van de ingegeven antwoorden, heeft deze een behoefte aan een intuïtieve omgeving
Representatiewijze:	Eén-veel representatie

### 8.2.2. Systeem- en netwerkbeheer view

De systeem- en netwerkbeheer view geeft weer hoe dat het GDSS platform verwezenlijkt moet worden op gebied van hardware en netwerkgebied. Deze view is dus als eerste instantie bedoelt om de systeem- en netwerkbeheerder(s) uit te leggen hoe dat het GDSS platform zal functioneren op hardware- en netwerkniveau.

De viewpoint is dan ook toegespitst op het persona van de systeem- en netwerkbeheerder Freja Dhont (zie [Bijlage 4]).

Viewpoint: Systeem- en Netwerkbeheer	
Belanghebbenden:	Freja Dhont (Systeem- en netwerkbeheerder, zie Persona's [Bijlage 4])
Belangen:	<ul style="list-style-type: none"><li>-Database-onafhankelijkheid (+) Het liefst zou het systeem database-onafhankelijk moeten zijn, zodat een migratie naar een ander databaseproduct geen (of zo min mogelijk) problemen teweeg brengt.</li> <li>-Stabiliteit (+) Het platform moet stabiel draaien, met zo min mogelijk interventie van de netwerkbeheerskant.</li> <li>-Data-uitwisseling met externe systemen(+) Het moet duidelijk zijn hoe dat het platform zal communiceren met andere (externe) systemen.</li></ul>
Representatiewijze:	Netwerk diagram

### 8.2.3. Ontwikkeling view

De ontwikkeling view geeft de relaties en indeling weer tussen de verschillende componenten waarover het GDSS platform beschikt. De viewpoint is ontworpen voor de persona van de ontwikkelaar, Steven Vervliet (zie [Bijlage 4]). Deze view is er ook op gericht om de externe koppeling met het KnoSoS-systeem duidelijk te maken.

Viewpoint: Ontwikkelaars	
Belanghebbenden:	Steven Vervliet (Programmeur, zie Persona's [Bijlage 4])
Belangen:	<ul style="list-style-type: none"><li>-Flexibiliteit(++) Aangezien ontwikkelaars in staat moeten zijn om uitbreidingen en/of aanpassingen te maken op de bestaande groepsbesluitvormingsprocessen, is het van belang dat de architectuur zo wordt opgezet dat extra processen kunnen worden toegevoegd.</li> <li>-Hergebruik van bestaande componenten(+) Om nieuwe groepsbesluitvormingsprocessen te ontwikkelen voor het platform, zou een generiek besluitvormingsproces ontwikkeld moeten worden waarop dat de andere besluitvormingsprocessen op kunnen inhaken/ verder werken.</li></ul>
Representatiewijze:	Component indeling

## 8.3. Resultaten

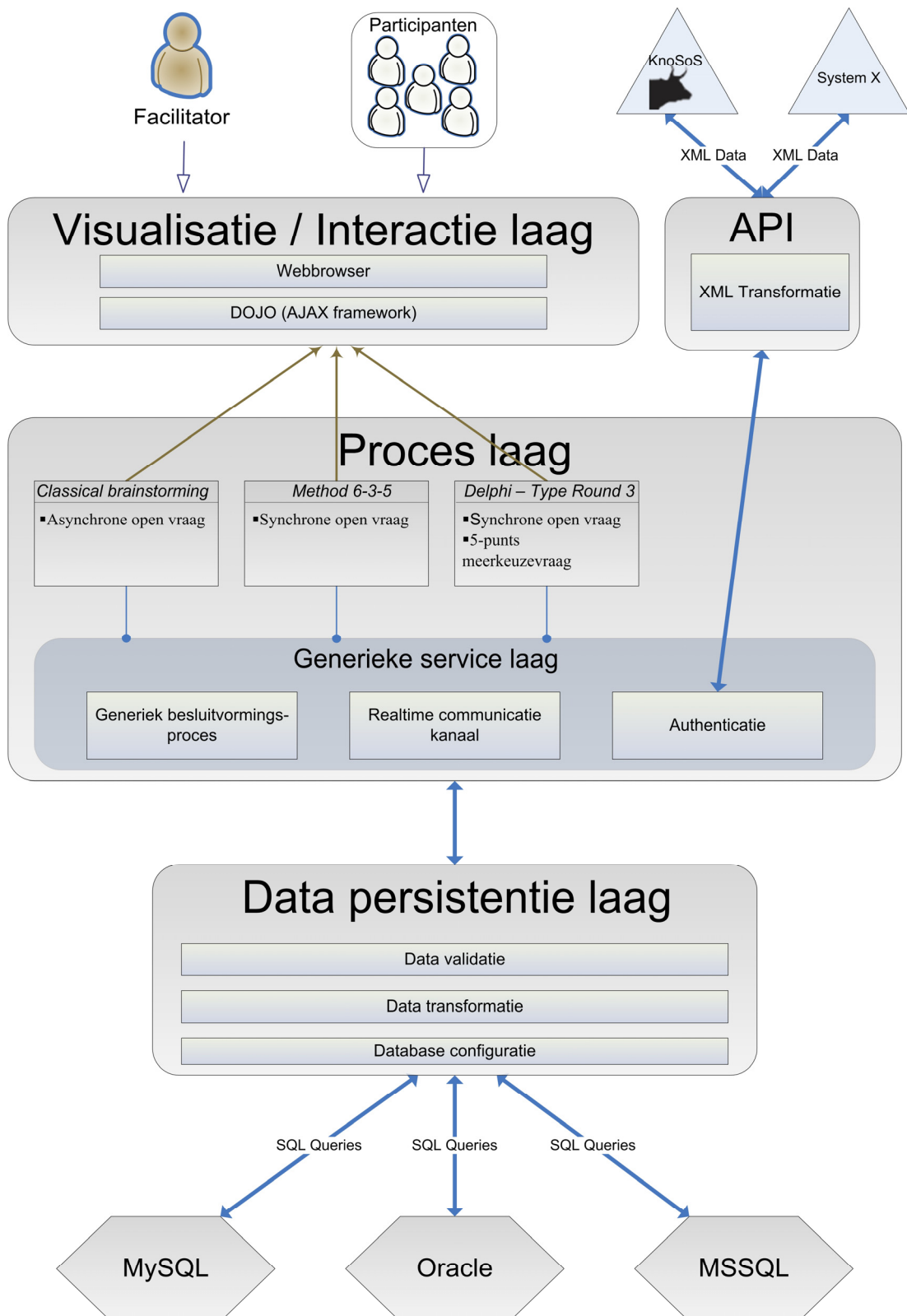
### 8.3.1. Views

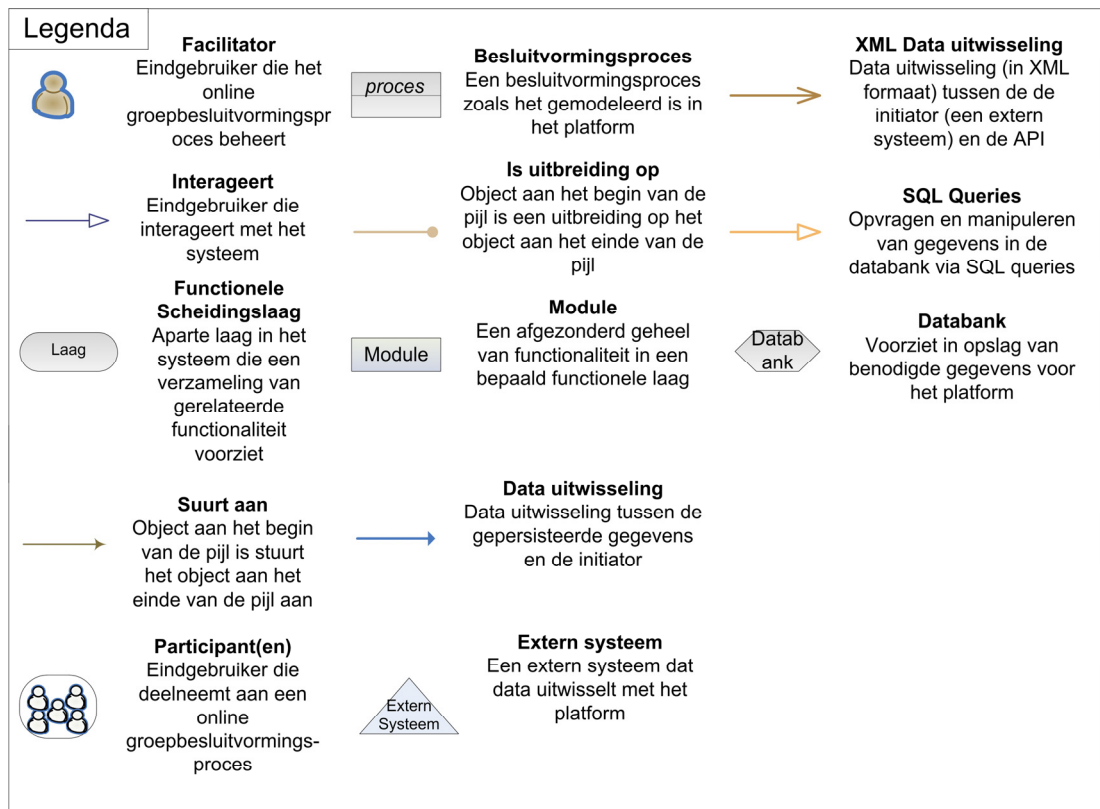
In dit hoofdstuk worden de verschillende views gegeven van de architectuur, uitgewerkt volgens de viewpoints uit hoofdstuk 8.2.





## Ontwikkeling view





**Figuur 8: Ontwikkelings view**

De ontwikkeling view wordt in [Bijlage 9] verder toegelicht, waarbij per onderdeel van de view een korte uitleg wordt gegeven over de functionaliteit die het onderdeel vervult.

### 8.3.2. Architectuurbeslissingen

Een aantal van de genomen architectuurbeslissingen worden hier gedocumenteerd, zodat het duidelijk is welke keuzes zijn genomen tijdens het opzetten van de software architectuur.

#### API als vorm van data-uitwisseling met externe systemen

Probleem	Gegevens uit het GDSS platform moeten beschikbaar worden gesteld aan (onder andere) het KnoSoS-systeem (zie ook Eis 18 "Beschikbaar stellen van gegevens voor externe systemen" in [Bijlage 6])
Genomen besluit	Een API opstellen waarmee externe systemen data kunnen opvragen. Deze API zal XML data teruggeven, zodat platformonafhankelijk de gegevens kunnen worden gebruikt.
Alternatieven	-Het platform volledige integreren met het KnoSoS-systeem.  -Gebruiken van een push mechanisme, waarbij dat het GDSS platform de gegevens automatisch verstuurt naar het KnoSoS-systeem (of een ander extern systeem) zodra nieuwe gegevens beschikbaar komen.
Argumentatie beslissing / voor-en nadelen	-Het opstellen van een API heeft de mogelijkheid om de beschikbare data in een later stadium redelijk eenvoudig uit te breiden.  -De API zorgt met een losse koppeling voor een betere scheiding tussen het platform en het KnoSoS-systeem. Hierdoor zijn veranderingen in één van deze systemen beter af te handelen.

	<p>-Indien een ander extern systeem ook gebruikt wenst te maken van het GDSS platform, moeten geen (of slechts een beperkt aantal) wijzigingen worden doorgevoerd in de API.</p> <p>-Ten opzichte van een volledige integratie is een API meer werk indien een uitbreiding van de data uitwisseling ontwikkeld moet te worden.</p>
Implicaties	<p>-In het KnoSoS-systeem zal een module moeten worden ontworpen die gegevens kan ophalen/verwerken van de API.</p> <p>-De data laag van het GDSS platform moet volledig gescheiden blijven van de besluitvormingsprocessen, zodat de API gegevens kan opvragen uit de data laag zonder dat deze expliciet kennis moet hebben van de besluitvormingsprocessen.</p>
Uitwerking / Resultaat in architectuur	De API is een aparte laag die zich boven de "Proces" laag bevindt. De API maakt gebruik van de "Authenticatie" module van de "Generieke proces" laag. Dit is te zien in de ontwikkeling view.

	API	Volledige integratie	Push van data
Data uitwisseling kan met meerdere systemen	++	-	0
Makkelijk uit te breiden	++	0	+
Verbruik van bandbreedte	-	++	--

#### Ondersteuning van meerdere databases

Probleem	De systeem- en netwerkbeheerder is verantwoordelijk voor het aanschaffen van licenties in het onderhouden van de databank. Voor haar is het belangrijk dat het mogelijk is om over te kunnen schakelen naar een andere databank zonder ingrijpende veranderingen.
Genomen besluit	Ondersteuning bieden voor verschillende databaseproducten.
Alternatieven	<p>-Geen gebruik maken van databanken, maar van files voor data opslag.</p> <p>-Slechts 1 database ondersteunen.</p>
Argumentatie beslissing / voor-en nadelen	Het kunnen ondersteunen van verschillende databases levert op zich niet erg veel extra complexiteit mee.. Het nadeel is echter dat bij per database extra code geschreven moet worden, waardoor (niet de complexiteit maar wel) de hoeveelheid aan code aanzienlijk zal verhogen. Dit zal de onderhoudbaarheid niet ten goede komen.
Implicaties	<p>-In de sourcecode zal een extra abstractie moeten worden toegevoegd die afschermt welke database daadwerkelijk wordt gebruikt.</p> <p>-Er zal een configuratiebestand aanwezig moeten zijn waarin wordt gespecificeerd welke database wordt gebruikt op het moment dat de databaseconnectie wordt geïnitieerd</p>
Uitwerking / Resultaat in architectuur	In de ontwikkeling view is duidelijk te zien hoe dat het configuratiebestand instaat voor het selecteren van een databaseproduct. De verschillende databaseproducten zijn ook zichtbaar aan de onderkant van de ontwikkeling view.

	Files	1 Database	Meerdere databases
Flexibel bij verandering van database / bestandsstructuur	--	-	++
Implementatietijd / Onderhoudbaarheid	++	0	-

### Ondersteunen van meerdere besluitvormingsprocessen

Probleem	Het ondersteunen van een rijk scala van groepbesluitvormingsprocessen is één van de vereisten aan het platform. Omdat deze besluitvormingsprocessen echter zeer uiteenlopend kunnen zijn, moet een manier worden gezocht om deze complexiteit te kunnen beperken. (Zie de bedrijfsdoelen 1 en 5 in hoofdstuk 6)
Genomen besluit	Opstellen van een generiek servicelaag dat de basiseigenschappen die voorkomen bij besluitvormingssystemen beschrijft en modeleert.
Alternatieven	-Alle besluitvormingsprocessen volledig uitwerken zonder rekening te houden met andere processen. Per proces dus een aparte implementatie.  -Systeem beperken tot 1 besluitvormingsproces.
Argumentatie beslissing / voor-en nadelen	Voordeel van het opstellen van een generieke service laag is dat terugkerende functionaliteit eenvoudig herbruikt kan worden. Het toevoegen of aanpassen van besluitvormingsprocessen zal ook aanzienlijk eenvoudiger worden wanneer bestaande functionaliteit wordt herbruikt.  Nadeel van een generieke servicelaag is echter dat een onderzoek moet worden ingesteld om deze kenmerken te achterhalen.
Implicaties	-Er moet een generiek servicelaag worden opgesteld waarvan alle besluitvormingsprocessen gebruik kunnen maken
Uitwerking / Resultaat in architectuur	Deze flexibiliteit wordt in de architectuur bekomen doordat een "Generieke service laag" zich bevindt in de "Proces laag". Deze "Generieke service laag" biedt wederkerige functionaliteit aan, zoals de "Generieke besluitvormingsproces", "Authenticatie" en "Realtime communicatiekanaal" module. De instanties van de besluitvormingsprocessen ("Method 6-3-5", "Classical Brainstorming",...) maken gebruik van deze generieke processen. Omdat zoveel mogelijk functionaliteit is afgezonderd in een generieke laag, kan er flexibel een extra proces worden toegevoegd, of kan een bestaand proces worden gewijzigd.  De "Generieke service laag" is in de ontwikkeling view terug te zien in de "Proces laag".

	1 besluitvormings- Proces	Generieke service laag	Ieder proces apart implementeren
Uitbreidbaar / Aanpasbaar	++	+	-
Hergebruik van functionaliteit	-	++	-
Ondersteund meerdere processen	--	++	+
Onderzoek- en Implementatietijd	++	--	-

### AJAX om interactiviteit te verhogen

Probleem	Participanten en facilitators verwerken veel gegevens. Deze gegevens worden in een typische applicatie geladen op het moment dat de pagina wordt geladen. In het platform zijn een aantal eisen (vb Eis 5: "Autosaving van antwoorden", Eis 8 "Ondersteuning synchrone intermenselijke communicatie", Eis 13 "Bekijken van geconnecteerde participanten") die ervoor zorgen dat de activiteiten van de participanten en facilitator realtime worden verwerkt / uitgevoerd zonder dat de hele webpagina zou moeten verversen.
Genomen besluit	Het inzetten van AJAX om deze interactiviteit te verhogen
Alternatieven	-Gebruik maken van Java applets om deze interactiviteit te verhogen  -Werken met verborgen frames om de gegevens door te sturen/op te Halen
Argumentatie beslissing / voor-en nadelen	Aangezien men in principe geen afhankelijkheden wenst van webtechnologieën die niet standaard in een webbrowser aanwezig zijn (zoals de Java runtime), is de oplossing beperkt tot het gebruik van hidden frames en AJAX. Het gebruik van hidden frames heeft als nadeel dat de code erg onoverzichtelijk wordt en het is bovendien lastig te implementeren. Hiernaast bieden de meeste AJAX frameworks naast deze asynchrone manier van gegevens versturen ook extra functionaliteit die het werken met Javascript in het geheel eenvoudiger maken. Een nadeel van AJAX is dat extra Javascriptcode moet worden geladen, waardoor de initiële laadtijd van de pagina groter kan zijn.
Implicaties	-Onderzoeken welk AJAX framework dat het meeste voordeel kan bieden en toch eenvoudig is in gebruik.  -Het hele platform wordt afhankelijk van Javascript. Gebruikers die Javascript (on)bewust hebben uitgeschakeld in hun browser verliezen functionaliteit of kunnen niet meer deelnemen aan de besluitvormingsprocessen.
Uitwerking / Resultaat in architectuur	Het AJAX framework is terug te vinden in de ontwikkeling view op de "Visualisatie / Interactie" laag als de "DOJO" module.

	Java applets	AJAX	Hidden frames
Eenvoudig te implementeren / te onderhouden	+	+	-
Onafhankelijk van plugin	--	++	++
Geeft extra functionaliteit	++	+	--
Geen extra kennis vereist	--	-	++
Eenvoudig functionaliteit herbruiken	++	+	--

De genomen architectuurbeslissingen worden geëvalueerd in [Bijlage 10]. De impact van de genomen architectuurbeslissingen op de architectuur worden besproken in [Bijlage 9].

## 9. Uitwerken van het prototype

### 9.1. Onderzoeksanpak

Het primaire doel van het uitwerken van een prototype is het valideren van de software architectuur (Onderzoeksvraag 3d) die is opgesteld in hoofdstuk 8 “Opstellen van de software architectuur”. Aan de hand van dit prototype moet duidelijk worden in hoeverre dat de opgestelde architectuur bruikbaar is om de de opgestelde eisen te kunnen verwezenlijken. Hierbij wordt het onderzoek afgerond, en kan worden onderzocht in hoeverre dat de onderzoeksvragen uit hoofdstuk 1 zijn opgelost.

Een secundair doel van het prototype is het ondersteunen van het onderzoek naar AJAX, dat reeds is beschreven in hoofdstuk 7: “Onderzoeken van het potentieel van AJAX”.

### 9.2. Uitvoering

De modules die worden weergegeven op de ontwikkelings view (zie hoofdstuk 8 “Opstellen van de software architectuur”) zijn (gedeeltelijk) uitgewerkt in het prototype. Aangezien voor de uitwerking van het prototype slechts een beperkte ontwikkelingstijd beschikbaar was, moest een duidelijk focus worden gemaakt van welke zaken wel en welke zaken niet in het prototype worden ontwikkeld. In 9.3.1 wordt uitgelegd welke onderdelen zijn uitgewerkt en waarom deze keuzes zijn gemaakt.

Het doel van het opzetten prototype is het valideren van de software architectuur. De fouten die aan licht kwamen tijdens deze validatie staan beschreven in [Bijlage 10]. De architectuurbeslissingen worden geëvalueerd aan de hand van het opgestelde prototype in [Bijlage 10].

### 9.3. Resultaten

#### 9.3.1. Visualisatie en Interactie laag

De “Visualisatie en interactie” laag stelt de eindgebruikers in staat om te interageren met het platform. Een schematisch overzicht van de activiteiten die de participanten en de facilitator kunnen uitvoeren zijn terug te vinden in [Bijlage 5]. [Bijlage 8] laat de daadwerkelijke uitwerking van deze activiteiten in het prototype zien door middel van screenshots en bijbehorende beschrijvingen.

#### 9.3.2. Besluitvormingsproces laag

De besluitvormingslaag wordt opgedeeld in twee verschillende delen. Een gemeenschappelijk gedeelte (het “Generiek besluitvormingsproces”, “Authenticatie” en het “Realtime communicatiekanaal”) en een specifiek gedeelte per proces. Het specifieke gedeelte per proces wordt gemodelleerd door de facilitator op het moment dat deze een besluitvormingsproces beheert. (Zie [Bijlage 5] voor een schematische voorstelling van deze activiteiten en [Bijlage 8] voor de uitwerking in het prototype).

De basisfuncties van een besluitvormingsproces worden geïmplementeerd in een ASP.NET webpagina (Participate.aspx). Deze pagina wordt geladen wanneer de participant begint aan de deelname van het besluitvormingsproces.

De pagina kijkt eerst of de participant voldoende rechten heeft om de pagina te bezoeken (“Authenticatie” module) en laadt vervolgens het besluitvormingsproces. Wanneer de participant zijn/haar ideeën en argumenten bewaart, worden deze gegevens asynchroon verstuurd en opgeslagen. Indien de participant contact wenst op te nemen met de facilitator

of met een andere participant, kan deze een bericht versturen ("Realtime communicatie kanaal").

### **9.3.3. Datapersistentie laag**

Al de inkomende gegevens worden doorgegeven aan de "Data persistentie" laag. Deze laag laadt, afhankelijk van het soort van de inkomende gegevens, een validatieprofiel. Dit profiel controleert of de ingevoerde gegevens binnen het opgegeven bereik liggen ("Data validatie" module). Indien al de gegevens gevalideerd zijn, worden deze gegevens doorgestuurd naar een transformatie module. Deze module converteert de inkomende data naar "datarow" objecten of naar "datarowCollection" objecten ("Data transformatie" module). Deze objecten zijn database-onafhankelijke objecten die één rij (datarow) of een collectie van rijen (datarowCollection) van een database tabel representeren. Aan de hand van de databaseconfiguratie worden deze objecten gebruikt om gegevens op te slaan, te wijzigen, te verwijderen of om gegevens op te halen uit de desbetreffende databank. ("Database configuratie" module).

De validatie van deze software architectuur wordt behandeld in [Bijlage 10].



## 10. Validatie en evaluatie onderzoekstraject

### 10.1. Validatie van onderzoekstraject

Het opzetten van het platform vond plaats in verschillende stappen: het opstellen van een generiek besluitvormingsproces, het achterhalen van de eisen, het opstellen van de software architectuur, etc. Bij deze stappen werd voortdurend met de begeleider getoetst of de resultaten strookten met de verwachtingen.

De gevonden resultaten werden onmiddellijk geïntegreerd in het verdere verloop van het onderzoek: het generieke besluitvormingsproces is in de architectuur geïntegreerd, de vastgestelde problemen uit het bestaande systeem zijn opgenomen als eisen, etc. Hierdoor vond onmiddellijk een eerste validatieslag plaats, aangezien meteen duidelijk werd of de gevonden resultaten bruikbaar en betrouwbaar waren of niet.

Het gebruik van reeds bewezen technieken op het gebied van (onder andere) requirements engineering [SoftwareRequirements] en software architecturen [SoftwareArchitecture] versterkte het vertrouwen bij de onderzoeksgroep dat het onderzoekstraject succesvol werd uitgevoerd. Om dit ook daadwerkelijk te kunnen bewijzen is een prototype opgezet van de software architectuur (zie hoofdstuk 9) die kon aantonen dat de gevonden resultaten valide zijn. Dit prototype zorgde ervoor dat een aantal tests konden worden uitgevoerd:

- 2 besluitvormingsprocessen werden opgezet, zodat aangetoond kon worden dat het systeem meerdere besluitvormingsprocessen ondersteund. Hiervoor werd een Delphi besluitvormingsproces en een Classical Brainstorming proces opgezet. De testomgeving werd echter niet uitgevoerd met echte participanten, maar het betrof slechts een simulatie.
- Een aantal gesimuleerde tests werd uitgevoerd om te kijken of AJAX voldoende krachtig was. Het asynchroon versturen van de antwoorden van de participanten, het modelleren van de vragen in een bevragsingsronde door de facilitator en het gebruiken van het realtime communicatiekanaal behoorden tot de uitgevoerde tests. Deze tests waren eerder technische van opzet, waarbij dat de mogelijkheden zo ver mogelijk werden onderzocht. Een uitgebreide performance/load test met echte participanten heeft niet plaats gevonden wegens tijdsgebrek.

Een aantal zaken konden niet worden gevalideerd binnen de doorlooptijd van dit onderzoek. Deze zaken zijn met name:

- Features die reeds in het vorige systeem aanwezig waren, en waarvan de functionaliteit niet drastisch is veranderd, zoals:
  - Importeren en exporteren van een besluitvormingsproces
  - Beheren van eindgebruikers
  - Toekennen van rechten aan eindgebruikers
- Features die oorspronkelijk op de planning stonden, maar waarvoor geen tijd meer kon worden vrijgemaakt, zoals:
  - Indelen van nieuwe informatie in categoriën tijdens het beantwoorden van een bevragsingsronde
  - Bekijken van geconnecteerde participanten door de facilitator

De overzichten uit [Bijlage 5] zorgden dat de opgestelde eisen deels gevalideerd konden worden. Door middel van het matrixdiagram dat de relatie tussen de bedrijfsdoelen en de eisen weergeeft en het workflow overzicht van de eindgebruikers, kan worden aangetoond dat de eisen volledig zijn: Indien er eisen ontbreken aan het systeem, dan zou dit in het workflow diagram aan het licht zijn gekomen. Indien er overbodige eisen zijn opgesteld, dan had het matrixdiagram dit aangetoond.

## 10.2. Wat is er bereikt

De resultaten van het onderzoekstraject kunnen als volgt worden samengevat:

- Een generiek besluitvormingsproces, wat de gemeenschappelijke kenmerken van besluitvormingsprocessen beschrijft, is opgesteld.
- De eisen en wensen ten aanzien van het platform zijn achterhaald en vastgelegd, rekening houdende met de sterke en zwakke punten van een voorgaand systeem.
- Het is duidelijk geworden wat AJAX kan betekenen voor het platform en waar dat het ingezet kan worden om de interactiviteit te bevorderen.
- Een software architectuur is opgesteld conform de vastgelegde eisen.
- De software architectuur is gedeeltelijk geïmplementeerd in een prototype, waarbij nog enkele verbeteringen voor het platform zijn voorgesteld.

## 10.3. Wat ging er mis

- Door tijdsgebrek is het prototype niet helemaal geïmplementeerd. Hierdoor is een uitgebreid testscenario (met echte participanten) uitgebleven.
- Een te grote scope in het begin van het traject zorgde voor veel overwerk. Tevens is een concrete doelstelling van het prototype te laat vast gesteld (welke delen implementeer ik wel, welke implementeer ik niet).
- Het kostte (te) veel tijd om diep in de materie te geraken van besluitvormingsprocessen in groepsverband. De echte hoogstaande en interessante discussies kwamen naar het einde van het traject toe, op het moment dat de architectuur moest worden vastgelegd. Het waren net deze discussies waarin nieuwe creatieve ideeën ontstonden.
- De prioritisering tijdens het uitwerken van het prototype was niet altijd goed. Soms werd teveel tijd besteed aan zaken die niet (direct) relevant zijn voor een prototype, zoals layoutproblemen (uitlijning, browserspecifieke eigenschappen), of werd teveel aandacht besteed aan code refactorings.

## 10.4. Bruikbaarheid voor de opdrachtgever

De resultaten in 10.2 “Wat is er bereikt” zijn bij uitstek bruikbaar voor de werkgever. Door middel van het opgestelde prototype zijn de betrokken onderzoekscentra in staat om het één en het ander te testen in functie van kennisoverdracht en besluitvormingsprocessen.

Het is nu aan hun om te beslissen of de ontwikkeling van het prototype wordt verdergezet, of dat de opgedane kennis wordt geassimileerd en vervolgens geïntegreerd in het KnoSoS-systeem. De opgestelde eisen en de software architectuur maken het in ieder geval een stuk eenvoudiger om over bepaalde concepten te redeneren en om nieuwe bepaalde ideeën te relateren aan het platform. De opgedane kennis van AJAX kan, naast het platform, ook worden gebruikt voor de ontwikkeling van het KnoSoS-systeem.

## 10.5. Reflectie op onderzoekstraject

Wanneer wordt teruggeblikt naar het doorlopen traject kunnen een aantal deficiënties worden vast gesteld. Deze deficiënties hebben in principe maar twee echte oorzaken:

1. Tijdsgebrek door onvoldoende afbakenen van het onderzoekstraject
2. Onvoldoende afgebakende doelstellingen

Deze twee punten zijn het gevolg van de evolutie van het onderzoekstraject. Bij het opstarten van het onderzoek lag de nadruk een stuk meer op het AJAX-gedeelte, naarmate het project evolueerde verschoof de focus meer en meer naar het opstellen van een software architectuur voor een flexibel platform.

Op zich is de evolutie van het onderzoektraject geen probleem, maar tijdens deze verschuivingen zijn de verwachte doelstelling niet (genoeg) mee geëvolueerd. Hierdoor bleef het gedurende een ruime tijd onduidelijk wat verwacht kon worden van het onderzoek.

Wanneer deze doelstellingen wel sneller zouden zijn vastgesteld, dan had het misschien mogelijk geweest om een prototype te bouwen waarin dat een volledige besluitvormingsproces werd uitgevoerd met participanten. Dit had een mooie testcase kunnen zijn voor het prototype.

## **10.6. Nieuwe wegen bewandelen**

Aan de hand van de geboekte resultaten kunnen nieuwe onderzoeken worden opgestart door de betrokken onderzoekscentra. Hierbij moet gedacht worden aan:

- Onderzoeksprojecten die kennisoverdracht door middel van besluitvormingsprocessen onder de loep nemen.
- Studies die de voor- en nadelen van synchrone besluitvormingsprocessen en asynchrone besluitvormingsprocessen aantonen.
- Onderzoeken die bekijken wat de fundamentele verschillen tussen besluitvormingsprocessen via een webbased systeem en een via face-to-face omgeving zijn.

## 11. Referenties

- [KnoSoSDeskresearch] D.G.A. Kenis: "Deskresearch Sociale Software", Interne publicatie, January 2006
- [VanGundy] A.B. VanGundy: "Techniques of Structured Problem Solving", Second Edition, Van Nostrand Reinhold, 1988
- [ParticipatoryToolkit] J. Elliott, S. Heesterbeek, C. J. Lukensmeyer, N. Slocum: "Participatory Methods Toolkit. A practitioner's manuel.", Koning Boudewijnstichting i.s.m. het Vlaams Instituut voor Wetenschappelijk en Technologisch Aspectenonderzoek (viWTA), September 2005
- [Kenis ImprovingGroupDecisions]D.G.A. Kenis: "Improving Group Decisions – Designing and Testing Techniques for GDSS Applying Delphi Principles", 1995
- [deSanctis] G.L. DeSanctis, R.B. Gallupe: "Group Decision Support System: A New Frontier", Data-Base, Winter, 1985
- [SDSS] M. Turoff, S. Hiltz, H.-K. Cho, Z. Li, Y. Wang: "Social Decision Support Systems (SDSS)," *hicss*, p. 11, 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 1, 2002.
- [FeelGood] W. Huang, V. Lai: "Can GSS Groups Make Better Decisions and Feel Good at the Same Time? A Longitudinal Study of Asynchronous GSS Groups," *hicss*, p. 1029, 34th Annual Hawaii International Conference on System Sciences ( HICSS-34)-Volume 1, 2001.
- [StimulatingThinking]K. M. Hilmer, A. R. Dennis: "Stimulating Thinking in Group Decision Making," *hicss*, p. 1028, 33rd Hawaii International Conference on System Sciences-Volume 1, 2000.
- [TheWisdomOfCrowds] J. Surowiecki: "The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations", Abacus, March 2005
- [SoftwareRequirements] S. Lauesen:"Software Requirements – Styles and Techniques", Addison-Wesley Professional, January 2002
- [InformationExchange] A. R. Dennis: "Information Exchange and Use in Group Decision Making: You Can Lead a Group to Information, but You Can't Make It Think", *MIS Quarterly*, p433, Vol. 20 Issue 4, Dec 1996
- [WikipediaSocialSoftware] Wikipedia.org: "Social Software". Wiki Entry, [http://en.wikipedia.org/wiki/Social\\_Software](http://en.wikipedia.org/wiki/Social_Software)
- [TheRiseOfSocialSoftware] M. Tepper: "The Rise of Social Software". *netWorker*, September 2003, pp. 18-23
- [Garrett Ajax] J. J. Garrett: "Ajax: A New Approach to Web Applications". Weblog Entry, February 2005, <http://adaptivepath.com/publications/essays/archives/000385.php>
- [FoundationsOfAjax] R. Asleson, N.T. Schutta: "Foundations of Ajax". Apress, 2006, ISBN 1-59059-582-3
- [Web20] O'Reilly: "What is Web2.0 - Design Patterns and Business Models for the Next Generation of Software", <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>

[AjaxInAction] D. Crane, E. Pascarello, D. James: "Ajax in Action". Manning, October 2005, ISBN 1-932394-61-3

[BuildingRichWebApps] L. D. Paulson: "Building Rich Web Applications with Ajax," *Computer*, vol. 38, no. 10, pp. 14-17, October 2005.

[WebOS] A. Weiss: "WebOS: Say Goodbye to Desktop Applications". *Computer*, December 2005, pp. 18-26

[SoftwareArchitecture] L. Bass, P. Clements, R. Kazman: "Software Architecture in Practice", Second Edition, Addison-Wesley Professional, April 2003

[GoF Design Patterns] E. Gamma, R. Helm, R. Johnson, J. Vlissides: "Design Patterns: Elements of Reusable Object-Oriented Software", Addison Wesley Professional, 1995.

[AlanCooperTheInmates] A. Cooper: "The Inmates Are Running the Asylum: Why High-Tech Products Drive Us Crazy and How to Restore the Sanity", Sams, February 2004

[JSON Wikipedia] Wikipedia.org: "JSON". Wiki Entry, <http://en.wikipedia.org/wiki/JSON>

[JSON Home] "Official JSON webpage", <http://www.json.org/>

[Dojo Home] "Dojo, the Javascript Toolkit", <http://dojotoolkit.org/>

[Open Rico Home] "Rico – JavaScript for Rich Internet Applications", <http://www.openrico.org>

[Atlas Home] "Official Atlas webpage", <http://atlas.asp.net/>

[W3C Home] "World Wide Web Consortium", <http://www.w3.org/>

[Delicious Home] "Del.icio.us Official webpage", <http://del.icio.us/>

[Drupal Home] "Drupal Official webpage", <http://drupal.org/>

[Flickr Home] "Flickr – The best way to store, search, sort and share your photos", <http://www.flickr.com/>

[KnoSoS Home] "Knowledge Sharing over Social Software", <http://www.knosos.be>

[CreativityAtWork] "Creativity at Work: Arthur B. VanGundy", <http://www.creativityatwork.com/CWServices/vangundy.htm>

## Bijlage 1: Woordverklaringen en definities

### **GDSS**

GDSS is de afkorting van Group Decision Support System. In dit onderzoek wordt de definitie van deSanctis en Galuppe [deSanctis] gebruikt : *“[A GDSS is] an interactive computer-based system that facilitates the solution of unstructured problems by a set of decision makers working together as a group”*.

### **GDSS Platform**

Het GDSS platform is een webapplicatie dat ondersteuning biedt aan participanten om op een interactieve manier ideeën te genereren en besluiten te nemen in groepsverband.

### **Sociale Software**

De term “sociale software” is moeilijk te omschrijven: er is niet een afgebakende en sluitende definitie van wat exact sociale software inhoudt en wat niet. In [WikipediaSocialSoftware] gebruiken ze de volgende definitie: “Social software enables people to rendezvous, connect or collaborate through computer-mediated communication and to form online communities”.

[TheRiseOfSocialSoftware] Heeft het over “Social software refers to various, loosely connected types of applications that allow individuals to communicate with one another and to track discussions across the Web as they happen”.

In [KnoSoSDeskresearch] gaat men ervan uit dat er (nog) geen sluitende definitie is. Hierbij probeert men eerder uit te leggen wat sociale software is door aan te tonen wat het te bieden heeft en uit welke technologieën het is opgebouwd: wiki's, blogs, RSS feeds, maar ook email en fora. Sociale software wordt dus opgebouwd uit zowel oude als nieuwe technologieën.

### **KnoSoS-systeem / KnoSoS omgeving**

Het KnoSoS (Knowlegde Sharing over Social Software) is een software systeem dat gebaseerd is op Drupal [Drupal Home]. Het systeem fungeert als hulpmiddel om het onderzoek naar kennisoverdracht over sociale software te ondersteunen. (Het KnoSoS onderzoeksproject).

### **Facilitator**

De begeleider en beheerder van een (online) discussie. Hij 'faciliteert' het groepsbesluitvormingsproces.

### **Participant**

Iemand die deelneemt aan het groepsbesluitvormingsproces als gebruiker van het platform om ( in een interactief proces) feedback te geven op ideeën van andere deelnemers en zelf antwoorden te geven op de vragen die worden gesteld.

### **Vraag / Antwoord**

Wanneer de facilitator aangeeft dat de participanten hun mening of ideeën moeten geven omtrent een bepaalde kwestie, wordt een “vraag” gesteld. Een vraag is van een bepaalde “vraagtype” (multiple choice, open vraag, ... ) en heeft een bepaalde interactiviteit (de vraag is synchroon of asynchroon).

Een vraag (en de antwoorden die hierop worden gegeven) moet ruimer gezien worden als dat van onze natuurlijke taal. Een vraag in het GDSS platform zou kunnen zijn: "Geef hier uw opmerkingen omtrent de volgende kwestie". De participant zal hierop zijn ideeën en meningen kunnen geven, dewelke in zijn geheel worden bestempeld als een "antwoord".

### **Synchrone / Asynchrone vraag**

Een synchrone vraag is een vraag waarbij het antwoord (het argument / het idee) van de participant rechtstreeks (in realtime) wordt doorgestuurd naar de andere participanten. Bij een asynchrone vraag wordt het antwoord niet rechtstreeks doorgestuurd naar de andere participanten.

### **Bevragingsronde**

Een bevragingsronde is een verzameling van vragen omtrent een bepaald probleem dat behandeld moet worden. De facilitator kan een bevragingsronde samen stellen door vragen te modelleren door participanten uit te nodigen voor de bevragingsronde (de discussie omtrent het probleem).

### **Groepsbesluitvormingsproces**

Een groepsbesluitvormingsproces bestaat uit één of meerdere bevragingsrondes. Om van de ene bevragingsronde over te gaan naar de volgende bevragingsronde wordt een analyse uitgevoerd door de facilitator. Deze bepaalt tijdens deze analyse welke antwoorden kunnen worden geherformuleerd tot nieuwe vragen. Deze kan (tijdens en) tussen de bevragingsrondes ook procesmatige informatie opvragen, zoals de gemiddelde doorlooptijd van de bevragingsronde, het percentage beantwoorde vragen, het aantal participanten die niet hebben deelgenomen aan de besluitvorming.

### **Synchroon / Asynchroon besluitvormingsproces**

Een besluitvormingsproces waarbij dat in één of meerdere bevragingsrondes synchrone of asynchrone vragen worden gesteld aan de participanten.

## **Bijlage 2: Opstellen van een generiek groepsbesluitvormingsproces**

### **2.1. Kwaliteit besluiten**

#### **2.1.1. GDSS communicatie vs face-to-face communicatie**

Het spreekt voor zich dat de communicatie tussen de eindgebruikers via de computer niet gelijkaardig verloopt als die in een face-to-face proces [Kenis ImprovingGroupDecisions]. De communicatie via PC's is informatie-arm : het bevat geen non-verbale communicatie, waardoor nuanceringen lastiger worden. De communicatie verloopt bijgevolg anders waardoor ook de kwaliteit van het besluitvormingsproces wordt beïnvloed. De participanten gaan immers, in vergelijking met een traditioneel face-to-face besluitvormingsproces, anders om met de aangeleverde informatie.

Eén van de grootste voordelen die een GDSS systeem biedt ten aanzien van een face-to-face besluitvorming, is dat de eindgebruikers anoniem hun mening kunnen uitbrengen, zonder dat een overste of een dominant persoon de leiding neemt over de discussie / besluitvorming [Kenis ImprovingGroupDecisions, InformationExchange].

Tegenover deze anonimiteit staat echter het risico dat participanten de aangeleverde informatie mogelijk niet (of minder) vertrouwen, omdat ze niet weten van wie deze informatie afkomstig is, en of deze informatie dan ook waardevol is. [InformationExchange]

Een ander groot voordeel van een GDSS systeem is dat gebruikers vanop diverse locaties kunnen deelnemen aan de besluitvorming, op voorwaarde dat ze een connectie kunnen opzetten met het systeem [Kenis ImprovingGroupDecisions]. Dit bespaart niet alleen in verplaatsingskosten, maar het zorgt er ook voor dat gebruikers (zeker voor synchrone besluitvormingsprocessen) makkelijker kunnen afspreken. Ze moeten zich niet verplaatsen naar één bepaalde vergaderzaal, maar kunnen van overal deelnemen. Dit verhoogt de kans dat de gebruikers ter beschikking kunnen zijn tijdens een besluitvormingsproces.

Een ander voordeel van een GDSS systeem, ten aanzien van klassieke face-to-face communicatie, is dat de geleverde informatie te raadplegen valt wanneer de participant dit wenst en zich mentaal kan concentreren op de nieuwe informatie. De gebruiker haalt de gegevens als het ware op uit het systeem, via een pull mechanisme [InformationExchange]). Bij traditionele face-to-face besluitvormingen wordt de informatie aangeleverd en moet de participant op dat moment aandachtig luisteren om de informatie te kunnen verwerken. [InformationExchange], waardoor het lastiger is om hun concentratie te behouden. Hierdoor zijn de participanten ook gelimiteerd: ze kunnen slechts één enkele actie tegelijk uitvoeren (communiceren of het verwerken van nieuwe informatie).

#### **2.1.2. Kennisuitwisseling voor het nemen van besluiten**

Voor het nemen van besluiten is een uitgebreide kennis nodig van het probleem (content) en de gerelateerde problematiek omtrent de kwestie (context). Zeker bij het oplossen van complexe problemen kan het zijn dat de benodigde kennis zich verspreidt over verschillende kennisdomeinen en bijgevolg bij verschillende personen aanwezig is. Om deze complexe beslissingen tot een goed einde te brengen, is het noodzakelijk dat de participanten in een besluitvormingsproces hun kennis aan elkaar beschikbaar stellen. Door te discussiëren en mogelijke oplossingen te geven stelt men zo de kennis ter beschikking.

Deze deling van kennis is de meerwaarde van een besluitvormingsproces in groep ten opzichte van een individueel besluitvormingsproces: men heeft toegang tot kennis waartoe men voordien geen toegang toe had. Essentieel hierbij is wel dat alle participanten hun kennis ter beschikking stellen.



Deze kennisdeling is echter niet iets vanzelfsprekends. Participanten delen namelijk (bewust of onbewust) niet al hun kennis en participanten nemen niet alle beschikbare informatie op [StimulatingThinking]. Hierdoor gaat een groot deel van de geleverde kennis verloren.

De factoren die hierbij een rol spelen zijn vooral :

- Onzichtbaarheid van de nieuwe aangeleverde informatie (onbewust niet alle informatie verwerken)
- Motivatieproblemen om de nieuwe informatie te verwerken (onbewust en/of bewust niet alle informatie verwerken)
- Niet alle kennis beschikbaar stellen aan andere participanten om de besluitvorming te manipuleren (bewust niet alle informatie delen)
- Overvloedig aanleveren van informatie, waardoor men door het bos de bomen niet meer ziet (onbewust teveel informatie aanleveren)

Naast deze kenmerken die in [StimulatingThinking] worden aangereikt, zijn er ook een aantal verschillen op te merken tussen de aard van de communicatie. De communicatie tussen de participanten verloopt immers via een informatie-arm medium. (zie “GDSS communicatie vs face-to-face communicatie”)

De motivatieproblemen om de nieuwe informatie te verwerken gaan meestal hand in hand met de overvloed van nieuwe gegevens. Indien een participant merkt dat hij de nieuwe gegevens niet kan verwerken, zal deze bewust/onbewust stoppen met opnemen en verwerken van nieuwe informatie, en zich toespitsen op de informatie waarover hij/zij reeds beschikt.

Deze informatie overvloed kan deels vermeden worden indien de nieuwe informatie wordt aangeboden in categorieën [StimulatingThinking] of wanneer de mogelijkheid wordt aangeboden aan de participant om zelf een indeling te laten maken (in bijvoorbeeld categorieën, of door de nieuwe informatie te rangschikken volgens toenemend belang, zodat de belangrijkste antwoorden bovenaan worden weergegeven).

Naast het feit dat de participant deze informatie beter kan verwerken omdat hij de informatie indeelt in categorieën, bevordert dit proces ook de integratie van de nieuwe informatie in het persoonlijk besluitvormingsproces van de participant. De participant is immers bezig met het analyseren van de nieuwe informatie, waardoor deze zich bewuster zal worden van de aangeleverde informatie [StimulatingThinking].

### **2.1.3. Tevredenheid van genomen besluiten**

[FeelGood] geeft aan dat tot 90% van de onderzoeken over tevredenheid van besluitvormingsprocessen via GDSS systemen moeten concluderen dat de tevredenheid in een groep niet stijgt (en meestal zelfs afneemt) ten opzichte van een face tot face besluitvorming. Opvallend hierbij is dat de grootte van een groep een invloed heeft op deze tevredenheid: bij grote groepen is men sneller tevreden over de genomen besluiten in een GDSS als bij kleine groepen.

Dit is toe te schrijven aan het feit dat kleine groepen (in deze context: minder als 5 personen) te lijden hebben onder twee negatieve neveneffecten van een GDSS systeem, met name:

- Het intypen van antwoorden is trager als verbale face-to-face communicatie
- Het overbrengen van een boodschap via tekst heeft slechts een beperkte expressiviteit: er kunnen geen/slecht emoties en nuances in worden opgenomen.

Grote groepen (5 of meer personen) kunnen echter meer profijt halen uit de voordelen van een GDSS systeem:

- Anonimiteit van personen tijdens het geven van hun antwoorden

- Parallel verlopen van het proces: er kunnen meerdere participanten tegelijk deelnemen aan éénzelfde discussie.

Bij kleine groepen wegen de positieve effecten niet op tegen de negatieve effecten, waardoor de groepstevredenheid over de besluitvorming lager uitvalt als bij face-to-face beslissingen. Een grote groep kan echter meer voordeel halen uit de positieve effecten dan een kleine groep [FeelGood]. Indien de negatieve effecten worden beperkt, kan bij grotere groepen wel een grote groepstevredenheid optreden.

Aangezien de snelheid van het intypen van de antwoorden bij de participanten wordt bepaald, kan een besluitvormingsproces hier geen verbeteringen in voorzien. Om de expressiviteit van de communicatie te verbeteren kunnen de invoervelden worden voorzien van “rich text controls”, waardoor de participanten, in plaats van enkel tekst, ook in beperkte maten opmaak kunnen geven aan hun antwoorden. Hierdoor kunnen ze meer nadruk leggen op ingevoerde tekst (het vet maken van bepaalde tekst, het cursief zetten van bepaalde tekst, het gebruik van opsommingstekens, etc).

Hiernaast kan een verhoogde groepscohesie bijdragen tot een verhoogde tevredenheid van de besluitvorming bij de participanten. Een verhoogde groepscohesie kan worden gevormd wanneer participanten zich “thuis voelen” in een bepaalde groep. Om een hogere groepscohesie te bereiken in een besluitvormingsproces waar de participanten elkaar nog niet kennen, zou de participant tenminste de profielen van andere participanten moeten kunnen bekijken. Hierdoor kan deze een indruk krijgen van wie er nog deel neemt aan de discussie. Een informele kennismaking en / of gesprek kan deze groepscohesie ook versterken.

## **2.2. De grootte van groepen**

### **2.2.1. Standaard GDSS: 5 à 10 personen**

Uit onderzoek van het boek [VanGundy], waarin in totaal 105 technieken worden beschreven om gestructureerd besluiten te kunnen nemen in groepsverband, blijkt dat een gemiddelde groep die aan een besluitvormingsproces deelnemen uit 5 à 10 participanten bestaat. In de sommige gevallen worden er geen beperkingen/richtlijnen opgegeven.

Bij sommige processen is deze limiet echter toe te schrijven aan de (organisatorische, communicatie) beperkingen die op treden tijdens face-to-face besluitvormingen. Het is namelijk zo dat een groep van 12 of meer personen niet meer rond één tafel kan zitten en elkaar kan aankijken tijdens het discussiëren. Vanaf dat moment moet meer structuur in de besluitvorming komen, door bijvoorbeeld de hand opsteken om het woord te vragen. Bij het nemen van besluiten in een digitale omgeving kan echter een groot deel van de verwerking van gegevens automatisch gebeuren, waardoor deze beperkingen gedeeltelijk kunnen worden vermeden.

### **2.2.2. Social Decision Support System**

In [SDSS] wordt gekeken in hoeverre dat het mogelijk is om een “Social Decision Support System” op te zetten, waarin duizenden mensen hun visie op een probleem kunnen geven. Hierbij wordt vooral gewerkt naar een democratische omgeving, waarin mensen contribueren om de standpunten omtrent een bepaald maatschappelijk dilemma in kaart te brengen.

Eén van de opmerkelijkste verschillen die de auteurs voorstellen, in tegenstelling tot een traditioneel GDSS, is dat ze een proces beschrijven waarbij geen facilitator nodig is om de discussie in goede banen te leiden.

*"[...] The process must be one that does not necessitate control by any single human being or a formal review group. Therefore, no single individual is to have the responsibility to edit or delete entries. [...]" [SDSS]*

Het ondersteunen van honderden tot duizenden mensen vraagt een aanzienlijke schaalbaarheid van het platform. Ook het zelfregulerend systeem waarbij geen facilitator meer nodig is om een proces te beheren stelt nieuwe vraagtekens bij reeds bestaande oplossingen.

Een mogelijke implementatie kan zijn om de gebruikers rechtstreeks argumenten van andere participanten te laten scoren. Hierbij kennen de participanten punten toe aan de argumenten van andere participanten. Deze punten kunnen gegeven worden op verschillende criteria, zoals bijvoorbeeld hoe interessant en hoe belangrijk een gegeven argument is. Overbodige of irrelevante argumenten kunnen zo "automatisch" weggefilterd worden door de participanten, zonder dat een facilitator het proces moet beheren. Omdat de participanten zelf de belangrijkste argumenten rangschikken van goed naar slechts, kan na een verloop van tijd een selectie worden gemaakt van de beste argumenten. Op basis van deze argumenten kunnen vervolgens enkele stellingen worden geformuleerd, dewelke de participanten kunnen beoordelen.

### **2.2.3. Wisdom of the Crowds**

Wisdom of the Crowds [TheWisdomOfCrowds] is een boek dat handelt over de intelligentie van een groep van mensen. De auteur gaat ervan uit dat een (grote) groep van mensen een probleem even goed, of zelfs beter, kunnen oplossen dan één of enkele experts.

Om deze beslissingen tot een goed einde te brengen, zijn twee basiseigenschappen vereist:

1. De groep moet groot genoeg zijn en een hoge mate van diversiteit hebben, waarbij iedereen een stuk unieke informatie bezit.
2. De groepsleden mogen onderling niet discussiëren, om groupthinking tegen te gaan. (een proces waarbij mensen snel geneigd zijn om hun antwoord aan te passen aan de reeds gegeven antwoorden)

Deze visie van besluitvorming is eigenlijk een radicale andere vorm van besluiten nemen dan het proces dat gebruikt werd in de vorige ontwikkelde systemen. Deze systemen baseerden zich namelijk op een vorm van Delphi waarbij enkel beroep wordt gedaan op experts en waarbij deze experts hun kennis ter beschikking stellen aan de rest van de groep. [TheWisdomOfCrowds] geeft echter aan om beroep te doen op een grote groep mensen, waarin zowel experts als niet-experts aanwezig zijn, en deze onafhankelijk een besluit te laten nemen.

*"[...]Chasing the expert is a mistake. We should stop hunting and ask the crowd (which, of course, includes the geniuses as well as everyone else) instead.[...]" [TheWisdomOfCrowds]*

Om dit soort van besluitvormingen te kunnen ondersteunen is het belangrijk dat het generieke besluitvormingsproces breed van opzet is, en dus niet afhankelijk is van één soort vaste opbouw van een proces.

Een selectie maken van experts, of het selecteren van een grote groep mensen, is een kwestie van het op voorhand vast leggen van welke personen dit probleem zouden kunnen oplossen. Dit selectieproces heeft op zich geen gevolgen voor het GDSS platform, maar is eerder een procesmatig / organisatorisch probleem voor de facilitator.

## **2.3. Opstellen van generiek besluitvormingsproces**

Uit de analyse van deze processen blijkt dat alle groepsbesluitvormingsprocessen een grote gemeenschappelijke deler hebben: alle besluiten worden genomen door gestructureerd

mensen aan het woord te laten op vastgelegde tijdstippen, of juist omgekeerd, door iedereen vrij te laten spreken. Tevens is er steeds een facilitator aanwezig, iemand die de discussie en het algemene besluitvormingsproces in goede banen leidt en beheert.

Meerdere processen hebben de mogelijkheid om personen anoniem te laten deelnemen. Deze anonimiteit zorgt ervoor dat gebruikers zich niet (of aanzienlijk minder) dominant kunnen opstellen dan bij traditionele face-to-face besluitvormingstechnieken. Het is immers niet duidelijk tijdens een anonieme besluitvorming wie de baas en wie een werknemer is. Hierdoor krijgt iedereen de kans om zijn/haar mening uitgebreid te geven, in plaats dat de baas, of een dominant persoon, beslist wie wanneer een inbreng in de discussie mag hebben (waardoor potentiële goede ideeën verloren kunnen gaan). De anonimiteit zorgt er ook voor dat in iteratieve processen, waarbij de participanten de mogelijkheid hebben om hun mening aan te passen, participanten geen gezichtsverlies kunnen lijden ten aanzien van de groep.

Bij het bestuderen en analyseren van de verschillende groepsbesluitvormingsprocessen werden enkele gemeenschappelijke kenmerken ontdekt. Deze kenmerken vormen op zich de ruggengraat van het toekomstige GDSS platform. De mate waarin deze kenmerken ondersteund worden in het GDSS platform bepalen het slagen of falen van het systeem

De basiskenmerken beschrijven een voldoende abstract proces dat steeds doorlopen wordt in een groepsbesluitvormingsproces. Wel dient opgemerkt te worden dat niet alle kenmerken noodzakelijk zijn om tot een besluitvorming te komen, en dus niet in alle processen per definitie terug komen. Daarom zijn een aantal punten voorafgegaan met een ster (\*). Deze ster wijst erop dat deze stap niet in alle processen terug te vinden zal zijn.

1. De facilitator moet (in overeenstemming met de klant, en eventueel al met de participanten) het probleem afbakenen door een consensus te bereiken over "wat is het probleem dat opgelost moet worden".
2. De facilitator moet het probleem publiceren/distribueren.
3. De facilitator selecteert de participanten en stuurt uitnodigingen naar deze participanten zodat deze aan de besluitvorming kunnen deelnemen.
4. (\*) De participanten bereiken een consensus over de probleemdefinitie indien dit nog niet is gebeurd.
5. (\*) Idee generatie deelproces:
  - a. De participanten doen aan "idee generatie" en bedenken zoveel mogelijke oplossingen / alternatieven voor het opgestelde probleem. Indien er geen oplossingen worden gegenereerd, dan worden de bestaande ideeën of alternatieven voorgelegd aan de participanten.
  - b. Onder leiding van de facilitator worden de ideeën verduidelijkt. (verheldering / afbakening / verbeteringen). Deze stap wordt enkel uitgevoerd indien in de vorige stap ideeën werden gegenereerd.
6. Participanten stemmen op bepaalde voorstellen, rangschikken ideeën volgens bepaalde criteria en elimineren de overbodige ideeën. Ideeën worden optioneel opgedeeld in categorieën.
7. (\*) Participanten krijgen de mogelijkheid om hun antwoord te herzien/te herformuleren nadat ze de feedback van andere participanten hebben gezien. Dit subproces kan eventueel meerdere keren worden herhaald.
8. Op basis van de eindstemmen kan de facilitator conclusies trekken en concrete implementatievoorstellen doen aan de opdrachtgever.

Een typerend kenmerk voor dat voor de meeste besluitvormingsprocessen terug komt, en wat ook in [VanGundy] wordt beschreven, is dat de bestudeerde processen ook gelijkaardige "fasen" doorlopen. In de eerste fase worden zoveel mogelijk ideeën gegenereerd (divergeren), bijvoorbeeld door middel van brainstorming of hitchhiking. In een tweede fase wordt naar een consensus toe gewerkt (afbakening van het probleem, voting en eliminatie). Deze fasen worden in een aantal processen iteratief doorlopen

## Bijlage 3: Analyseren van bestaand Delphi Systeem

### 3.1. Analyse Gebruikersproblemen

#### 3.1.1. Analyse support emails

In totaal werden 61 emails verdeeld over de verschillende categoriën. Deze mails waren afkomstig van 2 verschillende Delphi besluitvormingsprocessen. De opmerkelijkste bevindingen zijn:

- 19 participanten vroegen aan de facilitator een bevestiging om er zeker van te zijn dat hun antwoorden waren opgeslagen in het systeem. 4 van deze gebruikers stuurden een kopij van hun antwoorden “voor de zekerheid” als bijlage door.
- 25 participanten lieten weten dat ze geen tijd konden vrij maken om aan het besluitvormingsproces deel te nemen
- 1 gebruiker was zijn gebruikersnaam en wachtwoord vergeten.
- 13 participanten meldden dat de website ontoereikbaar was, of dat deze erg traag reageerde.
- 3 participanten waren geïnteresseerd in het eindresultaat van het besluitvormingsproces.

Slechts 57 van 159 uitgenodigde experts van het besluitvormingsproces SOC2004 antwoordden in ronde 2, terwijl dit in ronde 1 nog 87 van de 99 was. Ondanks dat in de analyses van de beheerders wordt aangetoond dat dit nog “relatief veel” is (gezien de tijdsdruk en volle agenda’s van de experts), lijkt dit een laag participatiepercentage (28%). Er werden van deze experts 5 emails ontvangen met de mededeling dat er geen tijd kon worden vrijgemaakt en 3 met de melding dat de expert “niets meer toe te voegen had” aan de discussie.

Dit hoog percentage van afhaken tussen ronde 1 en 2 kan toegeschreven worden aan tijdsgebrek, maar een mogelijke oorzaak kan ook gezocht worden in een ondermaatse gebruiksvriendelijkheid van de applicatie. De onzekerheid die de participanten hebben tijdens het werken met de applicatie zou verlaagd kunnen worden door de participanten op de hoogte te brengen indien de antwoorden succesvol zijn verwerkt in het systeem.

#### 3.1.2. Overzicht gebruikersproblemen beheerssectie

De beheerssectie van het Delphi systeem is simplistisch van opzet. De functionaliteit van het systeem is toegespitst op een chronologisch verloop van een Delphi discussie.

Wat meteen opvalt bij het doorlopen van deze functionaliteit, is de arbeidsintensiviteit die komt kijken bij het beheren van het Delphi proces. Een ongebruiksvriendelijke interface, die duidelijk geen rekening houdt met de wensen van de gebruiker, kan deze pijn niet verzachten.

Aangezien bij het analyseren van de antwoorden van de participanten alle gegevens worden gedumpt op het scherm, is het erg lastig om een overzicht te krijgen van welke informatie waar voor staat. Het toepassen van filters op een pagina, om de data te beperken tot het gewenste niveau, is niet mogelijk.

Verder is het voor de facilitator onduidelijk of een participant zijn/haar bevestigingsronde reeds heeft afgerond. De facilitator dient per gebruiker te kijken wanneer deze voor het laatst een antwoord heeft gegeven en moet handmatig controleren of deze al zijn/haar antwoorden heeft ingegeven in het systeem. Indien een participant reeds geruime tijd niet meer heeft gereageerd, en alle vragen zijn beantwoord, wordt verondersteld dat de participant klaar is met werken.

Uiteraard is dit een erg omslachtige en arbeidsintensieve manier om een overzicht te krijgen van wie reeds geantwoord heeft en wie nog niet. Daarbij komt nog eens dat niet alle participanten alle vragen wensen te beantwoorden, of dat participanten bijvoorbeeld na drie dagen rust hun antwoorden wensen na te lezen en te verbeteren waar nodig.

## **3.2. Analyse functionaliteit**

### **3.2.1. Beheerssectie**

Deze beheerskant van het systeem heeft twee onderdelen: het configureren van de “Delphi webserver” (standalone VisualWorks applicatie) en het beheren van de data (webapplicatie).

Het eerste wat opvalt bij het starten van het webbased beheerssectiegedeelte, is dat dit gedeelte geen inlogprocedure heeft. Hierdoor is de beheerssectie toegankelijk voor al wie de volledige URL kent.

De applicatie om de “Delphi webserver” te configureren, is een standalone VisualWorks applicatie. Het opzetten van een nieuw besluitvormingsproces vergt bijgevolg fysieke toegang tot de server, of er dient een remote desktop applicatie te draaien waartoe dat de facilitator toegang heeft. (Het toekomstige GDSS platform zal in ieder geval niet toelaten dat facilitators fysiek of remote desktop toegang krijgen tot de server. Het platform zal compleet webbased moeten zijn.)

Bij het aanpassen van bestaande participanten valt op dat de gebruikersnamen en wachtwoorden een vast patroon vertonen. De gebruikersnamen worden zo gemaakt dat de facilitator meteen kan zien wie de participant is. De wachtwoorden zijn niet veilig (kort, enkel cijfers), en vertonen een vast patroon. Elke gebruiker krijgt een uniek nummer toegekend als wachtwoord, bijvoorbeeld 500. Een volgende gebruiker zal in dit geval als wachtwoord 501 krijgen toegekend. Alhoewel hier nooit meldingen zijn gemaakt van misbruik, is deze vorm van inloggen uitermate ongeschikt voor het GDSS platform.

Verder valt op dat de hele applicatie ervan uit gaat dat slechts 1 besluitvormingsproces tegelijk plaats vindt. Het is onmogelijk om meerdere besluitvormingsprocessen op te zetten via dezelfde applicatie. Hiervoor moet een nieuwe instantie worden gestart van de applicatie. Aangezien de applicatie een geïntegreerde webserver heeft, zal deze op een alternatieve poort moeten draaien. Het is immers onmogelijk om twee webserver op dezelfde poort te draaien.

Een hidden feature die werd gevonden in de beheerssectie, maar die in feite zelden of nooit werd gebruikt, is de mogelijkheid om meerdere rondes van eenzelfde besluitvormingsproces simultaan te openen. Aangezien alle rondes handmatig gesloten moeten worden, is het (technisch gezien) mogelijk om meerdere rondes van een besluitvormingsproces naast elkaar te laten verlopen. Deze feature was nergens gedocumenteerd of was nog nergens ter sprake gekomen.

### **3.2.2. Gebruikersgedeelte**

Bij het doorlopen van de gebruikersgedeelte valt op dat er uitermate veel tekst op het scherm verschijnt. Het is namelijk zo dat alle vragen die worden gesteld in een bevragingronde worden opgesomd bovenaan de pagina. Deze vragen zijn gelinked en verwijzen door naar de vraagstelling waarop men kan antwoorden. Er wordt dus als het ware eerst een overzicht gegeven van al de vragen met daaronder per vraag de vraagstelling en de antwoordmogelijkheden.

De “opslaan” knop op de pagina bevond zich helemaal onderaan van de webpagina. Dit zorgt ervoor dat de participanten helemaal naar beneden moeten scrollen om gebruik te kunnen maken van de “opslaan” knop. Nadat de pagina is verstuurd, wordt een pagina

getoond waarop staat dat de participant de mogelijkheid heeft om uit te loggen of om terug te keren naar de vorige pagina. De gebruiker moet hiervoor gebruik maken van de “vorige” knop van de webbrowser, er is geen link voorzien. Indien de participant wenst verder te werken, moet de hele pagina met alle vragen worden herladen. Vervolgens moet de participant handmatig zoeken waar hij/zij was gebleven met het geven van antwoorden, zodat hij/zij verder kan werken. Dit is een uitermate omslachtige manier van werken.

Dit scrollen naar de “opslaan” knop en het herladen van de pagina zal voor verschillende participanten tot frustraties hebben geleid (verlies van kostbare tijd en concentratieverlies tussen het beargumenteren van verschillende vragen). Een oplossing die kan worden gebruikt om dit probleem weg te werken, is gebruik maken van een “opslaan” knop per vraag. Indien gebruik wordt gemaakt van Ajax [BuildingRichWebApps] is het zelfs mogelijk dat de antwoorden worden verstuurd zonder dat de hele webpagina moet worden herladen. Het spreekt voor zich dat deze twee verbeterpunten als eisen worden gesteld aan het GDSS platform.

## Bijlage 4: Persona's

De persona's die worden beschreven in deze bijlage zijn fictieve personen die typische eindgebruikers van het GDSS platform voorstellen. De kenmerken van deze persona's zijn als representatief bevonden bij het achterhalen van de typische kenmerken van de gebruikers van het platform. Het opstellen van deze persona's is gebeurd op basis van gesprekken met de facilitator van het bestaande Delphi systeem. Hiernaast is er gebruik gemaakt van de emails die werden geanalyseerd in hoofdstuk 5 "Analyseren van bestaand Delphi systeem" om de problemen en de belangen van de participanten te weten te komen.

Per persona wordt eerst nog geschetst waarom dat de opgestelde kenmerken en belangen nu typerend zijn voor die eindgebruiker, zodat het duidelijk is waarom een bepaald persoon die specifieke kenmerken krijgt toegekend.

### 4.1. Programmeur – Steven Vervliet

Voor de persona van de programmeur is een profiel opgesteld van een jonge en dynamische programmeur. Zijn belangrijkste interesse in het platform is het kunnen aanpassen van de besluitvormingsprocessen. Dit moet hij kunnen verwezelijken met slechts een beperkte kennis van besluitvormingsprocessen op zich, maar hij dient uiteraard wel kennis te hebben van de programmeeromgeving.

Naam	Steven Vervliet
Foto	
Leeftijd	28
Jobbeschrijving	Steven werkt al 5 jaar als programmeur bij EDS, waar hij voornamelijk als Java programmeur aan de slag gaat.
Opleiding	Steven heeft een bachelor in de Informatica gehaald. In zijn vrije tijd leest hij op geregelde basis vakliteratuur om bij te blijven.
Persoonlijkheid	Steven zijn opvallendste gedragskenmerken zijn:  -Leergierig: Steven is altijd bereid om bij te blijven op het gebied van technologie.  -Praktijk boven theorie: Steven richt zich eerder op de praktijk dan op de theorie. Hij wenst snel resultaten te kunnen boeken met de informatie die hij ter beschikking heeft.  -Sociaal: Steven is altijd bereid andere mensen te helpen en voelt zich het beste als hij omringt is door mensen
Technologie / webkennis	Steven heeft een redelijke ervaring met de programmertaal Java, maar heeft slechts een enkele keer in het .NET framework geprogrammeerd. Aangezien hij 2 websites onderhoudt (de website van zijn wielervereniging en die van de voetbalploeg van zijn jonge broer), heeft hij ook een redelijke kennis van HTML, CSS, PHP en MySQL. Steven houdt contact met zijn vrienden via email, MSN en Skype. Ook houdt hij een blog bij. Online shopping kent ook geen geheimen meer voor hem. Hierdoor heeft hij een goede feeling van wat er leeft op het



	internet.
Interesse / Belangen	<p>Steven is voornamelijk geïnteresseerd hoe hij, liefst zo snel mogelijk, een extra feature kan toevoegen aan het platform. Omdat hij praktijkgericht is, hoopt hij met zo min mogelijk werk snel resultaten te boeken.</p> <p>Aangezien hij het beste leert van bestaande voorbeelden, hoopt hij dat de meegeleverde features goed zijn gedocumenteerd. Hiernaast is hij gewoon om te werken met Javadoc. Een soortgelijke documentatie is volgens hem onmisbaar bij het aanpassen of toevoegen van een functionaliteit.</p>

## 4.2. Participant – Patrick Segers (Primaire Persona)

Het definiëren van een persona die zoveel verschillende kenmerken moet afdekken is niet eenvoudig. In overleg met de promotor is gekozen om het profiel van een ervaren manager op te zetten, iemand die typisch aan een besluitvormingsproces mee doet om zijn kennis te delen. Gezien de evolutie die managers (gemiddeld gezien) hebben doorlopen op het gebied van technologieën is een middenweg genomen tussen een digibeeet en een expert op het gebied van informatietechnologieën. De persona heeft dus een zekere (basis)kennis van computers, maar kan geen geavanceerde technologieën gebruiken. Dit profiel dekt de grootste lading van de participanten af: enige ervaring met PC's, maar geen ruime uitgebreide kennis.

Naam	Patrick Segers
Foto	
Leeftijd	58
Jobbeschrijving	Vice-Directeur "Van Loost Logistics"
Opleiding	Graduaat Expeditie.
Persoonlijkheid	<p>Patrick is al enkele jaren ondervoorzitter van "Van Loost Logistics". Deze functie bekwam hij ondermeer door zijn harde maar rationale aanpak op het gebied van bezuinigingen. Hij heeft een brede kennis van zowel politieke, sociale als economische kwesties en heeft op alle zaken een uitgesproken mening. Patrick is een denker en doorzetter. In zijn vrije tijd is hij fervent schaker in de lokale schaakvereniging.</p>
Technologie / webkennis	<p>Patrick is iemand die IT gebruikt voor de praktische zaken: Zijn PC gebruikt hij om inzichten te krijgen in de kosten en opbrengsten van zijn firma. Het controleren van zijn emailadres is voor hem het uitvoeren van een vast stappenplan: hij weet wanneer hij waar moet klikken, maar hij zal nooit trachten nieuwe functies te ontdekken in een programma.</p> <p>Op het gebied van websites is Patrick matig onderlegd. Hij bezoekt wekelijks de fansites van Bob Dylan en Tom Waits om op de fora zijn mening te geven en om nieuwigheden te ontdekken. Verder zoekt hij jaarlijks via Google zijn reisbestemming op. Het vastleggen van de vliegtuigtickets, het hotel, een huurauto, etc vindt hij echter te complex. Om deze klus op te knappen schakelt hij zijn zoon in.</p>
Interesse / Belangen	Patrick is in staat om de logica van een webpagina te begrijpen. Hij weet wat een knop is, wat een invoerveld is etc, maar hij kan zich ook

	mateloos ergeren als een pagina niet reageert zoals hij had gehoopt. Hierdoor is het voor hem uiterst belangrijk dat de vragen en antwoorden overzichtelijk worden weergegeven. Ook hoopt hij dat bij het starten van een ondervragingsronde er een soort van handleiding wordt gegeven. Aangezien Patrick een relatief trage typer is, hoopt hij niet benadeeld te worden tijdens het deelnemen van online discussies.
--	---

### 4.3. Facilitator – Filip Verhagen (Primaire Persona)

Het vastleggen van het profiel van de doorsnee facilitator is lastig, aangezien er in principe twee verschillende soorten groepen van facilitators zijn. De ene groep bevat facilitators die beroepsmatig groepsbesluitvormingen ondersteunen en zich daar dan ook in hebben gespecialiseerd, terwijl de andere groep bestaat uit mensen die het platform ad hoc gebruiken om besluiten te kunnen nemen. Om geen tweede persona te moeten opstellen, is daarom beslist om in één persona de beide kanten toe te lichten. Daarom werkt de volgende persona als crisismanager waar hij besluiten moet nemen (waarbij op ad hoc basis gebruik kan worden gemaakt van groepsbesluitvormingssysteem) in groepsverband, met hiernaast een éénpersoonszaak waarbij dat hij consultancy doet over besluitvormingen in groepsverband.

De opleiding communicatiewetenschappen duidt erop dat de facilitator een achtergrond heeft in communicatietechnieken en verschillende manieren kent om te kunnen overleggen in groep.

Naam	Filip Verhagen
Foto	
Leeftijd	46
Jobbeschrijving	Filip werkt al 15 jaar bij ContourIT, een bedrijf dat consultancy geeft aan (IT-)bedrijven in nood (crisismanagement). Hierin staat Filip in voor het nemen van (cruciale) beleidsbeslissingen (herstructureringen, aanpassen van plannings, etc). Soms moeten deze beslissingen zelfs worden genomen nog voor dat Filip de kans heeft om het probleem zelf helemaal te analyseren. Naast deze job heeft Filip 3 jaar geleden een éénmanszaak opgericht dat zich specialiseert in het nemen van besluitvormingen in groepsverband.
Opleiding	Filip is licentiaat (Master) in de communicatiewetenschappen en heeft een postlicentiaat (Master-na-Master) Bedrijfsbeheer gehaald.
Persoonlijkheid	<p>Typerende gedragskenmerken voor Filip zijn:</p> <ul style="list-style-type: none"> <li>-Orde in chaos: Filip is goed in staat om in een chaos van gegevens en opeenstapeling van problematiek de essentie van het probleem te vinden. Hij weet ook de juiste personen te contacteren die hem kunnen bijstaan in een besluitvormingsproces. Dit maakt hem dan ook een goed crisismanager.</li> <li>-Rustig: Filip weet bij de meest cruciale beslismomenten zijn hoofd</li> </ul>

	<p>koel te houden en de juiste beslissingen te nemen.</p> <p>-Openminded: “Het doel heiligt de middelen” is een uitspraak die op Filip zijn lijf staat geschreven. Filip vermijdt geen nieuwe technieken of ideeën en staat open voor alles wat mogelijk kan bijdragen tot een succesvolle afronding van een project.</p>
Technologie / webkennis	<p>Filip is gesteld op IT. Hij gebruikt dagelijks zijn emailprogramma (Outlook 2003), waarin hij al de inkomende en uitgaande mails sorteert per project. Ook maakt hij uitgebreid gebruik van de geïntegreerde agenda-functionaliteit. Deze agenda synchroniseert hij dagelijks met zijn PDA om geen afspraken te missen.</p> <p>Filip is een ervaren websurfer: hij surft dagelijks naar websites, voornamelijk naar nieuwssites en zoekmachines. Zelf heeft hij geen kennis van HTML of enige andere programmeer/scripting taal.</p>
Interesse	<p>Filip is geïnteresseerd in het platform om gestructureerde beslissingen te kunnen nemen in zijn functie van crisismanager. Hiernaast werkt hij ook op ad-hoc basis aan projecten waar in groep besluiten moeten worden genomen. Hiervoor werkt hij op dit moment met face-to-face meetings. Filip ziet in een online besluitvormingsplatform veel potentieel en is geïnteresseerd in de doeltreffendheid van dit systeem.</p>
Belangen	<p>Aangezien Filip voornamelijk bezig zou zijn met het opstellen van de vragen en met het analyseren van de antwoorden, zijn voor hem de gebruiksvriendelijkheid van de beheerssectie een belangrijk punt. Verder is het voor hem belangrijk dat hij een vrijheid heeft om een keuze te kunnen maken welke besluitvormingsproces hij wenst te selecteren.</p>

#### 4.4. Systeem- en Netwerkbeheerder – Freja Dhont

Alles wat met hardware en netwerken te maken heeft is toegekend aan de “Systeem- en Netwerkbeheer” persona. Deze heeft uiteraard als primaire interesse de structuur en de opbouw van het platform vanuit een hardwarematige invalshoek. Deze persona heeft een vrij voor de hand liggende achtergrond (Systeem- en netwerkbeheer opleiding).

Naam	Freja Dhont
Foto	
Leeftijd	25
Jobbeschrijving	Freja
Opleiding	Bachelor Systeem- en netwerkbeheer.
Persoonlijkheid	Freja is een zelfbewuste, onafhankelijke vrouw. Ze is de enige vrouw in haar bedrijf, maar dat zorgt ervoor dat ze steviger in haar schoenen staat. Ze is erg communicatiefvaardig en altijd goed gezind.
Technologie / webkennis	Freja heeft veel ervaring op het gebied van onderhoud van PCs en netwerken. Al vanaf haar jeugd werd ze van thuis uit met PCs geconfronteerd. Het aanpassen van de instellingen van een PC fascineerde haar al erg snel. Haar opleiding Systeem- en netwerkbeheer legde ze dan ook af met grote onderscheiding. Sinds ongeveer drie jaar heeft Freja een voorliefde gekregen voor alles wat opensource is.

	<p>Freja surft dagelijks enkele uren. Zowel tijdens als buiten haar werkuren bezoekt ze technologie gerelateerde websites om up to date te blijven. Ook fora zijn geen onbekende voor haar. Zelf heeft ze geen ervaring in het maken van websites (buiten de paar uur HTML die ze tijdens haar opleiding heeft gekregen).</p>
<p>Interesse / Belangen</p>	<p>Freja is voornamelijk geïnteresseerd in de stabiliteit van de software die gebruikt zal worden voor het GDSS platform, aangezien ze verantwoordelijk wordt gesteld voor de uptime van de systemen. Ook is de communicatie die plaats vindt tussen het GDSS platform en andere systemen voor haar belangrijk, omdat zij ook die verbindingen moet beheren. Hiernaast heeft ze een sterke interesse in de manier hoe dat de data opslag zal plaats vinden van het platform, aangezien ze ook instaat voor het aanschaffen van eventuele licenties voor een databank.</p> <p>Indien grote besluitvormingen moeten plaats vinden, zal ze ook willen kunnen zien hoe dat de schaalbaarheid van het platform gewaarborgd kan worden, zodat performantieproblemen vermeden kunnen worden.</p>



## 5.2. Matching van eisen aan bedrijfsdoelen

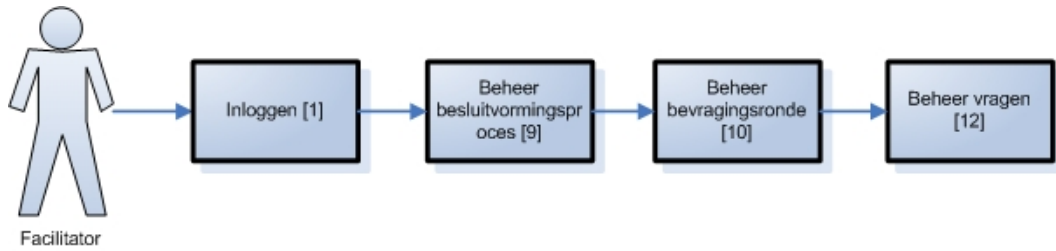
	Ondersteuning bieden v voor onderzoek naar groepsbesluitvorming	Onderzoek naar synchrone intermenselijke communicatie	Onderzoek naar synchrone data transfer	Het groepsbesluitvormingsproces ondersteunen	Ruimte bieden voor uitbreidingen en aanpassingen	Koppeling met KnoSoS platform
<b>Functionele eisen</b>						
1 Inloggen (participant & facilitator)				x		
2 Selecteer een bevragsingsronde				x		
3 Antwoorden ingeven tijdens een bevragsingsronde			x	x		
4 Afsluiten van een bevragsingsronde				x		
5 Autosaving van antwoorden	x		x			
6 Beheren van eindgebruikers				x		
7 Toekennen van rechten aan eindgebruikers				x		
8 Ondersteuning synchrone intermenselijke communicatie	x	x				
9 Beheren van een besluitvormingsproces				x		
10 Beheren van een bevragsingsronde				x		
11 Importeren en exporteren van een besluitvormingsproces				x		
12 Beheren van vragen	x		x	x	x	
13 Bekijken van geconnecteerde participanten	x	x	x			
14 Exporteren van bevragsingsronde voor analyse	x			x	x	
15 Tonen van progressiebalk	x					
16 Beheren van profielpagina	x					
17 Bekijken van profielenpagina	x					
18 Beschikbaar stellen van gegevens voor externe systemen						x
19 Indelen van nieuwe informatie in categorieën			x	x		

## 5.3. Workflow overzicht

Voor een overzicht te hebben van welke eisen tot welke acties leiden, wordt hier een overzicht gegeven van de workflow die de facilitator en de participant typisch doorlopen bij een bevragsingsronde. Het cijfer dat wordt aangeduid in een bepaalde stap refereert naar de overeekomstige eis. (vb "Inloggen [1]" verwijst naar Eis nummer 1. De volledige gedetailleerde eisen zijn terug te vinden in [Bijlage 6]).

### 5.3.1. Facilitator

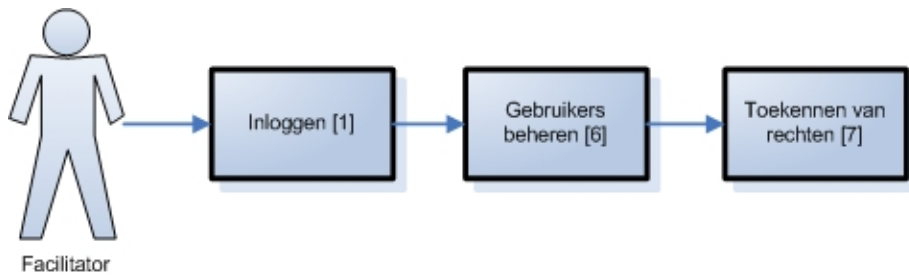
#### Opzetten van een bevragsingsronde



**Figuur 1: Opzetten van een bevragsingsronde**

Figuur 1 beschrijft het opzetten van een bevragsingsronde. De facilitator kan, na het inloggen (stap 1: "Inloggen"), een nieuw besluitvormingsproces aanmaken (stap 2: "Beheer besluitvormingsproces"). Indien hij/zij dit heeft aangemaakt, of indien hij/zij een bestaand besluitvormingsproces heeft geselecteerd, kan hij/zij een nieuwe bevragsingsronde opstellen (stap 3: "Beheer bevragsingsronde"). Zo een bevragsingsronde wordt opgebouwd uit één of meerdere vragen (stap 4: "Beheer vragen").

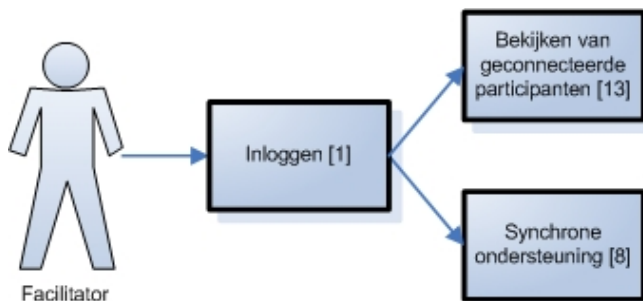
#### Beheren van eindgebruikers



**Figuur 2: Beheren van eindgebruikers**

Figuur 2 beschrijft hoe dat de facilitator de participanten kan beheren. Na het inloggen (stap 1: "Inloggen"), kan de facilitator nieuwe participanten aanmaken die moeten kunnen deelnemen aan het besluitvormingsproces (stap 2: "Gebruikers beheren"). Vervolgens kan hij/zij rechten toekennen aan de gebruikers die toegang moeten verkrijgen voor het deelnemen aan het besluitvormingsproces (stap 3: "Toekennen van rechten").

#### Ondersteuning bieden tijdens een bevragsingsronde

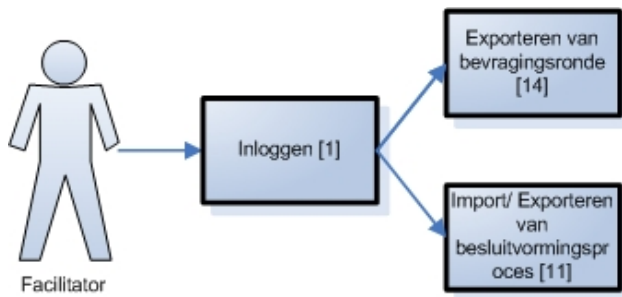


**Figuur 3: Ondersteuning bieden tijdens een bevragsingsronde**

Figuur 3 geeft de acties die de facilitator kan doen nadat hij/zij is ingelogd (stap 1: "Inloggen") om de participanten te ondersteunen in een bevragsingsronde. Hierbij kan de facilitator kijken welke participanten op dat moment zijn geconnecteerd, en welke participanten klaar zijn met het beantwoorden van de bevragsingsronde (stap 2a: "Bekijken

van geconnecteerde participanten”). Om hen te kunnen ondersteunen kan de facilitator realtime berichten uitwisselen met de participanten (stap 2b: “Synchrone ondersteuning”).

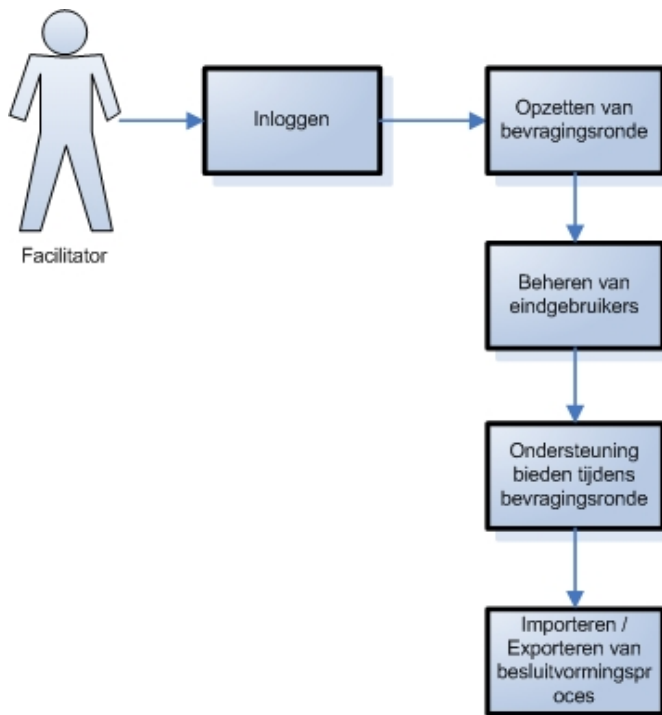
#### Importereren of exportereren van data



**Figuur 4: Importeren of exportereren van data**

Figuur 4 beschrijft de stappen die een facilitator kan ondernemen om de gegevens in het platform te beheren. Hiervoor kan hij/zij bij het inloggen (stap 1: “Inloggen”), gebruik maken van een exporteerfunctie die de antwoorden van een bevragsingsronde exporteert (stap 2a: “Exporteren van een bevragsingsronde”). Hierdoor kan de facilitator de gegevens in een andere applicatie terug importeren, zodat hij/zij de analyse indien gewenst in een separate applicatie kan uitvoeren. Op het einde van een besluitvormingsproces kan de facilitator een heel besluitvormingsproces exporteren (stap 2b: “Import/Exporteren van besluitvormingsproces”). Hierdoor kan deze het hele proces eenvoudig backuppen en later eventueel terugzetten.

#### Facilitator – Overview van workflow



**Figuur 5: Workflow van een facilitator**

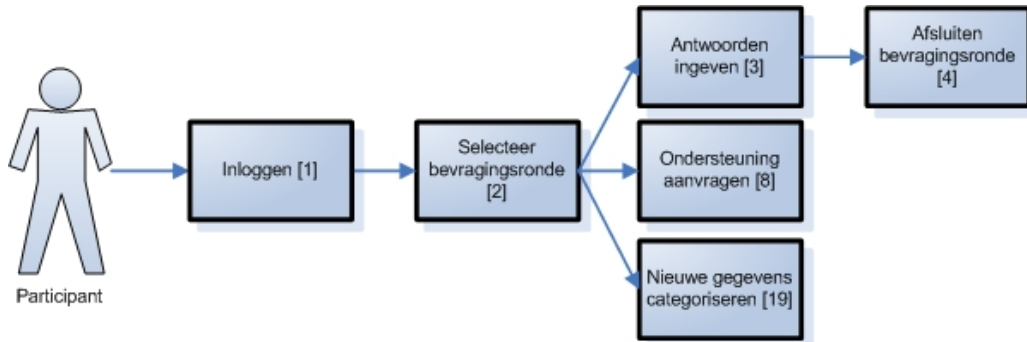
Indien de workflow van een facilitator wordt bestudeerd, zal dit een typisch verloop van de workflow zijn. Na het inloggen zal de facilitator de volledige bevragsingsronde opzetten (zie “Opzetten van een bevragsingsronde”). Hierna kan de facilitator toegangsrechten verlenen aan de participanten (zie : “Beheren van eindgebruikers”), waarna hij/zij de participanten daadwerkelijk kan ondersteunen tijdens een bevragsingsronde (zie: “Ondersteuning bieden tijdens een bevragsingsronde”). Tenslotte kan hij/zij na een bevragsingsronde de gegevens



exporteren. Na het afronden van een volledig besluitvormingsproces kunnen alle gegevens met betrekking tot dat besluitvormingsproces worden geëxporteerd. (zie :”Importeren/Exporteren van besluitvormingsproces”).

### 5.3.2. Participant

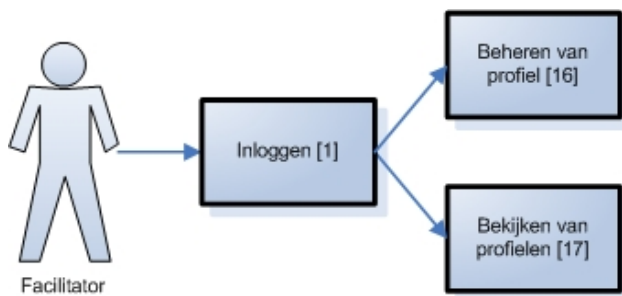
#### Contribueren aan een bevestigingsronde



**Figuur 6: Contribueren van een participant aan een bevestigingsronde**

Figuur 6 toont hoe een participant een bevestigingsronde afwerkt. Nadat hij/zij is ingelogd (stap 1: “Inloggen”), kan hij/zij een bevestigingsronde selecteren waaraan hij wenst deel te nemen, en waarover hij/zij de nodige rechten beschikt (stap 2: “Selecteerd bevestigingsronde”). Hierna heeft de participant de mogelijkheid om antwoorden in te geven op de opgestelde vragen (stap 3a: “Antwoorden ingeven”), om ondersteuning te vragen aan de facilitator (stap 3b: “Ondersteuning aanvragen”) of om de nieuwe gegevens die binnen komen (enkel van toepassing op processen waarbij synchroon antwoorden worden verstuurd tussen de participanten) te rangschikken in categoriën (stap 3c: “Nieuwe gegevens categoriseren”). Nadat de participant tevreden is met al zijn gegeven antwoorden, heeft deze de mogelijkheid om de bevestigingsronde voor hem af te sluiten (stap 4: “Afsluiten bevestigingsronde”). Hiermee geeft hij aan dat hij klaar is met het uitvoeren van de bevestigingsronde.

#### Profielpagina beheren of bekijken



**Figuur 7: Profielenpagina beheren of bekijken**

Figuur 7 toont hoe dat een participant, nadat hij/zij is ingelogd (stap 1: “Inloggen”), zijn profielenpagina kan beheren (stap 2a: “Beheren van profiel”). Hierin krijgt hij/zij de mogelijkheid om zijn/haar gegevens up-to-date te houden. Daarnaast kan hij/zij ook de profielen van de andere participanten opvragen (stap 2b: “Bekijken van profielen”).

## Bijlage 6: Belangrijkste eisen aan het platform

### 6.1. Functionele eisen

#### 6.1.1. Inloggen (participant & facilitator) [Eis 1]

Prioriteit:	Must	
Doel:	Gebruikers autoriseren (identificeren) en authenticeren (rechten verlenen), dit houdt in: - Onderscheid maken tussen typen gebruikers (facilitator/participant) - Uitsluitend toegang verlenen aan bevoegde gebruikers	
Actors en interesse:	<p><i>Facilitator:</i> Toegang tot platform verkrijgen om besluitvormingsrondes op te starten en te beheren (analyseren van antwoorden en opstellen van nieuwe vragen) en het verkrijgen van een overzicht van welke participanten zijn ingelogd en welke participanten klaar zijn met het beantwoorden van een bevragingronde</p> <p><i>Participant:</i> Toegang tot platform verkrijgen om deel te nemen aan een bevragingronde.</p>	
Omschrijving:	De gebruiker voert zijn/haar loginnaam en wachtwoord in op de login pagina, waarna deze worden doorgestuurd naar het platform. Indien de gegevens valide zijn wordt de gebruiker toegang verleend en doorgelinkt naar zijn startpagina (afhankelijk van de rechten van de gebruiker).	
Rationale:	Inloggen is noodzakelijk om ervoor te zorgen dat enkel participanten en facilitator toegang krijgen tot het platform. Dit is de belangrijkste manier om ongeauthoriseerde gebruikers te vermijden van het platform.	
Pre condities:	De gebruiker is nog niet ingelogd. De gebruiker heeft een loginnaam en wachtwoord ontvangen.	
Post condities:	De gebruiker heeft toegang verkregen tot het platform. Afhankelijk van de rechten van de gebruiker zal deze terecht komen bij een overzicht van beschikbare bevragingrondes (voor participanten) of bij de beheerssectie van het platform (voor facilitators)	
Main succes scenario		
#	Actor actie:	Platform reactie:
1	Gebruiker surft naar login pagina	
2		Het platform vraagt om loginnaam en wachtwoord.
3	Gebruiker voert loginnaam en wachtwoord in.	
4		Het platform controleert de loginnaam en het wachtwoord en verleent toegang tot het platform op basis van gebruikerstype (facilitator/participant).
Extensies (alternative flow)		
#4: Loginnaam/wachtwoord zijn incorrect, waardoor het platform de toegang weigert en de loginpagina opnieuw toont met een melding.		

#### 6.1.2. Selecteer een bevragingronde [Eis 2]

Prioriteit:	Must	
Doel:	De participant een overzicht geven van de beschikbare bevragingrondes waartoe dat deze rechten heeft gekregen (en waaraan kan worden deelgenomen), zodat deze een bevragingronde kan opstarten.	
Actors en interesse:	<i>Participant:</i> moet aan de hand van deze lijst kunnen kiezen aan welke bevraging hij gaat deelnemen.	
Omschrijving:	Nadat de participant is ingelogd, zal hij automatisch worden	

	doorgestuurd naar een pagina waarop een lijst wordt weergegeven met alle beschikbare bevraging rondes waartoe deze rechten heeft. De participant kan vervolgens een bevraging ronde kiezen, waarna deze bevraging ronde zal worden geladen in zijn browser.	
Rationale:	De participant moet kunnen kiezen (indien hij toegang heeft tot meerdere rondes) aan welke bevraging ronde hij gaat deelnemen.	
Pre condities:	Participant is ingelogd. Participant heeft minstens rechten op 1 bevraging ronde.	
Post condities:	De participant heeft een ronde geselecteerd, waarna deze zal worden opgestart.	
Main succes scenario		
#	Actor actie:	Platform reactie:
1	Participant vraagt een lijst met bevraging rondes aan	
2	Het platform geeft weer tot welke bevraging rondes de participant rechten heeft en welke bevraging rondes er actief zijn.	
3	Participant selecteert een ronde uit de lijst	
4	Het platform start de geselecteerde bevraging ronde op.	
Extensies (alternative flow)		
#2: Er zijn geen bevraging rondes waarin de participant kan deelnemen. Het platform geeft een boodschap terug om later nog eens te proberen.		

### 6.1.3. Antwoorden ingeven tijdens een bevraging ronde [Eis 3]

Prioriteit:	Must	
Doel:	De participant moet antwoorden kunnen geven op de vragen die worden gesteld in de bevraging ronde. Deze antwoorden kunnen zowel open vragen zijn (via een tekstveld) als ook gesloten vragen (ja/nee, bepaalde argumenten rangschikken,...).	
Actors en interesse:	<i>Participant:</i> de participant zijn bijdrage aan het besluitvormingsproces is om zijn mening en kennis vast te leggen door middel van het geven van antwoorden op de gestelde vragen.	
Omschrijving:	De pagina van de participant wordt geladen met de vragen van een (door de participant geselecteerde) bevraging ronde. De participant heeft de mogelijkheid om een vraag open te laten en om vragen in willekeurige volgorde te beantwoorden. Hierbij moet een overzicht worden bijgehouden met de vragen die zijn beantwoord en met dewelke nog open staan.	
Rationale:	Het ingeven van antwoorden op vragen uit een bevraging ronde is het belangrijkste onderdeel voor de participanten in het platform. Het stelt hen in staat om hun kennis en ideeën te formuleren in de voorziene ruimtes.	
Pre condities:	Participant heeft een bevraging ronde opgestart.	
Post condities:	De vragen op één of meerdere antwoorden zijn opgeslagen in het platform, zodat de facilitator in een later stadium hierop analyses kan uitvoeren en conclusies kan trekken.	
Main succes scenario		
#	Actor actie:	Platform reactie:
1	Participant selecteert een vraag die hij wenst te beantwoorden.	
2	De participant formuleert zijn antwoord of selecteert een optie uit een meerkeuzelijst.	
3	De participant slaat de gegevens op.	
4	Het platform slaat het antwoord van de participant op en geeft een bericht terug waarin staat of dat de actie succesvol was of niet.	
Extensies (alternative flow)		
3.a#: Het platform slaat de gegevens automatisch op na een vastgestelde interval. 3.b#: Het platform verstuurt de gegevens van een antwoord op het moment dat de participant begint met het antwoorden van een nieuwe vraag.		

### 6.1.4. Afsluiten van een bevragingronde [Eis 4]

Prioriteit:	Must
Doel:	De participant moet op het einde van bevragingronde een signaal kunnen geven waarop hij bevestigt dat hij klaar is met antwoorden.
Actors en interesse:	<i>Participant:</i> de participant moet zijn goedkeurig geven om de antwoorden als "finaal" te markeren. Hierdoor beëindigt hij de bevragingronde. <i>Facilitator:</i> De facilitator ziet in zijn overzicht wie reeds gedaan heeft met antwoorden van een bevragingronde
Omschrijving:	Op het einde van een bevragingronde moet de participant een signaal geven aan het platform om te laten weten dat hij klaar is. Op dit moment zal het platform de bevragingronde voor deze participant markeren als voltooid. Hierna zal de participant worden teruggebracht naar de startpagina, waarop de bevragingronde is verdwenen uit zijn lijst met beschikbare bevragingrondes.
Rationale:	Bij het afsluiten van een bevragingssessie bevestigt de participant dat hij klaar is met het beantwoorden van de vragen. Dit signaal moet worden gegeven omdat het mogelijk is dat de participant enkele vragen wenst open te laten. Bij processen met synchrone intermenselijke communicatie kan het ook zijn dat een participant zijn antwoord wenst te herzien nadat hij nieuwe informatie heeft verkregen. Door een signaal te versturen dat hij klaar is met het beantwoorden van de vragen, worden al deze vragen gemarkeerd als voltooid, waarna er geen wijzigingen meer kunnen worden aangebracht.
Pre condities:	Participant is klaar met het beantwoorden van de vragen uit een bevragingronde.
Post condities:	De participant is terug op zijn startpagina en de afgelegde bevragingronde is verdwenen uit zijn lijst met mogelijke bevragingrondes
Main succes scenario	
#	Actor actie: Platform reactie:
1	De participant geeft het signaal dat hij klaar is met het formuleren van zijn antwoorden.
2	Het platform markeert alle gegeven antwoorden in het platform als "eindantwoord" en sluit de bevragingronde voor de participant af.
3	Het platform herleidt de participant naar de startpagina, waarbij de afgelegde bevragingronde is verwijderd uit zijn lijst.
Extensies (alternative flow)	
/	

### 6.1.5. Autosaving van antwoorden [Eis 5]

Prioriteit:	Could
Doel:	De participant moet de mogelijkheid hebben om één of enkele vragen te beantwoorden en zijn sessie af te sluiten om op een later tijdstip verder te werken. Ook een onvoorziene crash van de webbrowser of een stroomuitval mag er niet voor zorgen dat de antwoorden van de participant die reeds werden gegeven verloren kunnen gaan.
Actors en interesse:	<i>Participant:</i> Voor de participant is het belangrijk dat zijn antwoorden zo snel mogelijk worden bijgehouden door het platform. Hierdoor wordt voorkomen dat deze geen (of niet veel) gegevens kan kwijtspelen bij een crash (stroompanne, internetconnectie die wegvalt, webbrowser die crasht,...). <i>Facilitator:</i> Indien in het besluitvormingsproces wordt beslist om de antwoorden van de participanten verschillende malen door te sturen, dan ontstaat er een soort van evolutie per antwoord per gebruiker. Hierop kan de facilitator analyses doen om te kijken in hoeverre dat bepaalde meningen veranderen over de ronde heen.

Omschrijving:	Via een vaste interval, of na elk gegeven antwoord, moeten de nieuwe gegevens worden verstuurd naar de server. Dit kan gebeuren op het moment dat de participant aan de volgende vraag begint, of op het moment dat de participant expliciet wenst op te slaan via een "antwoord opslaan" knop. De eigenlijke keuze (via vaste interval, via een knop, of andere) moet onderzocht worden op gebied van performantie/ gebruiksvriendelijkheid tijdens het testen van het prototype.
Rationale:	Tijdens het beantwoorden van een bevragingronde zullen verschillende antwoorden worden gegeven door de participant. Bij een stroomstoring of een crash van de webbrowser zouden echter alle gegevens verloren kunnen gaan. Om dit te voorkomen moet, per antwoord van de participant, de gegevens worden opgeslagen. Dit zorgt er ook voor dat de participant op elk gewenst ogenblik moet kunnen beslissen om zijn werk stop te zetten en uit te loggen. Wanneer hij op een later stadium terug in logt, moeten alle gegeven antwoorden automatisch terug worden ingevuld, zodat de participant rustig kan verder gaan met het antwoorden van de vragen, en indien gewenst, nog veranderingen kan doorvoeren op bestaande antwoorden.
Pre condities:	Één of meerdere vragen zijn beantwoord door de participant.
Post condities:	De webserver heeft de antwoorden opgeslagen.
Main succes scenario	
#	Actor actie: Platform reactie:
1	De participant beantwoordt 1 of meerdere vragen en slaat de gegevens op.
2	Het platform verwerkt de gegevens en geeft een boodschap of het opslaan van de gegevens succesvol is verlopen of niet.
Extensies (alternative flow)	
#1.a: De antwoorden van de participant worden via een vastgestelde interval verstuurd naar de webserver	
#1.b: De antwoorden worden automatisch verstuurd op het moment dat de participant met de volgende vraag begint	

### 6.1.6. Beheren van eindgebruikers [Eis 6]

Prioriteit:	Must
Doel:	De facilitator moet participanten en facilitators kunnen aanmaken.
Actors en interesse:	<i>Facilitator</i> : Moet gebruikers kunnen aanmaken, aanpassen en verwijderen uit het platform.
Omschrijving:	De facilitator moet in de beheerssectie de mogelijkheid hebben om gebruikers aan te maken, aan te passen, te verwijderen uit het platform.
Rationale:	Het platform moet naast een koppeling met het KnoSoS-systeem ook onafhankelijk kunnen werken. Hiervoor is het beheren van gebruikers noodzakelijk. Ook dient elke participant van een unieke login te worden voorzien, zodat bij het analyseren van de antwoorden achterhaald kan worden wie welke antwoorden heeft gegeven
Pre condities:	Er is minstens 1 facilitator in het systeem om andere gebruikers aan te maken
Post condities:	Er zijn minstens 2 eindgebruikers in het systeem
Main succes scenario	
#	Actor actie: Platform reactie:
1	De facilitator creëert een nieuwe eindgebruiker
2	Het platform vraagt om de gegevens van een gebruiker in te geven/ aan te passen (zie bij data requirements om welke gegevens het gaat).
Extensies (alternative flow)	
#1.a: De facilitator wenst een gebruiker aan te passen.	
#1.b.1: de facilitator wenst een gebruiker af te vegen	
1.b.2: het platform verwijdert de gebruiker.	

### 6.1.7. Toekennen van rechten aan eindgebruikers [Eis 7]

Prioriteit:	Must	
Doel:	Participanten toegang verlenen tot bepaalde bevraging rondes en participanten rechten geven op bepaalde besluitvormingsprocessen.	
Actors en interesse:	<i>Facilitator</i> : bepaald wie welke rechten heeft in het platform.	
Omschrijving:	De facilitator krijgt in de beheerssectie de mogelijkheid om bepaalde rechten toe te kennen aan andere facilitators en heeft de mogelijkheid om participanten toegang te verlenen tot bepaalde bevraging rondes	
Rationale:	Het toekennen van rechten aan gebruikers zorgt ervoor dat er restricties kunnen worden opgelegd van wie wat mag uitvoeren.	
Pre condities:	Facilitator heeft toegang tot de beheerssectie. 1 of meerdere eindgebruikers bestaan wel in het platform	
Post condities:	De eindgebruikers hebben toegang gekregen tot het platform en zijn hiervan op de hoogte gebracht.	
Main succes scenario		
#	Actor actie:	Platform reactie:
1	De facilitator selecteert de eindgebruiker(s) en geeft deze toegang tot een bepaald onderdeel van het platform.	
2		Het platform geeft de geselecteerde eindgebruikers de gewenste toegang.
3		Het platform brengt de eindgebruikers op de hoogte van hun nieuwe rechten (via mail)
Extensies (alternative flow)		
/		

### 6.1.8. Ondersteuning synchrone intermenselijke communicatie [Eis 8]

Prioriteit:	Must	
Doel:	De mogelijkheid bieden om een communicatiekanaal op te zetten tussen de facilitator en de participanten.	
Actors en interesse:	<i>Participant</i> : kan indien hij een vraag heeft contact opnemen met de facilitator  <i>Facilitator</i> : kan vragen van de participanten beantwoorden + kan een boodschap sturen naar alle participanten die zijn geconnecteerd.	
Omschrijving:	De synchrone (intermenselijke) communicatie bestaat uit verschillende informatiestromen. Zo moet het mogelijk zijn dat de facilitator een bericht stuurt naar alle geconnecteerde participanten (een soort van broadcast mechanisme). De participanten moeten zelf ook de mogelijkheid hebben om een vraag te sturen naar de facilitator.	
Rationale:	Synchrone intermenselijke communicatie biedt nieuwe mogelijkheden tijdens het uitvoeren van een bevraging ronde. Zo kan er ondermeer rechtstreekse support worden gegeven door de facilitator tijdens een bevraging ronde.	
Pre condities:	Er is minstens 1 participant geconnecteerd en de facilitator is ter beschikking. Een participant heeft een vraag/ een facilitator wenst een bericht te versturen.	
Post condities:	Eén of meerdere participanten hebben een bericht ontvangen/ de facilitator heeft een antwoord gegeven op een inkomende vraag	
Main succes scenario		
#	Actor actie:	Platform reactie:
1	De participant stelt een vraag aan de facilitator	
2		Het platform ontvangt de vraag en stuurt de vraag door naar de facilitator

3	De facilitator leest en beantwoordt de vraag en verstuurt het antwoord naar de participant	
4		Het platform ontvangt het antwoord en stuurt dit antwoord door naar de betreffende participant
5	De participant ontvangt het antwoord. Indien deze meer informatie wenst, herbegint hij bij stap n°1	
<b>Extensies (alternative flow)</b>		
#1: De facilitator wenst een bericht te sturen naar alle geconnecteerde participanten. #2: Het platform ontvangt het bericht en verstuurt dit bericht naar alle geconnecteerde participanten #3: De participanten ontvangen het bericht en kunnen, indien gewenst, een antwoord versturen.		

### 6.1.9. Beheren van een besluitvormingsproces [Eis 9]

Prioriteit:	Must	
Doel:	Een besluitvormingsproces bevat één of meerdere bevraging rondes. Het aanmaken, aanpassen en verwijderen van een besluitvormingsproces zorgt ervoor dat de facilitator een besluitvormingsproces kan beheren en waardoor een logische afbakening ontstaat tussen verschillende bevraging rondes.	
Actors en interesse:	<i>Facilitator</i> : de facilitator moet verschillende besluitvormingsprocessen kunnen beheren, zodat er een logische scheiding is tussen verschillende bevraging rondes.	
Omschrijving:	Op het moment dat de facilitator in de beheerssectie terecht komt, krijgt hij de mogelijkheid om een nieuw besluitvormingsproces op te starten, een bestaand besluitvormingsproces aan te passen of een besluitvormingsproces af te vegen.	
Rationale:	Het kunnen uitvoeren van meerdere besluitvormingsprocessen op hetzelfde ogenblik	
Pre condities:	De facilitator is ingelogd in de beheerssectie.	
Post condities:	De facilitator heeft een besluitvormingsproces aangepast, aangemaakt of afgeveegd.	
<b>Main succes scenario</b>		
#	Actor actie:	Platform reactie:
1	De facilitator creeërt een nieuw besluitvormingsproces	
2		Het platform vraagt om de gegevens van een besluitvormingsproces in te geven/ aan te passen (zie bij data requirements om welke gegevens het gaat).
3	De facilitator wenst het besluitvormingsproces op te slaan	
4		Het platform slaagt de gegevens van het besluitvormingsproces op.
<b>Extensies (alternative flow)</b>		
#1.a: De facilitator wenst een besluitvormingsproces aan te passen. #1.b.1: de facilitator wenst een besluitvormingsproces af te vegen 1.b.2: het platform verwijdert het besluitvormingsproces.		

### 6.1.10. Beheren van een bevragingronde [Eis 10]

Prioriteit:	Must	
Doel:	Een besluitvormingsproces bevat één of meerdere bevraging rondes. Het aanmaken, aanpassen en verwijderen van een bevragingronde zorgt ervoor dat de facilitator de vragen voor de participanten kan ordenen en structureren.	
Actors en interesse:	<i>Facilitator</i> : stelt de facilitator in staat om vragen te groeperen in logische blokken.	
Omschrijving:	Nadat een facilitator een besluitvormingsproces heeft geselecteerd in de beheerssectie, krijgt deze de mogelijkheid om een nieuwe	

	bevragsingsronde op te starten, om een bestaande bevragsingsronde aan te passen (enkel indien de ronde nog niet is gestart), of om een bevragsingsronde af te vegeen.	
Rationale:	Het onderverdelen van vragen in bevragsingsrondes zorgt voor een conceptuele onderverdeling. Het biedt tevens een afscheiding tussen verschillende discussies die gevoerd moeten worden.	
Pre condities:	De facilitator heeft een besluitvormingsproces geselecteerd waarin hij een bevragsingsronde wenst te beheren	
Post condities:	De facilitator heeft een bevragsingsronde aangepast, aangemaakt of afgeveegd.	
Main succes scenario		
#	Actor actie:	Platform reactie:
1	De facilitator creeert een nieuwe bevragsingsronde.	
2		Het platform vraagt om de gegevens van een bevragsingsronde in te geven/ aan te passen (zie bij data requirements om welke gegevens het gaat).
3	De facilitator wenst de bevragsingsronde op te slaan.	
4		Het platform slaagt de gegevens van de bevragsingsronde op.
Extensies (alternative flow)		
#1.a: De facilitator wenst een bevragsingsronde aan te passen.		
#1.b.1: de facilitator wenst een bevragsingsronde af te vegeen.		
1.b.2: het platform verwijdert de bevragsingsronde.		

#### 6.1.11. Importeren en exporteren van een besluitvormingsproces [Eis 11]

Prioriteit:	Could	
Doel:	De facilitator moet de mogelijkheid hebben om een afgerond besluitvormingsproces te exporteren naar een bestand. Dit bestand moet op een later tijdstip terug kunnen worden geïmporteerd in het platform.	
Actors en interesse:	<p><i>Facilitator:</i> heeft de mogelijkheid om een afgerond besluitvormingsproces te exporteren. Dit zorgt ervoor dat de gegevens veilig op een extra plaats kunnen worden bewaard. Tevens biedt dit de mogelijkheid om in de beheerssectie overbodige besluitvormingsprocessen te verwijderen, zodat de beheerssectie overzichtelijk blijft.</p> <p><i>Platformbeheerder:</i> Zorgt ervoor dat enkele de actieve data in het platform blijft, waardoor het opvragen van gegevens sneller kan verlopen. Het opslaan van een besluitvormingsproces in een apart bestand verhoogt tevens de eenvoud van het maken van een backup.</p>	
Omschrijving:	Nadat een besluitvormingsproces volledig is voltooid (nadat alle bevragsingsrondes en analyses zijn uitgevoerd en het besluitvormingsproces is afgerond) kan een besluitvormingsproces worden geëxporteerd vanuit de beheerssectie. Hierbij wordt een bestand aangeboden waarin alle informatie omtrent het besluitvormingsproces bevat zit: alle vragen en antwoorden, alle gebruikers en alle andere informatie die behoort tot dat besluitvormingsproces. (wat voor formaat dit bestand heeft, is nog onduidelijk)	
Rationale:	Om de beheerssectie van het platform overzichtelijk te houden kunnen besluitvormingsprocessen die zijn afgesloten worden geëxporteerd. Dit bestand kan op een later stadium terug worden geïmporteerd, zodat er nadien ook nog de mogelijkheid is om alle gegevens terug te gebruiken.	
Pre condities:	Er is een besluitvormingsproces succesvol afgerond	
Post condities:	De facilitator heeft een bestand op zijn PC staan waarin alle gerelateerde gegevens staan van één besluitvormingsproces.	
Main succes scenario		
#	Actor actie:	Platform reactie:
1	De facilitator selecteert het afgeronde besluitvormingsproces dat hij wenst te exporteren	



2		Het platform verzamelt alle gerelateerde gegevens en voegt deze samen tot 1 document. Dit document wordt ter download aangeboden aan de facilitator.
Extensies (alternative flow)		
#1: De facilitator wenst een besluitvormingsproces in het platform te importeren. #2: Het platform importeert alle gegevens met betrekking tot het besluitvormingsproces.		

### 6.1.12. Beheren van vragen [Eis 12]

Prioriteit:	Must	
Doel:	Een bevragingronde bevat meerdere vragen. Zo een vraag moet kunnen worden aangemaakt, aangepast en verwijderd worden uit het platform.	
Actors en interesse:	<i>Facilitator</i> : de facilitator kan vragen beheren om zo een bevragingronde samen te stellen.	
Omschrijving:	Op het moment dat een facilitator een bevragingronde heeft geselecteerd in de beheerssectie van het platform, krijgt deze de mogelijkheid om een nieuwe vraag toe te voegen, een vraag aan te passen of een vraag te verwijderen uit het platform. Dit beheren kan enkel wanneer de bevragingronde nog niet is gestart. Dit om te voorkomen dat mogelijke antwoorden die al zijn gegeven niet meer helemaal overeenstemmen met de nieuwe vraag. Bij het aanmaken van een nieuwe vraag moet er de mogelijkheid zijn om een grafiek te tonen met antwoorden van een vraag uit een vorige bevragingronde (maar wel uit hetzelfde besluitvormingsproces). Een voorbeeld hiervan is bij het Delphi proces, waarbij participanten in ronde 3 kunnen zien wat zij, en andere participanten, hebben geantwoord op een vraag uit ronde 2.	
Rationale:	Het beheren van de vragen is uiterst belangrijk en vormt ook één van de steunpilaren van het platform. Het dynamisch kunnen opstellen van de vragen, waarbij er de mogelijkheid is om grafieken uit vorige rondes te integreren, is wat dit platform uniek en krachtig maakt.	
Pre condities:	De facilitator heeft een bevragingronde geselecteerd waarin hij vragen wenst te beheren.	
Post condities:	De facilitator heeft een vraag toegevoegd, aangepast of verwijderd.	
Main succes scenario		
#	Actor actie:	Platform reactie:
1	De facilitator creëert een nieuwe vraag.	
2		Het platform vraagt om een nieuwe vraag samen te stellen (zie bij data requirements om welke gegevens het gaat).
3	De facilitator wenst de vraag op te slaan.	
4		Het platform slaagt de vraag op.
Extensies (alternative flow)		
#1.a: De facilitator wenst een vraag aan te passen. #1.b.1: de facilitator wenst een vraag af te vege 1.b.2: het platform verwijderd de vraag.		

### 6.1.13. Bekijken van geconnecteerde participanten [Eis 13]

Prioriteit:	Should	
Doel:	De facilitator een lijst aan te bieden met geconnecteerde participanten en hun status (hoeveel vragen er zijn beantwoord, laatste activiteit van participant,...). Bij de niet geconnecteerde participanten wordt hun status weergegeven (nog niet begonnen, gestopt tijdens beantwoorden van bevragingronde, bevragingronde afgerond)	
Actors en interesse:	<i>Facilitator</i> : krijgt een overzicht van de (geconnecteerde) participanten met hun status. Deze lijst biedt ook de mogelijkheid om contact op te nemen met een geconnecteerde participant.	

Omschrijving:	Nadat de facilitator een bevragingronde heeft geselecteerd in de beheerssectie, krijgt deze te zien welke participanten zijn geconnecteerd en welke niet. Hierbij kan hij, indien gewenst, contact opnemen met de participant.	
Rationale:	De facilitator moet een overzicht hebben van alle geconnecteerde participanten indien deze contact met hem wenst op te nemen. Tevens kan hij het verloop van een bevragingronde volgen. Zeker bij een synchroon besluitvormingsproces kan het handig zijn voor de participant om te volgen welke participant verbonden is en wat hun laatste activiteit was. Bij de niet geconnecteerde participanten wordt hun status weergegeven.	
Pre condities:	De facilitator heeft een bevragingronde geselecteerd in de beheerssectie. De bevragingronde is actief en er is minstens 1 participant die rechten heeft op de bevragingronde.	
Post condities:	De facilitator heeft kennis van wie er geconnecteerd is en van de status van de niet geconnecteerde participanten.	
Main succes scenario		
#	Actor actie:	Platform reactie:
1	De facilitator vraagt een lijst aan met geconnecteerde participanten en de statussen van de niet-geconnecteerde participanten	
2		Het platform geeft een lijst terug met geconnecteerde gebruikers met hun laatste activiteit en hun progressie in de bevragingronde. Voor de niet-geconnecteerde participanten wordt hun status weergegeven.
Extensies (alternative flow)		
/		

#### 6.1.14. Exporteren van bevragingronde voor analyse [Eis 14]

Prioriteit:	Should	
Doel:	De facilitator de mogelijkheid geven om een afgeronde bevragingronde te laten exporteren naar een formaat dat gebruikt kan worden om analyses uit te voeren. In de eerste plaats zal dit formaat CSV zijn, maar nadien kan er beslist worden om dit uit te breiden of te vervangen door een ander gangbaar formaat.	
Actors en interesse:	<i>Facilitator</i> : heeft deze export nodig om analyses te kunnen uitvoeren van een bepaalde bevragingronde.	
Omschrijving:	De facilitator selecteert in de beheerssectie een (afgeronde) bevragingronde. Deze bevragingronde wordt geëxporteerd naar een formaat dat geïmporteerd kan worden in bijvoorbeeld excel.	
Rationale:	Het platform kan in een later stadium ondersteuning bieden voor analyses, maar in de eerste plaats zal er een export functie voorzien zijn zodat analyse van antwoorden kan worden uitgevoerd in verschillende programma's. Hierdoor kan de facilitator zijn besluitvormingsproces verder zetten in de door hem geliefde omgeving (excel, een business intelligence omgeving, of een statistisch programma) om zo verbanden te zoeken tussen antwoorden en/of correlaties zoeken tussen participanten hun profiel en participanten hun antwoorden.	
Pre condities:	De bevragingronde is beëindigd en gesloten. De facilitator heeft in de de beheerssectie een besluitvormingsproces geselecteerd waaruit hij een bevragingronde wenst te exporteren.	
Post condities:	De facilitator ontvangt van het platform een bestand dat kan worden geïmporteerd.	
Main succes scenario		
#	Actor actie:	Platform reactie:
1	De facilitator selecteert de bevragingronde die hij wenst te exporteren.	
2	De facilitator selecteert het formaat waarin de export moet plaatsvinden	

3		Het platform bouwt een bestand op in het geselecteerde formaat
Extensies (alternative flow)		
/		

### 6.1.15. Tonen van progressiebalk [Eis 15]

Prioriteit:	Could	
Doel:	Het platform toont een overzicht van het aantal afgelegde vragen en het aantal vragen die nog niet zijn beantwoord om de participant zijn progressie te tonen in het beantwoorden van zijn vragenlijst.	
Actors en interesse:	<i>Participant:</i> Wilt een overzicht hebben van de progressie van zijn geleverd werk.	
Omschrijving:	Om de participant een idee te geven van het werk dat hij/zij reeds heeft verricht, moet deze op de hoogte worden gehouden van de progressie van zijn/haar werk. Hiervoor moet er gebruik worden gemaakt van een progressiebalk, of een gelijkaardig mechanisme, dat de participant in staat stelt om te kunnen zien hoeveel werk deze nog voor de boeg heeft. Deze aanduiding kan een overzicht zijn van het gegeven aantal vragen (X van de Y vragen zijn beantwoord), kan procentueel worden uitgedrukt (% van de antwoorden is ingevuld), of er kan een schatting worden gemaakt van de resterende tijd (X minuten resterend) aan de hand van de ingegeven antwoorden en de gemiddelde tijd die andere participanten nodig hadden om te antwoorden op de resterende vragen.	
Rationale:	Participanten moet weten hoeveel werk ze al hebben verricht, en hoeveel werk ze nog te doen hebben om een bepaalde bevragingronde af te werken.	
Pre condities:	Participant heeft een bevragingronde gestart.	
Post condities:	De participant ziet een progressiebalk waarop aangegeven staat hoeveel werk er nog moet worden verricht.	
Main succes scenario		
#	Actor actie:	Platform reactie:
1	De participant beantwoordt een vraag.	
2		Het platform past het overzicht aan waarin de progressie van de participant wordt weergegeven.
Extensies (alternative flow)		
/		

### 6.1.16. Beheren van profielpagina [Eis 16]

Prioriteit:	Should	
Doel:	De participant de mogelijkheid geven om bepaalde data over hem/haar te beheren in een profiel	
Actors en interesse:	<i>Participant:</i> wenst zijn informatie accuraat en up to date te houden. De overige participanten hebben mogelijk interessen om dit profiel op te vragen.	
Omschrijving:	De participant krijgt de mogelijkheid om zijn/haar profiel in te vullen of aan te passen. Hij/zij bepaalt (in beperkte mate) welke gegevens zichtbaar mogen worden gesteld door bepaalde velden leeg te laten.	
Rationale:	De participant wilt dat de gegevens die over hem/haar openbaar worden gemaakt correct zijn.	
Pre condities:	De participant is ingelogd in het systeem	
Post condities:	De participant heeft zijn/haar profiel aangemaakt of aangepast	
Main succes scenario		
#	Actor actie:	Platform reactie:
1	De participant bezoekt zijn/haar profiel	
2		Het platform geeft het bestaande profiel terug met de mogelijkheid om de bestaande waarden te wijzigen.

3	De participant verandert 1 of meerdere gegevens aan het profiel en slaat de gegevens op.
4	Het platform toont het gewijzigde profiel.
Extensies (alternative flow)	
#2.a: Op het moment dat nog geen profiel aanwezig is, zal het platform een leeg profiel voorleggen aan de participant.	

### 6.1.17. Bekijken van profielenpagina [Eis 17]

Prioriteit:	Should
Doel:	De participanten de mogelijkheid geven om de profielen van de andere participanten op te vragen.
Actors en interesse:	<i>Participant:</i> de participant wenst informatie te hebben over zijn collega's in het besluitvormingsproces. Deze informatie kan gaan van opleidingen en ervaringen tot geografische en persoonlijke data.
Omschrijving:	De participant krijgt de mogelijkheid om de profielen van de andere participanten te bekijken. Welke gegevens deze profielen bevat wordt gespecificeerd in "Beheren van profielenpagina"
Rationale:	Wanneer participanten de mogelijkheid hebben om de profielen van andere participanten op te vragen, kan dit het groepsgevoel van de participanten verhogen. Dit groepsgevoel is iets dat vaak ontbreekt bij GDSS systemen. Via het bekijken van de profielen van de participanten kan toch een samenhangingsgevoel worden gekweekt.
Pre condities:	De participant is ingelogd in het systeem
Post condities:	De participant heeft één of meerdere profielen bezocht
Main succes scenario	
#	Actor actie: Platform reactie:
1	De participant vraagt een overzicht van alle profielen.
2	Het platform geeft een overzicht van alle beschikbare profielen (die de participant mag zien).
3	De participant selecteert een profiel dat hij/zij wenst te bekijken.
4	Het platform geeft het profiel weer.
Extensies (alternative flow)	
/	

### 6.1.18. Beschikbaar stellen van gegevens voor externe systemen [Eis 18]

Prioriteit:	Must
Doel:	Gegevens beschikbaar stellen voor externe systemen die gegevens over een besluitvormingsproces wensen te gebruiken
Actors en interesse:	<i>Extern systeem:</i> Externe systemen wensen mogelijk gebruik te maken van gegevens die worden beheerd in het platform. Deze gegevens kunnen ze gebruiken voor diverse doeleinden.
Omschrijving:	Een extern systeem dat gegevens wenst op te halen kan, na authenticatie, bepaalde gegevens ophalen uit het platform.
Rationale:	Externe systemen die gegevens wensen op te vragen over een besluitvormingsproces moeten via een gecentraliseerde systeem gegevens kunnen opvragen over een bepaald besluitvormingsproces.
Pre condities:	Het externe systeem heeft een authenticatiegegevens ontvangen en is in staat om de gegevens die worden teruggestuurd te verwerken.
Post condities:	Het externe systeem is in het bezit van de gewenste gegevens
Main succes scenario	
#	Actor actie: Platform reactie:
1	Het externe systeem doet een aanvraag voor bepaalde gegevens en geeft de authenticatiegegevens mee.
2	Het platform authenticatieert de aanvraag en zoekt de gevraagde gegevens op. De gegevens worden aan het externe systeem

	aanbieden.
Extensies (alternative flow)	
#2.a: De authenticatie faalt. Het platform geeft een melding terug dat de authenticatie niet gelukt is.	

### 6.1.19. Indelen van nieuwe informatie in categoriën [Eis 19]

Prioriteit:	Must
Doel:	De participant de mogelijkheid geven om nieuwe gegevens (die in een synchroon besluitvormingsproces binnen komen) in te delen in categoriën
Actors en interesse:	<i>Participant:</i> De participant wenst de nieuwe informatie overzichtelijk te houden. Hiervoor kan deze een beroep doen op een aantal voorgelegde categoriën of kan deze een aantal categoriën zelf definiëren. Hierdoor kan deze de argumenten van de andere participanten indelen in een aantal categoriën
Omschrijving:	Wanneer de participant (in een synchroon besluitvormingsproces) de argumenten en ideeën ontvangt van andere participanten, moet deze de mogelijkheid hebben om deze ideeën in te delen in een aantal categoriën. Het moet ook mogelijk zijn (indien het besluitvormingsproces dit zo voorschrijft) dat de participant zelf zijn categoriën kan opstellen.
Rationale:	Het overzichtelijk houden van de binnenkomende informatie is een cruciale factor in het kunnen beheersen van de complexiteit voor het nemen van besluiten. Indien vele argumenten binnenkomen, zal de participant geen overzicht meer kunnen behouden van al deze argumenten, en ontstaat er een informatie-overload. Het categoriseren kan hierop een oplossing bieden.
Pre condities:	De participant zit in een synchroon besluitvormingsproces waarbij dat nieuwe informatie aangeleverd gaat worden
Post condities:	De participant heeft de aangeleverde informatie onderverdeeld in een categorie
Main succes scenario	
#	Actor actie: Platform reactie:
1	De participant merkt dat nieuwe informatie beschikbaar is voor een bepaalde vraag
2	Het platform toont het nieuwe argument en geeft een aantal categoriën waaruit de participant kan kiezen.
3	De participant selecteert 1 of meerdere categoriën waaraan hij het argument wenst toe te kennen
4	Het platform kent het argument toe aan de categorie.
Extensies (alternative flow)	
#3a.De participant maakt een nieuwe categorie aan, waarna hij het argument toe kan kennen aan de nieuwe categorie.	

## 6.2. Niet functionele eisen

### 6.2.1. Nieuwe groepsbesluitvormingsprocessen aanmaken [Eis NF1]

Prioriteit:	Must
Rationale:	Ontwikkelaars moeten aanpassingen kunnen maken aan groepsbesluitvormingsprocessen en nieuwe besluitvormingsprocessen kunnen ontwikkelen. Het platform moet flexibel genoeg zijn om al deze processen te kunnen ondersteunen.
Actor:	Programmeur
Stimulus:	Wenst een besluitvormingsproces te programmeren dat opgenomen kan worden in het platform
Artifact:	Broncode van het platform en de groepsbesluitvormingsprocessen.

Omgeving:	/
Platform reactie:	/
Reactie meeting:	Een programmeur met minstens 3 jaar programmeerervaring (die de syntax kent van .NET) moet in staat zijn om een proces aan te maken in minder dan 2 volledige weken (inclusief testen)

### 6.2.2. Bestaande groepsbesluitvormingsprocessen aanpassen [Eis NF2]

Prioriteit:	Must
Rationale:	Ontwikkelaars moeten aanpassingen kunnen maken aan groepsbesluitvormingsprocessen en nieuwe processen kunnen ontwikkelen. Het platform moet flexibel genoeg zijn om al deze processen te kunnen ondersteunen.
Actor:	Programmeur
Stimulus:	Wenst een kleine aanpassing te maken in een besluitvormingsproces dat reeds is aangemaakt en getest.
Artifact:	Broncode van het platform en de groepsbesluitvormingsprocessen
Omgeving:	/
Platform reactie:	/
Reactie meeting:	Een programmeur met minstens 3 jaar programmeerervaring (die de syntax kent van .NET) moet in staat zijn om simpele aanpassingen aan een proces aan te brengen in minder dan 1 werkdag. (Onder kleine aanpassingen wordt verstaan: In plaats van een standaard een 5 puntsschaal te gebruiken in een vraag, standaard een 7 puntsschaal gebruiken, etc.)

### 6.2.3. Initiële laadtijd van pagina [Eis NF3]

Prioriteit:	Should
Rationale:	De laadtijd van de pagina moet worden beperkt tot 20 seconden. Na deze laadtijd, moet de participant kunnen beginnen met het antwoorden van vragen. Data die nog niet is geladen moet op de achtergrond worden geladen.
Actor:	Participant
Stimulus:	Wenst onmiddellijk te beginnen met het geven van antwoorden
Artifact:	Het platform, onder normale werkdruk
Omgeving:	Het bevragsingsgedeelte van het platform
Platform reactie:	Het laden van de bevragsingsronde is (deels) geladen
Reactie meeting:	Na 20 seconden moet de participant in staat zijn om te beginnen met antwoorden op de vragen. De data die nog niet (geheel) is geladen, moet in de achtergrond worden geladen.

### 6.2.4. Gevoel van realtime intermenselijke communicatie [Eis NF4]

Prioriteit:	Should
Rationale:	Participanten moeten het gevoel hebben dat de communicatie met de facilitator realtime verloopt, zonder een (aanzienlijke) vertraging.
Actor:	Participant
Stimulus:	Wenst onmiddellijk antwoorden te ontvangen van de facilitator.
Artifact:	Het platform, onder normale en zware werkdruk.
Omgeving:	Het communicatiekanaal tussen participant en facilitator.
Platform	Versturen van berichten tussen facilitator en participanten.

reactie:	
Reactie meeting:	Het platform mag een maximale vertraging hebben van <b>x</b> seconden (nader te onderzoeken aan de hand van een enquête die wordt afgenomen bij participanten na het werken met een prototype)

## 6.3. Bedrijfsdoelen

### 6.3.1. Ondersteuning bieden voor onderzoek naar groepsbesluitvorming

De 2 betrokken onderzoekscentra (VUB en Memori) die werken aan het KnoSoS-systeem werken ook rond kennismanagement. Beide partijen willen het nut van groepsbesluitvormingsondersteunende systemen onderzoeken bij het structureren van kennisdeling. Hiervoor willen zij een platform (laten) ontwikkelen waarin ze voldoende vrijheid hebben om de ondersteunende processen zelf, naar behoeven, samen te stellen. Het platform moet ook de mogelijkheid bieden om nieuwe ideeën en theoretische stelling omtrent groepsbesluitvormingsondersteunende processen te modelleren.

Hiernaast zal het platform ook toelaten om de kwaliteit van deze besluiten te vergelijken met die van face-to-face beslissingen. Daarom is het belangrijk dat het platform aanpasbaar is om nieuwe processen te kunnen aanmaken en om huidige processen te kunnen aanpassen.

### 6.3.2. Onderzoek naar synchrone intermenselijke communicatie

In tegenstelling tot het vorige asynchrone systeem laat het GDSS platform toe besluitvormingsprocessen synchroon op te zetten tussen de eindgebruikers (tussen participanten en de facilitator en tussen participanten onderling).

Dit brengt echter nieuwe problemen met zich mee:

- Hoe wordt deze nieuwe informatie zichtbaar gemaakt,
- Hoe gaat de participant om met deze informatie-overload,
- Wat is de performantie van deze communicatie.

Al deze vragen zullen, met behulp van het platform verder onderzocht kunnen worden. Hierdoor wordt het belang van flexibiliteit nogmaals benadrukt. Het kunnen aanpassen en toevoegen van functionaliteit is belangrijk om het onderzoek te kunnen ondersteunen.

### 6.3.3. Onderzoek naar synchrone data transfer

AJAX is een verzameling van technieken (Javascript, XML, CSS, DOM) die het mogelijk maakt om informatie op een webpagina asynchroon te versturen naar de webserver.[AjaxInAction, FoundationsOfAjax]. Hierdoor is het mogelijk om gegevens uit te wisselen tussen de webpagina van de eindgebruikers en het platform zonder dat hiervoor de hele webpagina bij de eindgebruiker opnieuw moet worden herladen (zoals in traditionele webpagina's).

Deze vorm van werken biedt enkele nieuwe opportuniteiten:

- Het versturen van antwoorden van de participant naar de webserver zonder dat een webpagina hiervoor moet worden ververst.
- Het tonen van antwoorden van andere participanten zonder een aanzienlijke vertraging.
- Een hogere vorm van interactie tussen de participanten en de facilitator en participanten onderling, waardoor het besluitvormingsproces een stuk natuurlijker verloopt.

Om te testen of deze opportuniteiten ook daadwerkelijk kunnen worden gebruikt in een besluitvormingproces, moet het GDSS platform de mogelijkheid bieden om AJAX te verwerken.

#### **6.3.4. Het groepsbesluitvormingsproces ondersteunen**

##### **Gestructureerd ideeën genereren**

Het platform moet in staat zijn om ondersteuning te bieden aan participanten bij idee-generatie processen (zoals bijvoorbeeld brainstorming,...). De idee-generatie (divergerende) processen die worden beschreven in [VanGundy], zullen door het platform (in zekere mate) moeten worden ondersteund.

##### **Gestructureerd besluiten kunnen nemen**

Vervolgens is het belangrijk dat een groep, op basis van de gegenereerde ideeën, de juiste ideeën verder kan uitwerken en de juiste beslissingen kan maken. Ook hiervoor zijn verschillende convergerende processen ontwikkeld en beschreven(zie [VanGundy]). Een correcte ondersteuning van deze processen bepaalt in grote mate het succes van het platform.

#### **6.3.5. Ruimte bieden voor uitbreidingen en aanpassingen**

Flexibiliteit is belangrijk in het GDSS-platform: het platform moet de mogelijkheid bieden om nieuwe processen toe te voegen en bestaande processen aan te passen. Hierbij is het belangrijk dat bij aanpassingen in een proces er geen of nauwelijks veranderingen nodig zijn bij andere processen die aanwezig zijn in het platform. Het platform moet dus breed worden opgezet en ondersteuning geven voor generieke besluitondersteunende activiteiten.

#### **6.3.6. Koppeling met KnoSoS-systeem**

Het platform zal een koppeling moeten hebben met het KnoSoS-systeem. Gebruikers in het KnoSoS-systeem moeten kunnen deelnemen als eindgebruikers in het GDSS platform.



## Bijlage 7: Het potentieel van AJAX

### 7.1. Resultaten van de vergelijking van data-overdracht technieken

	Afhankelijkheid	Snelheid verwerking	Bandbreedte
XML	++	-- / -	-
JSON	0 / +	+	- / 0
Javascript functies	--	++	+

#### 7.1.1. Afhankelijkheid

Deze eigenschap geeft aan hoe dat het gebruik van een bepaalde techniek een koppeling veroorzaakt met een webpagina : hoe losser een koppeling, hoe beter. XML bijvoorbeeld scoort erg goed, omdat hetzelfde XML bestand eventueel door een ander systeem kan worden verwerkt. Bij het gebruik van Javascript functie aanroepen is de koppeling erg sterk: enkel webpagina's die over de Javascriptfuncties beschikken zijn in staat om de functie aanroepen correct uit te voeren. Bij het wijzigen van de Javascript functie interface zal bijgevolg ook een verandering op de serverkant moeten plaats vinden, waardoor de afhankelijkheid van de code op de server erg groot is met die van de client. [AjaxInAction]

JSON schommelt tussen deze twee uitersten in. Alhoewel JSON een Javascript Object representeert, bestaan voor praktisch alle programmeertalen conversiebibliotheken die bijvoorbeeld een PHP klasse kunnen omzetten van en naar het JSON formaat. Deze bibliotheken zorgen ervoor dat de afhankelijkheid niet slecht is. Wel zorgt dit voor een extra afhankelijkheid van externe bibliotheken om de conversie te maken. [AjaxInAction]

#### Snelheid verwerking

Omdat Javascript functies direct uitgevoerd kunnen worden, hebben deze een minimale overhead. Het inkomende bericht moet met de Javascript "eval" functie aangeroepen worden om de code daadwerkelijk uit te voeren. Hierdoor is de snelheid van verwerking in de browser het snelste via de Javascript functie aanroepen.

XML is het andere uiterste omdat het een hoop overhead bevat. Het is opgemaakt uit veel tags die geparsed moeten worden door de browser, via de W3C DOM navigatie- en manipulatiefuncties. Dit parsen is een traag proces dat enige tijd in beslag neemt. [AjaxInAction]

JSON bevindt zich weer tussen deze twee uitersten in. Aangezien JSON, via de "eval" functie van Javascript, omgezet kan worden naar een Javascript Object, is het verwerken van deze data snel. Deze data kan daarna meegegeven worden als parameter(s) aan bepaalde Javascript functies.

#### Bandbreedte

Het verschil in verbruik van bandbreedte kan aardig oplopen indien voor een bepaalde techniek wordt gekozen. Stel dat een functie geschreven wordt die de gegevens van een gebruiker toont. Indien gewerkt wordt met "Javascript functie aanroepen" techniek, dan kan de webserver bijvoorbeeld antwoorden met de volgende response:

**Server antwoordt met een directe Javascript functie aanroep**

```
return "toonGebruikerGegevens('Jan','Amsterdam');"
```

In de browser zal, overeenstemmend met de functieaanroep, een functie gedefinieerd zijn die deze gegevens verwerkt.

#### Browser verwerkt het antwoord van de server

```
function toonGebruikerGegevens(naam, woonplaats)
{
    alert(naam + " woont in " + woonplaats);
}
```

Hieruit blijkt dat de benodigde data (en dus de benodigde bandbreedte) minimaal is wanneer gewerkt wordt met directe Javascript functie aanroepen. Wanneer echter met XML wordt gewerkt als data uitwisselingsformaat, dan valt de overhead van XML meteen op.

#### Server antwoordt met XML data

```
return " <persoon>
        <voornaam>Jan</voornaam>
        <woonplaats>Amsterdam</woonplaats>
    </persoon>";
```

#### Browser verwerkt het antwoord van de server

```
function laadGebruikerXMLGegevens(xmlstring)
{
    var xmlDoc = new ActiveXObject("Microsoft.XMLDOM");
    xmlDoc.loadXML(xmlstring);
    //navigeer in de XML string naar de juiste locatie om de
    gegevens eruit te filteren
    var
    voornaam=xmlDoc.getElementsByTagName("voornaam")[0].nodeValue;
    var woonplaats=
    xmlDoc.getElementsByTagName("woonplaats")[0].nodeValue;
    //herbruik de functie uit het vorige voorbeeld
    toonGebruikerGegevens(voornaam, woonplaats);
}
```

Omdat de opmaak van XML veel extra tekens bevat (de tags), neemt het volume van een bericht snel toe, en dus ook de benodigde bandbreedte.

Het JSON formaat bevindt zich weer tussen deze uitersten in. Het JSON formaat heeft een gelijkaardige datastructuur als XML. In plaats van de tags die worden gebruikt in XML, wordt bij JSON gebruik gemaakt van accolades om de data in te delen.

#### Server antwoordt met JSON data

```
//ge-escaped response
return "    {\"persoon\":
        {
            \"voornaam\" : \"Jan\",
            \"woonplaats\" : \"Amsterdam\"
        }
    }";
```

#### Browser verwerkt het antwoord van de server

```
function laadGebruikerJSONGegevens(jsonstring)
{
    var gebruiker = eval("(" + jsonstring + ")");
    toonGebruikerGegevens(gebruiker.voornaam, gebruiker.woonplaats);
}
```

Alhoewel JSON in dit geval dus een kortere notatie heeft als XML, is het niet altijd zo dat JSON minder tekens in beslag neemt om een bepaald object te beschrijven. [JSON Wikipedia]. Indien voor XML veel gebruik wordt gemaakt van attributen in plaats van innernodes (in dit voorbeeld zou dat het volgende resultaat geven : <persoon voornaam="Jan" woonplaats="Amsterdam" /> ), dan kan het zijn dat de XML notatie even kort of korter is als de JSON notatie. Dit omdat attributen, in tegenstelling tot nodes, geen afsluitende tag nodig hebben.

Er kan dus geconcludeerd worden dat de structuur en de opmaak van het object (aantal attributen, aantal inner objecten) dat verstuurd moet worden tussen de server en de client bepaalt welke techniek het minste bandbreedte verbruikt.

## **7.2. Analyseren van AJAX frameworks**

### **7.2.1. Waarom een AJAX framework gebruiken**

Omdat sommige functionaliteit verschillend zijn geïmplementeerd van browser tot browser, is het soms erg lastig om functies te schrijven die worden ondersteund op (bijna) alle gangbare browsers. Om een consistente interface te creëren, zodat de programmeur geen rekening meer moet houden met welke browser de eindgebruiker de webapplicatie benaderd, zijn een aantal AJAX frameworks ontwikkeld. De meeste van deze frameworks trachten ook terugkomende functionaliteit te verzamelen. De programmeur kan hier dan rechtstreeks gebruik van maken, en kan zo voorkomen dat het spreekwoordelijke wiel niet nog eens opnieuw moet worden uitgevonden [AjaxInAction, FoundationsOfAjax].

Nog voor de opkomst van AJAX werden Javascript frameworks aangeboden. Deze spitste zich meestal toe op het aanbieden van nieuwe invoerelementen (boomstructuur, kalender) [FoundationsOfAjax]. Met de opkomst van AJAX zijn echter verschillende frameworks ontwikkeld, gaande van kleine frameworks, die enkel de XMLHttpRequest versimpelen, tot de uitgebreide frameworks, die nieuwe controls, XMLHttpRequest functies, ondersteuning voor webservices en DOM manipulatie functies aanbieden [FoundationsOfAjax, AjaxInAction]

Om een gefundeerde keuze te maken uit deze grote verzameling van frameworks, is het belangrijk om de criteria voor de selectie goed vast te stellen

### **7.2.2. Opstellen van criteria voor selectie van een AJAX framework**

Het GDSS platform zal naast de asynchrone besluitvormingsprocessen ook ondersteuning bieden voor synchrone processen, waarbij gebruikers onderling, realtime, berichten kunnen versturen. Hiernaast moeten de antwoorden van de participanten kunnen worden verstuurd zonder dat de hele webpagina hiervoor moet worden ververs. Deze functies lenen zich goed voor gebruik te maken van de XMLHttpRequest functie, om zo asynchroon gegevens naar en van de webserver te versturen.

In het GDSS platform is flexibiliteit een grote eis: het moet mogelijk zijn om bestaande besluitvormingsprocessen aan te passen, of om geheel nieuwe processen aan te maken. Om de gebruiksvriendelijkheid voor de participanten te verhogen, zouden nieuwe invoerelementen beschikbaar moeten worden gemaakt in het te kiezen AJAX framework. Met nieuwe invoerelementen wordt hier in eerste instantie gedacht aan :

- Een kalender control
- Een rich textbox control
- Drag and drop support

Voor deze drie invoerelementen kan, op dit moment, functionaliteit worden bedacht waar dat nieuwe invoerelementen een meerwaarde zouden kunnen vormen. Een kalender control kan de participant in staat stellen om op een eenvoudige manier een datum te selecteren en een

rich textbox control stelt deze in staat om zijn antwoorden te benadrukken met opmaak (het vet, cursief of onderlijnen van tekst, het gebruik van opsommingstekens, etc.). De drag and drop support is een feature die gebruikt zou kunnen worden om de participanten bepaalde criteria te laten sorteren. Hierbij kan deze dan op een intuïtieve manier de argumenten naar boven of naar onder slepen.

Naast dit aanbod van controls ligt de prioriteit ook op flexibiliteit van het AJAX framework. Indien bijvoorbeeld het hele AJAX framework bestaat uit een 20 controls, en in een bepaald besluitvormingsproces worden slechts 4 van die controls gebruikt, dan moet het mogelijk zijn om de overige controls niet te laten downloaden door de client, om zo de laadtijden te verkorten.

De criteria voor selectie van een AJAX framework zijn dus in dit geval:

- Het aanbieden van een aantal nieuwe invoerelementen, met een voorkeur voor de volgende controls / features:
  - Een kalender control
  - Een rich textbox control
  - Drag and drop support
- Mogelijkheid tot het configureren van welke nieuwe invoerelementen geladen moet worden
- Het beschikbaar stellen van de XMLHttpRequest via een eenvoudige interface

Naast deze functionele criteria wordt ook gekeken naar hoe dat het framework wordt aangeboden. Deze criteria zijn

- Maturiteit : in hoeverre is het framework volwassen en stabiel
- Documentatie : wordt er veel/ goede documentatie aangeboden
- Extra features : welke extra (unieke) features worden aangeboden die potentieel interessant zijn

### 7.2.3. Analyseren van de frameworks

	Dojo	Open Rico	Atlas
XMLHttpRequest	V	V	V
Configureren van het laden van controls	V		V
Kalender control	V		
Rich Textbox control	V		
Drag and Drop support	V	V	V
Maturiteit / Ontstaan	2004 -	2005 +	2006 ?
Documentatie	--	--	+
Extra features	-Javascript collections -Event system -Backbutton support	-Cinematic effects -Livegrid control	-ModalPopup control - Always Visible control
Nadelen	Groot in omvang		Enkel ondersteuning voor ASP.NET

Ondersteuning voor het XMLHttpRequest object is in de onderzochte frameworks overal aanwezig. De werking is echter van framework tot framework anders: Bij Dojo en Open Rico wordt de programmeur in staat gesteld om de response van de server af te handelen. Bij Atlas wordt alle code automatisch gegenereerd, waardoor de programmeur enkel de Atlas-enabled control moet toevoegen aan zijn webformulier tijdens het ontwikkelen van de webpagina.

Het Dojo framework laat toe om Javascript libraries te laden in de browser van de eindgebruiker door middel van het definiëren van afhankelijkheden. Hierdoor zal de webpagina zelf kunnen beslissen, op het moment dat de pagina geladen wordt, welke code geladen moet worden, en welke niet. Deze ondersteuning werd niet terug gevonden bij Open Rico, waar alle code in één Javascriptbestand wordt gedownload naar de browser van de eindgebruiker toe. Atlas bepaalt zelf welke Javascript deze nodig heeft voor het uitvoeren van de webpagina, waardoor dus ook geen overbodige Javascripten geladen zullen worden.

Het Dojo framework geeft standaard ondersteuning voor de drie controls waarvan het GDSS platform gebruik kan maken. Open Rico en Atlas ondersteunen beide drag and drop, maar bieden geen van de andere controls aan. (Op de website van Open Rico [Open Rico Home] staat ook duidelijk vermeld dat het ontwikkelen van de kalender control buiten de scope van het project valt. Bij de Atlas documentatie was geen roadmap te vinden over de toekomstige features)

Het beoordelen van de maturiteit van de frameworks was een tijdsgebonden meting: wegens de beperkte doorlooptijd van het onderzoek konden slechts één of twee releases worden onderzocht. Hoewel één of twee bepaalde releases weinig kunnen zeggen over de gehele stabiliteit van een framework, geeft het wel een indicatie van hoe de stabiliteit van het framework kan evolueren. Zo gaf de release van Dojo 0.3.0 verschillende problemen bij het laden van elementen, terwijl de voorgaande versie hier duidelijk geen problemen mee had (0.2.2). In het Dojo framework zijn bij nieuwe releases ook een aantal ingrijpende wijzigingen gebeurd bij de functie aanroepen, waardoor deze niet altijd backwards compatible zijn.

Het Open Rico framework lijkt minder wijzigingen door te voeren aan de interface van functies, waardoor het voor de programmeur eenvoudiger is om van versie te upgraden. Omdat het Atlas framework nog maar één update onderging na de initiële release (versie 1.0.60504.0) is het echter onmogelijk om te oordelen over de stabiliteit tussen de releases in.

De documentatie van zowel het Dojo framework en het Open Rico framework laat veel te wensen over. De documentatie van het Dojo systeem is soms helemaal afwezig of is niet up to date. Het Rico framework bevat geen (officiële) documentatie. Hierbij moeten de programmeurs hun kennis opdoen door bestaande problemen en voorbeelden te analyseren op het forum van Rico. Het Atlas framework heeft daarentegen een uitgebreide documentatie ter beschikking: per aangeleverd invoerelement is een voorbeeld ter beschikking met voorbeeld code. De officiële Atlas website [Atlas Home] bevat ook uitgebreide "walkthroughs" waarop stap voor stap wordt uitgelegd hoedat een bepaald invoerelement kan worden uitgebreid met nieuwe functionaliteit.

Elk van de onderzochte frameworks bevat ook een gamma aan unieke functionaliteit, gaande van een uniek invoerselement, tot uitbreidingen van de Javascript taal.

## **7.3. Integreren van AJAX in GDSS Platform**

### **7.3.1. Opslaan en uitwisselen van antwoorden van participanten tijdens een bevragingronde**

Het verwerken van de antwoorden van de participant verliep in het vorige systeem (de Delphi versie) via een complete postback, waarbij de pagina helemaal opnieuw geladen moest worden. Bij de .NET herimplementatie werd gebruik gemaakt van verborgen frames om de data door te sturen naar de server, AJAX avant la lettre.

Deze functionaliteit was uiteraard bij uitstek geschikt om te testen in het prototype met behulp van AJAX. Het alternatief, werken via verborgen frames, was erg omslachtig, en vergde veel aandacht van de programmeurs om ervoor te zorgen dat alle code browseronafhankelijk werkte.

Voor de eindgebruiker was hier in principe geen verschil merkbaar ten opzichte van de .NET herimplementatie: de functionaliteit bleef hetzelfde. Voor de ontwikkelaars is de AJAX manier wel een stuk overzichtelijker en een stuk beter onderhoudbaar (veel transparantere code). Ten opzichte van de Delphi versie, die een volledige postback deed naar de webserver, betekent het asynschron versturen van de antwoorden van de participanten wel een enorme stap voorwaarts. De participanten kunnen immers bij elke vraag eenvoudig hun argumenten opslaan zonder dat de pagina opnieuw geladen hoeft te worden.

### **7.3.2. Communiceren via een tekstbased kanaal**

Voor het ontstaan van AJAX was het praktisch onmogelijk om een realtime communicatiekanaal op te zetten zonder gebruik te maken van Java of Flash. Echte alternatieven zijn er dus niet voor AJAX, indien men enkel met standaard aanwezige webtechnologieën wenst te werken. Het opzetten van een communicatiekanaal is een mooi voorbeeld van de nieuwe functionaliteit die AJAX met zich meebrengt in het GDSS platform.

Het "realtime" communiceren wordt hier aangeduid met dubbele aanhalingstekens, omdat in feite de browser om een bepaalde interval controleert op de webserver of nieuwe chatberichten beschikbaar zijn. Het is namelijk onmogelijk om van de webserver een connectie op te zetten met een browser. Het kan dus voorkomen dat de gebruiker pas na enkele seconden een chatbericht ontvangt.

Voor de eindgebruiker zal deze vertraging niet of slechts nauwelijks opvallen. Deze weet immers niet (exact) op welk tijdstip het bericht is verstuurd. Een mogelijke uitbreiding van dit communicatiekanaal kan zijn dat de frequentie van het controleren op nieuwe berichten variabel is, en afhangt van de tijdstippen van de vorige ontvangen berichten. Hierdoor zou de browser dus op frequentere basis controleren op nieuwe berichten indien de participant een conversatie aan het voeren is. Indien de gebruiker geen conversatie aan het voeren is, kan de interval worden vergroot.

### **7.3.3. Opstellen van een bevragsingsronde**


De facilitator moet voor het begin van een bevragsingsronde de mogelijkheid hebben om de bevragsingsronde op te zetten en te configureren. In vergelijking met de Delphi versie is dit opstellen een stuk complexer. De Delphi versie had immers een vaste structuur waar niet van afgeweken kon worden. Bij het GDSS platform wordt de facilitator echter in staat gesteld om het besluitvormingsproces zelf te sturen.

De facilitator moet de vrijheid hebben om vragen, van verschillende types (meerkeuze, open vragen, het sorteren van waarden) voor te leggen aan de participanten. Ook de manier waarop dat de participanten onderling de informatie verspreiden moet kunnen worden ingesteld: indien voor een synchroon proces wordt gekozen, moet de facilitator in staat zijn om de vragen zo te configureren dat de antwoorden van de participanten onderling gedistribueerd kunnen worden. Indien een bevragsingsronde wordt geanalyseerd, en uit deze analyse een nieuwe bevragsingsronde wordt opgesteld, moet de facilitator in staat zijn om de antwoorden van de participanten samen te voegen. Op het moment dat de facilitator beslist om op basis van een bepaald antwoord een nieuwe vraag te formuleren, dan wordt deze nieuwe vraag toegevoegd in de nieuwe bevragsingsronde, en kan de facilitator deze nieuwe vraag weer verder configureren (type vraag instellen, etc).

In een traditioneel systeem zou bij het opstellen van zo een besluitvormingsproces tussen elke nieuwe vraag een volledige postback gemaakt worden naar de webserver, waarna de hele webpagina opnieuw geladen moet worden. Het gebruik van AJAX laat hier echter toe om elke opgestelde vraag asynchroon door te sturen naar de webserver, zodat de hele bevragsingsronde gevormd kan worden, zonder dat de facilitator zijn webpagina ververst moet worden.

Dit systeem zorgt wel voor een langere ontwikkeltijd in vergelijking met een traditioneel systeem, aangezien een redelijk grote aantal Javascript functies geschreven moet worden om alle gegevens asynchroon te kunnen verwerken.

## Bijlage 8: Screenshots prototype

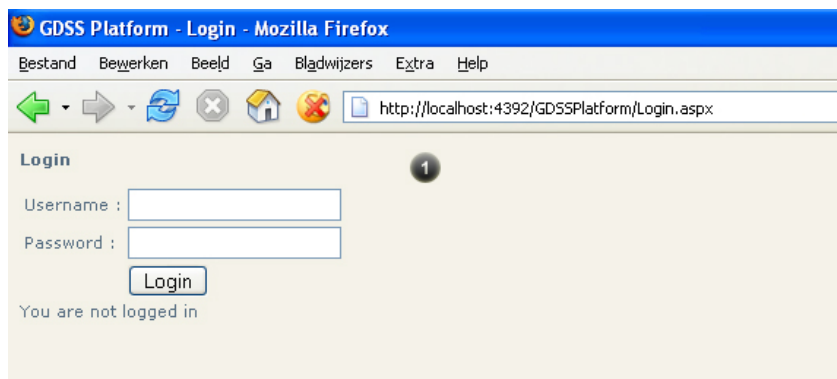
In deze bijlage worden enkele screenshots getoond van het prototype. Het icoon  duidt met het getal een activiteit aan die de gebruiker uit kan voeren in het platform.

### 8.1. Participant

De participant kan onderstaande acties uitvoeren.

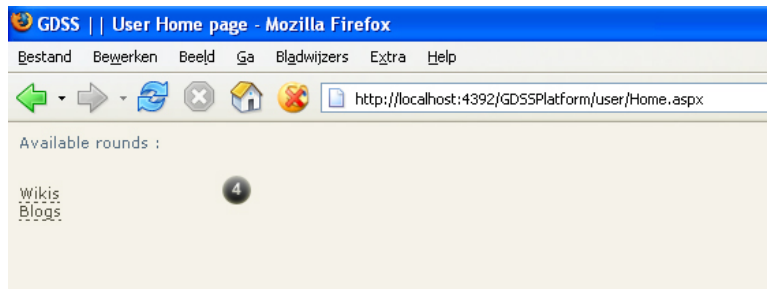
1. De participant begeeft zich naar de website waar het prototype draait.
2. De participant moet inloggen om toegang te krijgen tot het platform.
3. Het platform geeft al de bevraging rondes waartoe dat de participant rechten heeft om deel te nemen.
4. De participant selecteert een bevraging ronde.
5. Het platform toont de bevraging ronde en stuurt deze naar de participant.
6. De participant heeft de mogelijkheid om zijn/haar antwoorden toe te voegen/aan te passen in de bevraging ronde. Hij/zij kan elk antwoord apart opslaan zonder dat de webpagina moet worden verversd. De participant kan op de hoogte worden gebracht van gebeurtenissen (nieuwe antwoorden van participanten die hij/zij moet analyseren, nieuw bericht van de facilitator,...). Hiernaast heeft de participant de mogelijkheid om contact op te nemen met de facilitator om uitleg te vragen.
7. De participant is klaar met het invullen van de bevraging ronde en beëindigt de bevraging ronde door expliciet kenbaar te maken dat deze de ronde wenst te sluiten.
8. Het systeem herleidt de participant naar een “dank u voor deze participatie” pagina.

#### Stap 1: Inloggen in het systeem



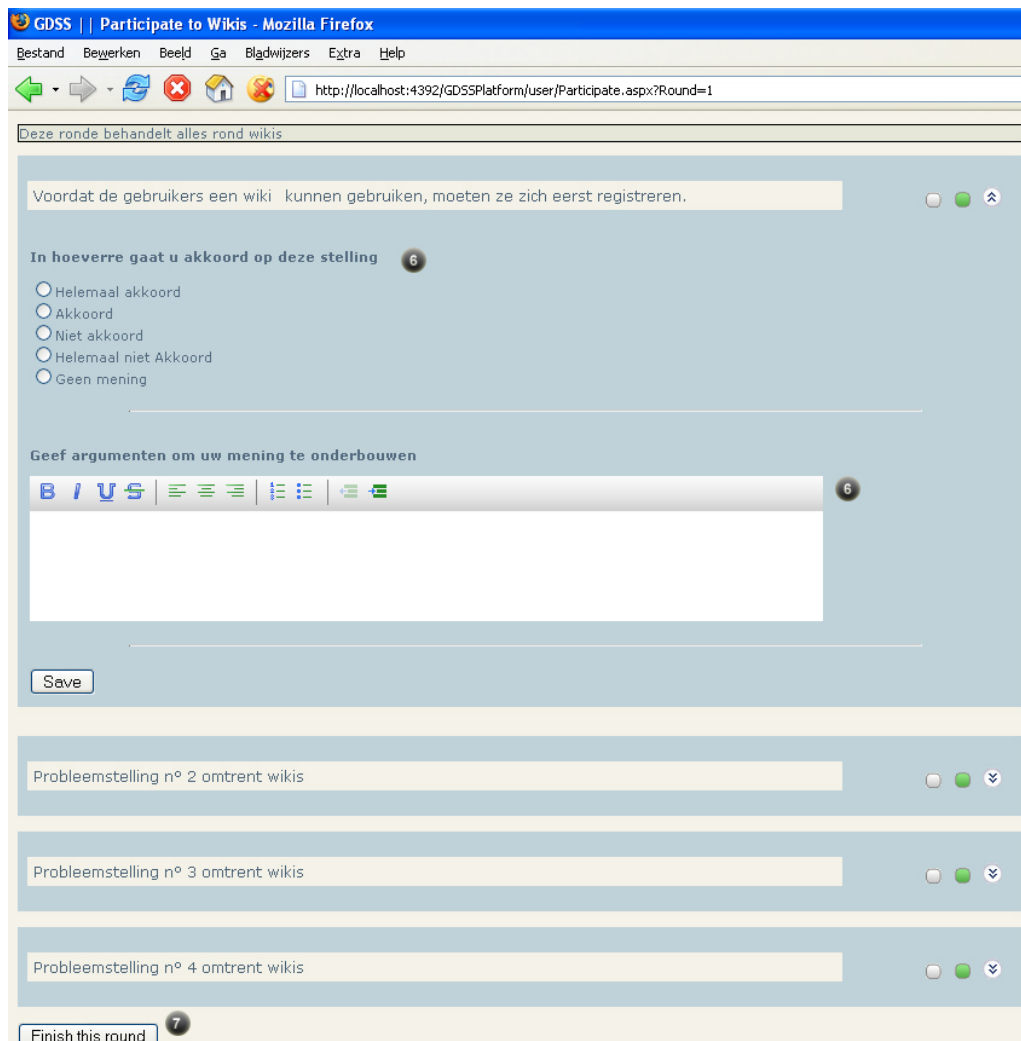
Wanneer een gebruiker een pagina bezoekt maar nog niet is ingelogd, zal deze naar de login pagina worden omgeleid. Na het inloggen wordt de gebruiker naar zijn gewenste pagina doorgeleerd (standaard is dit de “Home pagina”).

#### Stap 3: Selecteer een bevraging ronde



De participant heeft in dit voorbeeld rechten op twee verschillende bevestigingsrondes. De eerste noemt “Wikis” en de andere “Blogs”.

### Stap 6: Ingeven van antwoorden



Dit is de pagina waarop de participant zijn antwoorden kan geven. Eén afgebakende blok bevat één probleem, en één probleem bestaat uit meerdere vragen (zie hoofdstuk 8.3.1 “Views” – Facilitator / Overkoepelende view). De probleemdefinitie wordt voorgelegd als de titel van het afgebakende blok. Bij de eerste probleemdefinitie is dit “Voordat de gebruikers een wiki kunnen gebruiken, moeten ze zich eerst registreren”. De overige probleemstellingen zijn in dit geval dummygegevens.

Naast de probleemdefinitie zijn 3 iconen op te merken. De eerste twee iconen geven de status weer van het probleem, het 3<sup>de</sup> icoon stelt de participant in staat om de afgebakende box dicht te schuiven of terug te openen. Het eerste icoon geeft aan of de participant

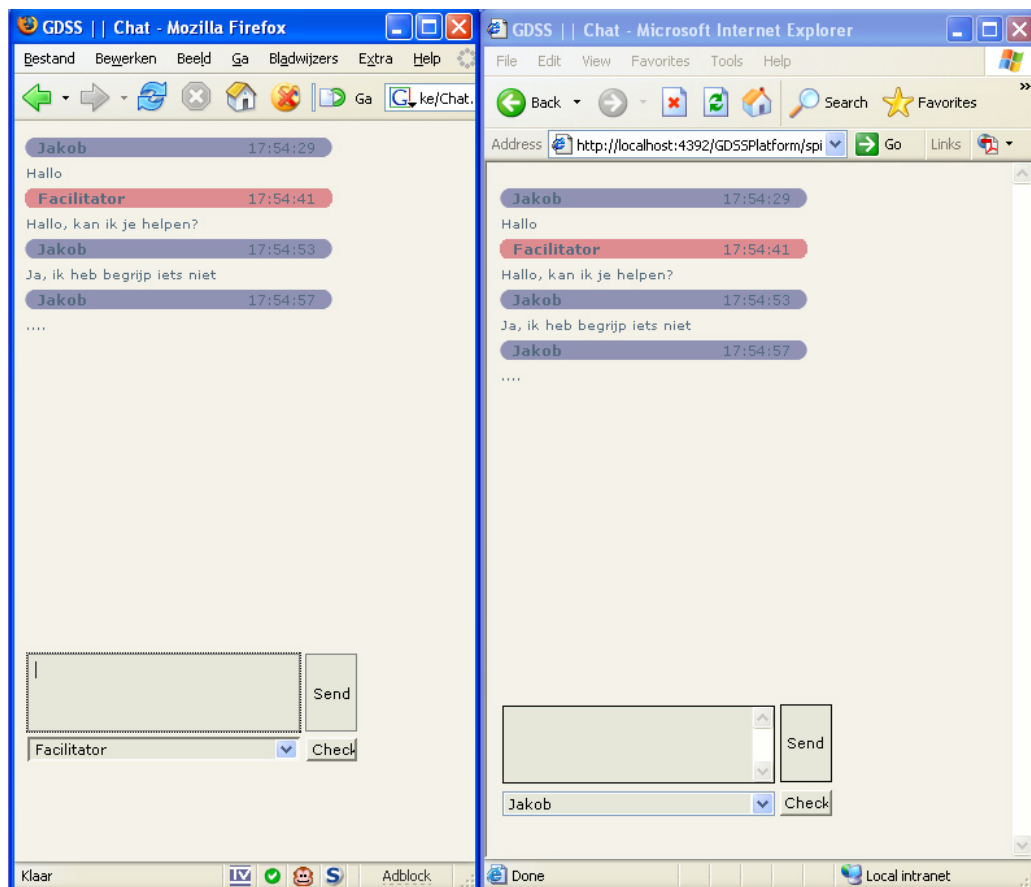


wijzigingen heeft aangebracht in zijn antwoord(en) voor dat probleem. Het tweede icoon geeft weer of er nieuwe informatie beschikbaar is van de andere participanten over dat probleem (enkel van toepassing bij synchrone besluitvormingsprocessen). Een overzicht van de mogelijke statussen van een probleem wordt weergegeven in de volgende tabel:

	Wijzigingen in antwoorden	Nieuwe informatie beschikbaar
Wit	Er zijn nog geen antwoorden ingegeven of opgeslagen door de participant.	/
Groen	De antwoorden zijn opgeslagen in de databank en zijn nadien niet meer gewijzigd door de participant.	Alle informatie is verwerkt/bekeken.
Orange	De antwoorden zijn al een keer opgeslagen, maar zijn nadien nog veranderd. De antwoorden zijn dus nog een keer opgeslagen worden.	/
Rood	Er zijn antwoorden ingegeven, maar deze zijn nog niet opgeslagen.	Er is nieuwe informatie beschikbaar voor dit probleem.

Door middel van deze statusiconen is de participant in staat om een overzicht te houden van welke problemen hij reeds heeft beantwoord en welke niet. Tevens kan hij/zij bij synchrone processen een overzicht houden van op welke problemen extra informatie beschikbaar is gekomen.

De participant kan ook contact op nemen met de facilitator indien hij/zij een vraag wenst te stellen. De volgende figuur laat zien hoe dat in zijn werk gaat. Op het linkse gedeelte zien we het scherm van de participant ("Jakob") die uitleg wenst. Het rechtse gedeelte is het gedeelte dat de facilitator ziet ("Facilitator").

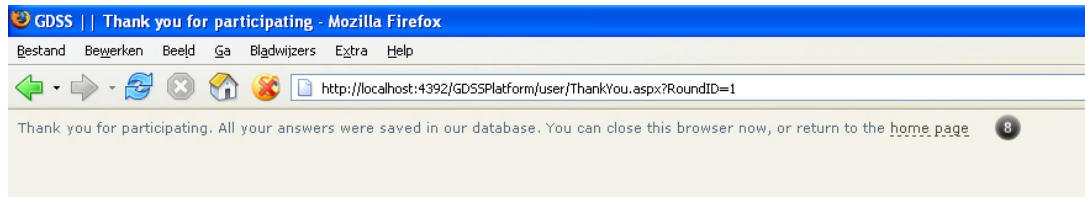


## Stap 7: Afronden van bevragsronde

Zie eerste figuur uit Stap 6. Wanneer de participant op deze knop drukt (en een bevestiging geeft), meldt hij/zij dat hij/zij klaar is met het beantwoorden van de vragen. Hierna kan hij/zij geen wijzigingen meer aanbrengen in deze bevragsronde. Ook kan de facilitator zien dat deze participant klaar is met het beantwoorden van de bevragsronde.

## Stap 8: Dank u voor het participeren.

Hierbij wordt de participant bedankt voor het deelnemen aan de bevragsronde. Hij/zij krijgt de mogelijkheid om terug naar het overzicht van de beschikbare bevragsrondes te gaan. De bevragsronde die hij/zij net heeft afgerond zal niet meer in de lijst voorkomen.



## 8.2. Facilitator

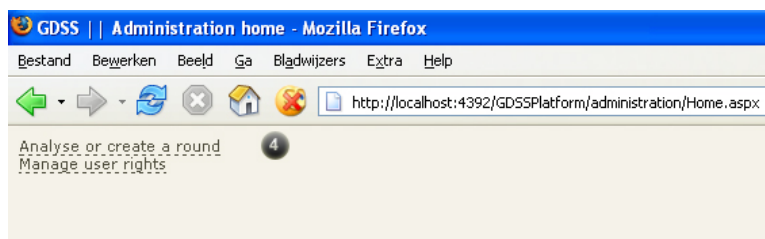
De volgende stappen bevatten de acties die een facilitator kan uitvoeren:

1. De facilitator begeeft zich naar de website waar het prototype draait.
2. De facilitator moet inloggen om toegang te krijgen tot het platform.
3. De facilitator krijgt een overzicht van de mogelijke acties die hij kan uitvoeren als facilitator.
4. De facilitator kiest om een nieuwe bevragsronde op te zetten.
5. De webpagina wordt ingedeeld in twee delen: in het linkse deel moet de facilitator aangeven welke bevragsronde hij wenst te beheren. Hierbij krijgt deze de mogelijkheid om nieuwe problemen en vragen te definiëren. In het rechter gedeelte kunnen, indien de bevragsronde verder gaat op een bestaande bevragsronde, al de antwoorden van de participanten geladen worden. Hierdoor is de facilitator in staat om uit de bestaande argumenten en ideeën nieuwe problemen te definiëren en voor te leggen aan de participanten
6. Op het moment dat de facilitator klaar is met het beheren van een bevragsronde zal deze doorgestuurd worden naar een pagina waar de rechten van de participanten kunnen worden ingesteld.

### Stap 1: Inloggen in het systeem

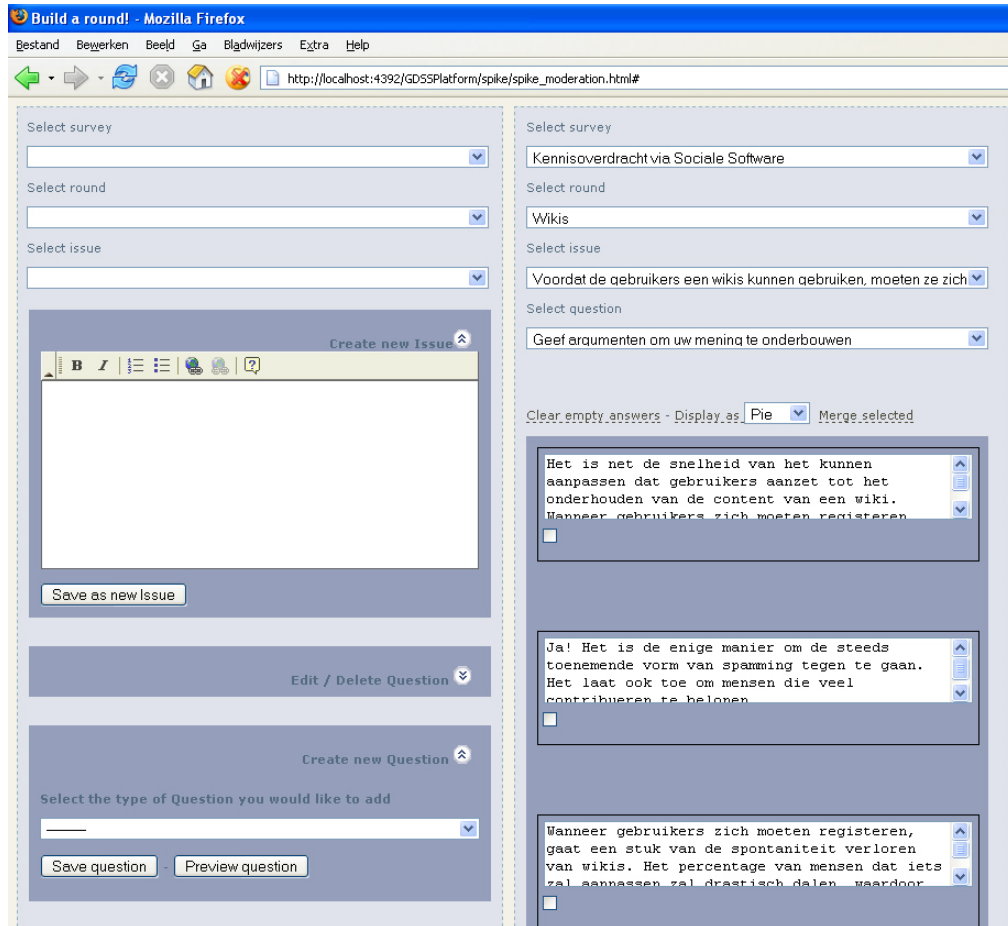
Deze procedure verloopt hetzelfde als voor de participant.

### Stap 3: De facilitator krijgt een overzicht van de mogelijke acties die hij kan uitvoeren



## Stap 5: Analyseren van een bestaande bevestigingsronde + opzetten van nieuwe bevestigingsronde

In dit voorbeeld worden bestaande antwoorden die zijn gegeven in de "Wiki" bevestigingsronde geanalyseerd. De gegeven antwoorden kunnen door middel van de onderstaande checkboxes worden samengevoegd tot 1 nieuwe probleemstelling.



Indien de vraag uit een multiplechoice vraag bestond, kunnen de antwoorden ook worden gerepresenteerd als een grafiek. De manier hoe dat de grafiek wordt gerepresenteerd kan worden geselecteerd in de dropdownbox "Display as ...". Onderstaande figuur geeft een voorbeeld van een taartdiagram van de antwoorden op de vraag "In hoeverre gaat u akkoord op deze stelling" van het probleem "Voordat gebruikers een wiki kunnen gebruiken moeten ze zich eerst registreren".

Build a round! - Mozilla Firefox

Bestand Bewerken Beeld Ga Bladwijzers Extra Help

http://localhost:4392/GD55Platform/spike/spike\_moderation.html#

Select survey  
Kennisoeverdracht via Sociale Software

Select round  
Wikis - deel 2

Select issue  
[Empty]

Create new Issue ^

**B I** [List Icon] [Link Icon] [Help Icon]

Save as new Issue

Edit / Delete Question v

Create new Question ^

Select the type of Question you would like to add

[Dropdown]

Save question Preview question

Select survey  
Kennisoeverdracht via Sociale Software

Select round  
Wikis

Select issue  
Voordat de gebruikers een wikis kunnen gebruiken, moeten ze zich

Select question  
In hoeverre gaat u akkoord op deze stelling

Clear empty answers - Display as: Pie Merge selected

**In hoeverre gaat u akkoord op deze stelling**

	Helemaal akkoord
	Akkoord
	Niet akkoord
	Helemaal niet Akkoord
	Geen mening

Nadat de facilitator een nieuw probleem heeft geformuleerd kan deze hier 1 of meerdere vragen voor aanmaken. De onderstaande figuur is een voorbeeld van het opzetten van een vraag waarop dat de gebruiker exact 1 antwoord kan selecteren ("single option" vraag).

Create new Question 

Select the type of Question you would like to add

singleoption 


Select the subtype of the Question you would like to add

radio 

Select the type of synchronisation you want to apply on this answers

Immediately distribute answers 


How should the question be displayed to the user?

horizontal 

What is the title of the question?

In hoeverre gaat u akkoord op deze stelling?

How should the input from the user be validated?

At least 1 value must be selected 

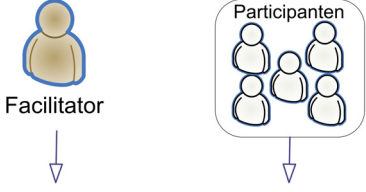
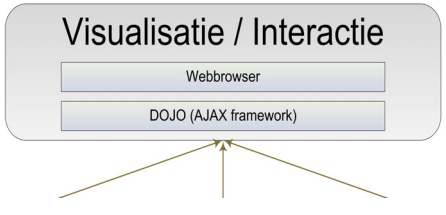
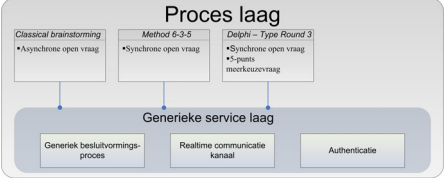
Add options

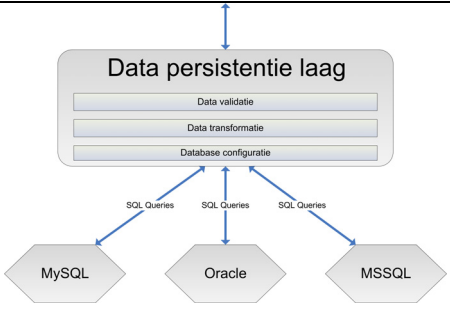
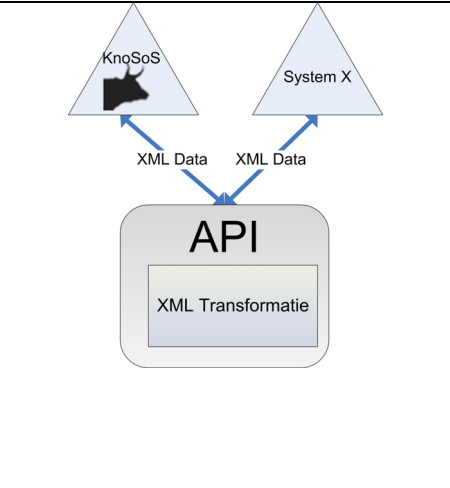
Display to User	Internal Value	
<input type="text"/>	<input type="text"/>	<input type="button" value="Add"/>
Helemaal akkoord	2	<a href="#">Edit</a> - <a href="#">Delete</a>
Akkoord	1	<a href="#">Edit</a> - <a href="#">Delete</a>
Neutraal	0	<a href="#">Edit</a> - <a href="#">Delete</a>
Niet akkoord	-1	<a href="#">Edit</a> - <a href="#">Delete</a>
Helemaal niet akkoord	-2	<a href="#">Edit</a> - <a href="#">Delete</a>

-

## Bijlage 9: Toelichting ontwikkeling view

De ontwikkelingsview wordt hier verder toegelicht. Per onderdeel van de view wordt een korte uitleg gegeven over de functionaliteit die het onderdeel vervult. Hierbij wordt de functionaliteit ook gekoppeld met de opgestelde eisen, zodat het duidelijk is hoe dat deze eisen in de architectuur terug komen.

	<p>De eindgebruikers van het systeem (facilitator en participanten) interageren met de “Visualisatie en interactielaag”. Deze interactie is zeer uiteenlopend. Voor de participanten zijn dit de acties die in de eisen 1, 2, 3, 4, 16 en 17 worden beschreven. Voor de facilitator zijn dit eisen 1, 6, 7, 9, 10, 11, 12, 13 en 14.</p>
	<p>De “Visualisatie en interactielaag” bepaalt de layout en interactiemogelijkheden voor de eindgebruikers. De eindgebruikers interageren met de webbrowser. De webbrowser maakt gebruik van het “DOJO framework” (=AJAX framework) om de gegevens uit te wisselen met het besluitvormingsproces. (Architectuurbeslissing “AJAX om interactiviteit te verhogen”)</p> <p>De bruine pijlen aan de onderkant wijzen erop dat de “Visualisatie en interactie” wordt aangestuurd door de onderliggende bevragingronde. Deze bevragingronde bepaalt in sterke mate hoe dat de interactie zal plaats vinden.</p> <p>De eisen die kunnen worden gekoppeld aan de “Visualisatie en interactielaag” zijn eisen 5, 15 en Eis Niet-Functioneel 3.</p>
	<p>Deze laag bevat de “Generieke service” laag en de verschillende besluitvormingsprocessen.</p> <p>De “Generieke service” laag bevat al de wederkerige eigenschappen die aan besluitvormingsprocessen kunnen worden toegeschreven. Het “Generiek besluitvormingsproces” (vervult architectuurbeslissing “Ondersteunen van meerdere besluitvormingsprocessen”), “Authenticatie” (vervult eis 1) en “Realtime communicatiekanaal” (vervult eis 8 en Eis Niet-Functioneel 4) zijn in de “Generieke service” laag terug te vinden.</p> <p>De besluitvormingsprocessen (“Classical Brainstorming”, “Method 6-3-5” en “Delphi – Type Round 3”) maken gebruik van de functionaliteit die de “Generieke service laag” aanbiedt. Hierbij representeert elk proces een specifieke implementatie van een besluitvormingsproces.</p> <p>Door het loskoppelen van de eigenschappen in een aparte service laag en verschillende specifieke implementaties, kan de flexibiliteit worden bekomen</p>

	<p>zoals die is beschreven in Niet-Functionele Eis 1 en de Niet-Functionele-Eis 2 (en daardoor ook Bedrijfsdoel 5 “Ruimte bieden voor uitbreidingen en aanpassingen” vervult).</p>
	<p>De “Data persistentie laag” zorgt voor de data opslag van het platform. De pijl aan de bovenkant wijst er op dat de “Proces laag” (zie vorige punt) de gegevens kunnen opvragen en bewerken via de “Data persistentie laag”.</p> <p>Het onderdeel “Data validatie” staat in om alle inkomende gegevens te valideren voor dat ze worden opgeslagen in de database.</p> <p>Het “Data transformatie” onderdeel vormt alle inkomende gegevens om naar database-onafhankelijke objecten.</p> <p>De “Database configuratie” bepaalt at runtime welke database wordt gebruikt. Deze 2 onderdelen zijn nodig voor architectuurbeslissing “Ondersteuning van meerdere databases” mogelijk te maken.</p>
	<p>De “API” zorgt ervoor dat gegevens van het platform met externe systemen, waaronder het KnoSoS systeem, kunnen worden uitgewisseld.</p> <p>De “XML Transformatie” module zorgt ervoor dat de gegevens uit de “Data Persistentie laag” worden omgevormd tot een XML structuur.</p> <p>De API maakt gebruik van de “Authenticatie” module uit de “Generieke proceslaag”.</p> <p>De “API” vervult Eis 18 en rechtstreeks ook Bedrijfsdoel 6 “Koppeling met KnoSoS platform”. Ook de architectuurbeslissing “API als vorm van data-uitwisseling met externe systemen” wordt hiermee vervult.</p>

## Bijlage 10: Validatie architectuur

### 10.1. Valideren van de software architectuur

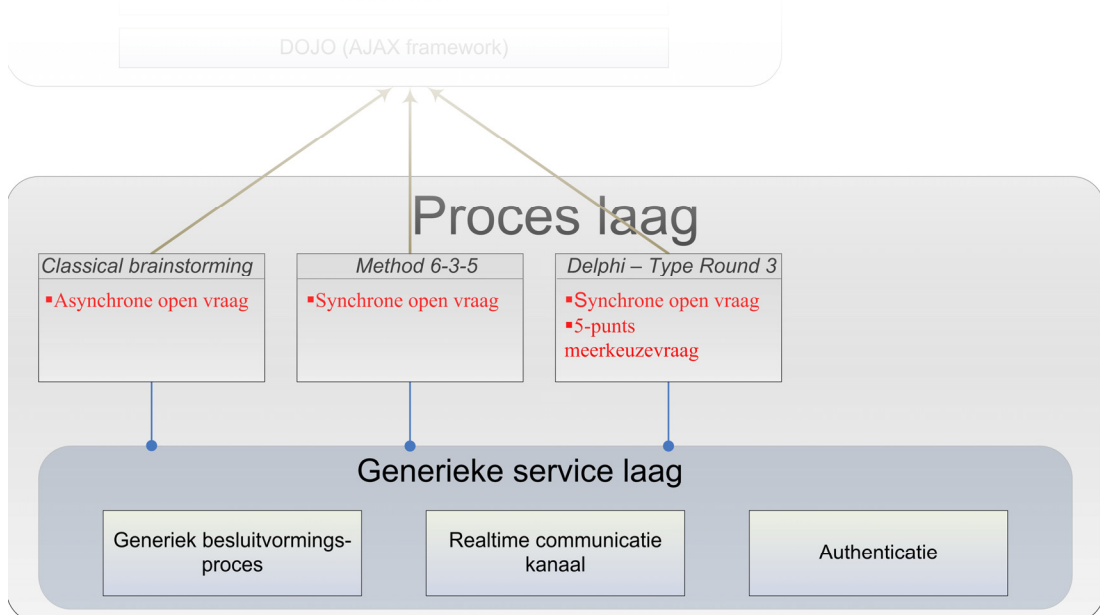
Wanneer na het opstellen van het prototype terug gekeken wordt naar de architectuur, kunnen een paar onvolmaaktheden worden vastgesteld. De belangrijkste fout die werd vastgesteld wordt hier uitgebreid beschreven, inclusief een voorstel tot verbetering.

#### Verkeerde vorm van abstractie in besluitvormingsproces

De architectuur geeft een oplossing om flexibel besluitvormingsprocessen te kunnen toevoegen of aanpassen door middel van een “Generieke servicelaag” en aparte besluitvormingsprocessen die hiervan gebruik kunnen maken.

In de architectuur (“Ontwikkeling view”) wordt echter in zo een besluitvormingsproces aangegeven wat voor vraagtypes er worden gebruikt (vb: “Asynchrone open vraag”). Welke vraagtypes worden gebruikt is echter niet afhankelijk van een besluitvormingsproces, maar wel van de bevraging rondes die in het besluitvormingsproces voorkomen (Eén besluitvormingsproces heeft één of meerdere bevraging rondes, zie “Overkoepelende / Facilitator view”).

Hierbij kunnen tijdens de verschillende bevraging rondes die tot één besluitvormingsproces behoren toch verschillende vraagtypes worden gebruikt. Deze fout is te zien in Figuur 9. De rode tekst wijst op de vraagtypes die onterecht worden gespecificeerd op het besluitvormingsproces niveau.



**Figuur 8: Foute abstractieniveau gevonden in architectuur (ontwikkeling view)**

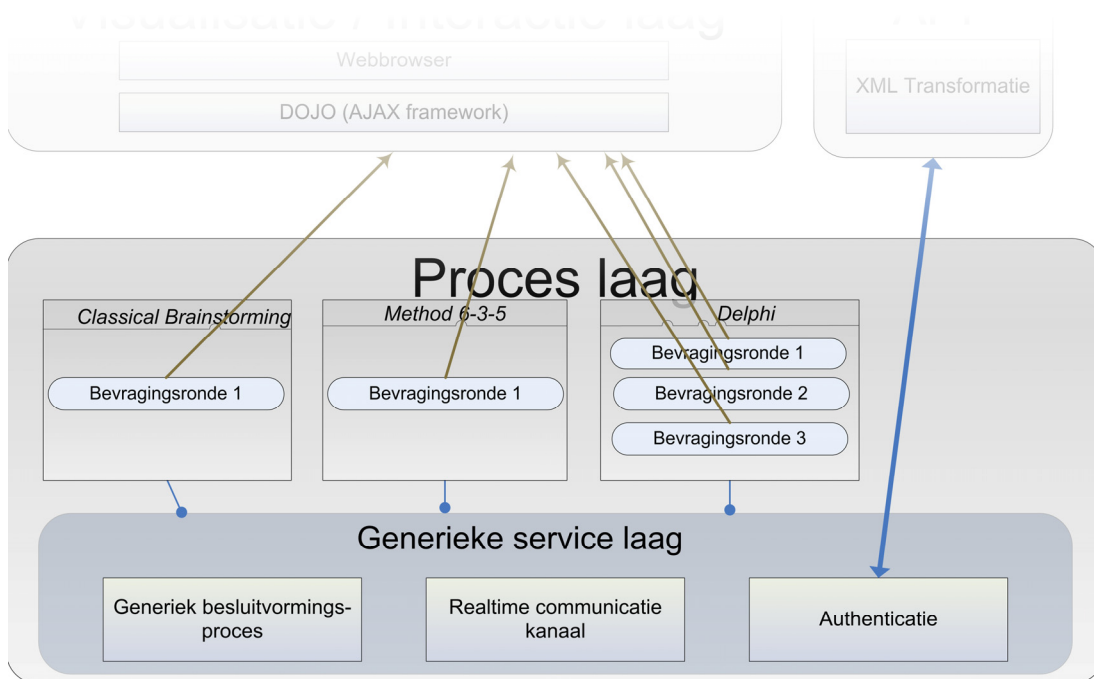
Een andere fout, die hiermee sterk gerelateerd is, is dat het abstractieniveau niet eenduidig is. De view specificieert namelijk dat “Classical Brainstorming”, “Method 6-3-5” en “Delphi – Type Round 3” besluitvormingsprocessen zijn. “Delphi – Type Round 3” is echter een bevragingronde in het “Delphi” proces, en is geen besluitvormingsproces op zich.

Omdat de bevraging rondes de “Visualisatie- en interactie” laag moeten aansturen, is ook hier een fout in het ontwerp gekropen. In de ontwikkeling view staat namelijk dat de “Visualisatie- en interactie” laag wordt opgebouwd uit besluitvormingsprocessen, wat niet het



geval is. Het is één specifieke bevragingronde die bij de eindgebruiker geladen wordt (met de bijbehorende interactie), en niet het hele besluitvormingsproces.

Al deze fouten zijn terug te leiden tot één ontwerpfout: de processen die op de “Generieke service” laag inkoppelen zijn op een verkeerd abstractieniveau gemaakt. Om dit probleem op te lossen, zouden deze processen moeten worden aangepast, zodat bevraging rondes de “Visualisatie en interactie” laag aansturen en niet de besluitvormingsprocessen. Figuur 10 geeft een schets hoe dat een verbetering eruit zou kunnen zien.



**Figuur 9: Mogelijke verbetering software architectuur - ontwikkeling view**

## 10.2. Evaluatie architectuurbeslissingen

### 10.2.1. API als vorm van data-uitwisseling met externe systemen

De API is in het prototype uitgewerkt als een webservice. Hierdoor kunnen externe systemen (waaronder het KnoSoS-systeem) platformafhankelijk gegevens opvragen. De API is, in beperkte mate, uitgewerkt: enkel gegevens over bevraging rondes die open staan kunnen worden opgevraagd. Hierbij zijn geen architecturale beperkingen of noemenswaardige fouten opgetreden en verliep alles zoals beschreven in de architectuurbeslissing.

### 10.2.2. Ondersteuning van meerdere databases

De ondersteuning van meerdere databases gaf bij de implementatie meer problemen dan aanvankelijk werd verwacht. In de architectuurbeslissing staat vermeld dat het implementeren van deze feature geen aanzienlijke extra ontwikkelingstijd zal vragen. Hierbij is echter geen rekening gehouden met databasespecifieke handelingen en commando's. Zo is het selecteren van de laatst toegevoegde unieke ID van een rij in MySQL "LAST\_INSERT\_ID()" en bij MSSQL is dit "@@IDENTITY". Bijgevolg moet per database een aangepast query worden aangemaakt om de gegevens te verwerken. Hierdoor neemt de onderhoudbaarheid van de persistentie laag enorm af.

Er moet dus een groot vraagteken worden geplaatst bij de eis van de ondersteuning van meerdere databases. Hierbij zal een trade-off moeten worden gemaakt tussen enerzijds extra onderhoudskosten en tussen anderzijds de mogelijkheid om van databaseproduct te kunnen omschakelen.

### **10.2.3. Ondersteunen van meerdere besluitvormingsprocessen**

Het ondersteunen van meerdere besluitvormingsprocessen door middel van een “Generieke service” laag was een correcte beslissing. Deze “Generieke service” ondersteunt de gemeenschappelijke kenmerken van de processen. Indien een uitbreiding of aanpassing moet gemaakt worden bij deze eigenschappen moet dit ook maar op één plaats gebeuren, wat dus de onderhoudbaarheid en flexibiliteit van het systeem ten goede komt.

### **10.2.4. AJAX om interactiviteit te verhogen**

Het gebruik van AJAX heeft reeds zijn toegevoegde waarde bewezen. Met behulp van enkele slimme technieken kon op een vrij eenvoudige manier een structuur worden opgezet die het toelaat om gegevens uit te wisselen tussen de eindgebruiker en de webserver om zo de interactiviteit voor de eindgebruikers te verhogen.

Hiernaast biedt Dojo, het gekozen AJAX framework, ook enkele handigheidjes die het werken met Javascript vereenvoudigen. Waar echter onvoldoende rekening mee is gehouden bij het nemen van deze architectuurbeslissingen, is de impact die deze beslissing heeft op de ontwikkelingstijd van het platform: het ontwikkelen in Javascript is namelijk erg arbeidsintensief. Op dit moment zijn er te weinig ondersteunende tools (IDE, syntax checkers, debuggers) beschikbaar die het werken met Javascript kunnen vereenvoudigen. Het gebruik van AJAX in het platform heeft dus zeker zijn toegevoegde waarde, maar dat gaat ten koste van ontwikkelingstijd.