

Afstudeerverslag

Integratie van Distributed Directory Service in een betrouwbare IPv6 netwerkomgeving

Uitgevoerd bij Logica Nederland BV in het kader van het Working Tomorrow programma

Auteur	: R.D. van Reems
Datum	: 21-03-2010
Versie	: 1.0
Divisie	: Working Tomorrow
Competence	: Technical Software Engineering
Onderwijsinstelling	: Haagse Hogeschool
Opleiding	: Technische Informatica
Begeleidende examinerator	: Dhr. J.P.M. de Vreught
Expert examinerator	: Dhr. F. Sikkema
Logica opdrachtgever	: Dhr. M. Konijn
Logica begeleider	: Dhr. M. Stikkelorum
Logica voormalig begeleider	: Dhr. H. De Vreeze
Logica technisch expert	: Dhr. P. Rodriguez de Andrés
Logica architect	: Dhr. E. Essenius

Versiebeheer

Versie	Omschrijving
v 0.1	Pre-conceptversie voor review met dhr. J.P.M. de Vreught.
v 0.2	Conceptversie voor concept-bespreking.
v 0.3	Conceptversie voor tussentijdse assessment.
v 0.4	Conceptversie voor tussentijdse assessment HHS.
v 0.5	Conceptversie voor bespreking Logica-architect en begeleider.
v 1.0	Definitieve versie.

Tabel 1: Versiebeheer

Distributielijst

Naam	Instantie	Rol/ Functie
Dhr. J.P.M. de Vreught	HHS	Begeleidende examinerator
Dhr. F. Sikkema	HHS	Expert examinerator
Extern gecommiteerde		Extern gecommiteerde

Tabel 2: Distributielijst

Lijst van afkortingen

Afktoring	Betekenis
API	Application Programming Interface.
BID	Backlog ID.
CVIS	Cooperative Vehicle Infrastructure Systems.
CVIS-DDS	CVIS Distributed Directory Service.
DNS	Domain Name System.
FOAM	Framework for Open Application Management (een sub-project van CVIS).
IDE	Integrated Development Environment.
OMG	Object Management Group.
OMG-DDS	OMG Data Distribution Service.
OSGi	Open Services Gateway initiative (organisatie voor open standaarden).
UML	Unified Modeling Language.
URI	Universal Resource Identifier.
URL	Uniform Resource Locator.

Tabel 3: Lijst van afkortingen

Verklarende woordenlijst

Term	Toelichting
Backlog	Scrum term voor prioriteitenlijst van de te ontwikkelen functionaliteiten van een applicatie.
CVIS-object	Een hardware of software-matig onderdeel van CVIS.
DDS-configurator	Applicatie dat de CVIS-DDS cliënt configureert met het adres van de CVIS-DDS server.
Demonstrator	Parkeergarage systeem dat belangrijke eigenschappen van het CVIS-DDS systeem demonstreert.
FOAM-omgeving	Platform waarop de bundels van het CVIS-DDS systeem draaien, dit is gebaseerd op de OSGi-standaard.
In-Vehicle-Unit	Systeem dat zich in het vervoersmiddel bevindt.
PostGIS	Voegt ondersteuning voor geografische objecten toe aan het PostgreSQL database systeem.
PostgreSQL	Open source object-relationeel database systeem.
Property	Eigenschap van een CVIS-object dat geregistreerd is in het CVIS-DDS systeem.
Roadside-Unit	Systeem langs de kant van de weg die deel uit maken van de infrastructuur waarmee In-Vehicle systemen kunnen communiceren.
SADescription	Klasse van de CVIS-DDS API. Een SADescription bevat de URL, SADescription type en eventuele properties.
SADescription type	Typeaanduiding van een SADescription waarop een CVIS-DDS object gelokaliseerd kan worden.
SearchConstraint	Klasse van de CVIS-DDS API waarmee de resultaten van een zoekopdracht beïnvloed kunnen worden.
Software-provisioning systeem	Onderdeel van FOAM waarmee bundels automatisch geladen, gestart, gestopt en herladen worden.
SQL-creation script	Creatie script van een database object.
Use-case	Toont de actoren en de gebruiksfunctie van het systeem.

Tabel 4: Verklarende woordenlijst

Voorwoord

Voor u ligt het afstudeerverslag van Robert van Reems. Dit verslag is opgesteld tijdens het afstudeerproject bij Logica Nederland BV. Dit rapport beschrijft het proces dat is doorlopen, de daarbij gemaakte keuzes worden hier toegelicht.

Een dankwoord is verschuldigd aan de begeleiding vanuit Logica en de Haagse Hogeschool:

Hans de Vreught,

Frits Sikkema,

De extern gecommitteerde,

Edwin Essenius,

Hans de Vreeze,

Marco Stikkelorum.

Veel leesplezier toegewenst!

Met vriendelijke groet,

Robert van Reems

Inhoudsopgave

1 Inleiding.....	1
2 De organisatie.....	3
3 De initiatiefase	4
3.1 De uitgangssituatie.....	4
3.2 Wijziging in opdrachtomschrijving.....	4
3.3 Strategie	6
3.4 Fasering.....	9
3.5 De toegepaste technieken.....	11
3.6 Het ontwikkelplatform.....	12
3.7 De afbakening	13
3.8 Het opstellen van de risico's.....	13
3.9 Kwaliteitsbevordering	14
4 De onderzoeksfase.....	17
4.1 Beschrijven van de netwerk-topologie.....	17
4.2 Literatuurstudie CVIS-DSS.....	17
4.3 Literatuurstudie OMG-DDS.....	19
4.4 Vergelijking CVIS-DDS en OMG-DDS.....	20
4.5 Klassediagrammen van CVIS-DDS API	20
4.6 Verificatie functionaliteiten CVIS-DDS.....	22
4.7 Exploratie van nieuwe functionaliteiten CVIS-DDS	27
4.8 Uitloop onderzoeksfase.....	28
5 Verbeterde strategie.....	30
5.1 De aanpak van de sprints.....	31
5.2 De sprints.....	33
6 De realisatie fase.....	36
6.1 Prototype sprint	36
6.2 Parkeer-systeem sprint.....	41
6.3 Parkeer-systeem verbetering sprint.....	47
6.4 Conclusie.....	50
6.5 Afsluiting van het project.....	51
7 Evaluatie.....	53
7.1 Productevaluatie.....	53
7.2 Procesevaluatie	57
7.3 Lessons learned.....	59
7.4 Probleem opgelost?.....	60
7.5 Doelstelling behaald?.....	60
8 Literatuurlijst.....	61
9 Bijlagen.....	63

1 Inleiding

De opdracht is uitgevoerd bij Logica Nederland BV. Deze organisatie maakt onderdeel uit van het CVIS-consortium. Aan dit consortium zijn meer dan zestig Europese organisaties verbonden, die in opdracht van de Europese Commissie aan coöperatieve verkeerssystemen werken. Om deze coöperatieve verkeerssystemen mogelijk te maken is er een applicatie ontwikkeld waarin verschillende systemen zich kunnen registreren; het Distributed Directory Service systeem, ook wel het CVIS-DDS systeem genoemd. Deze applicatie maakt het mogelijk om de verschillende systemen te lokaliseren, om deze zo met elkaar te kunnen laten communiceren.

Voorafgaand aan dit project waren het nut en de mogelijkheden van het CVIS-DDS systeem onbekend. Hieruit is de volgende doelstelling geformuleerd:

“De functionaliteiten en werking van het CVIS-DDS systeem achterhalen en verhelderen binnen de organisatie. Daarbij wordt de werking van het systeem geïllustreerd en worden de tekortkomingen van het systeem verhelderd.”[R2009-PVA]

Dit is een heel andere doelstelling dan dat de titel in het Afstudeerplan van dit project doet vermoeden. Aan het begin van de initiatiefase is duidelijk geworden dat de opdrachtschrijving in het Afstudeerplan niet meer aan de orde was. Het belangrijkste doel was het ontwikkelen van een betrouwbaar IPv6 netwerk; een dergelijk netwerk bleek echter al te bestaan. De nieuwe opdracht heeft een groot onderzoeksaspect gekregen, waardoor de strategie en fasering van dit project aangepast zijn. Er is dan ook gebruik gemaakt van de Prince 2 projectmanagement methode; niet van de ASI-methode zoals omschreven in het Afstudeerplan.

Het onderzoek naar de functionaliteiten en werking van het CVIS-DDS systeem is onderverdeeld in drie stappen:

- Een literatuurstudie;
- Validatie;
- Exploratie.

In de literatuurstudie is er onderzoek gedaan naar de werking en functionaliteiten die het CVIS-DDS systeem zou moeten hebben volgens de literatuur. Hierbij is tevens gekeken naar literatuur over de OMG-DDS standaard, die enkele overeenkomsten met het CVIS-DDS systeem vertoont.

In de verificatie stap zijn de functionaliteiten die uit de literatuurstudie voortgekomen zijn geverifieerd met testopstellingen.

Gedurende de exploratie van het CVIS-DDS systeem zijn er verschillende experimenten uitgevoerd om aanvullende functionaliteiten en toepassingen van het CVIS-DDS systeem te ontdekken. Deze zijn ook gevonden.

Tijdens de realisatiefase is er een demonstrator ontwikkeld, die laat het nut van zien hoe de verschillende onderdelen van het CVIS-DDS systeem werken en of deze onderdelen wel werken op de manier waarop het staat beschreven in de literatuur. Hieruit is gebleken dat niet alle onderdelen daadwerkelijk werken. Hierover staat meer beschreven in het hoofdstuk 4.

Logica maakt deel uit van het CVIS-consortium. Omdat dit project een product van dit consortium onderzoekt is enige kennis van CVIS nodig. De lezers die niet bekend zijn met CVIS of de CVIS-terminologie worden aangeraden eerst bijlage 3 te lezen voordat verder gegaan wordt aan het lezen van dit verslag. Bepaalde specifieke termen en systemen van CVIS worden in deze bijlage toegelicht.

Naast CVIS-terminologie wordt in dit verslag enkele veel gebruikte ICT-termen gebruikt. Deze termen worden niet in dit verslag niet toegelicht.

2 De organisatie

Dit hoofdstuk beschrijft de omgeving waarin het afstudeertraject plaatsvindt. De opdracht wordt uitgevoerd bij de organisatie Logica, in het paragraaf hieronder wordt de organisatie verder toegelicht.

De huidige organisatie Logica is op 30 december 2002 ontstaan uit het voormalige Logica (60%) en het voormalige CMG (40%). Sinds 2006 maakt het Franse Unilog onderdeel uit van het bedrijf. Logica is een internationale ICT-dienstverlener en heeft wereldwijd momenteel 40.000 werknemers in 39 verschillende landen in dienst. Het behoort, ook qua omzet, tot de internationale top-20 in de ICT-dienstverlening.

Het CVIS-team van Logica en daarmee ook dit project valt onder de practice “Professional Skills”. Omdat CVIS raakvlakken heeft met Intelligent Transport Systems (ITS) valt dit project onder de Business Unit TSE (Technical Software Engineering).

3 De initiatiefase

Dit hoofdstuk beschrijft het proces van de initiatiefase. Aan het begin van de initiatiefase is tijdens het oriënterend onderzoek naar CVIS-DDS duidelijk geworden dat de opdrachtschrijving zou wijzigen ten opzichte van de opdrachtschrijving in het Afstudeerplan. Er is een nieuwe opdrachtschrijving gemaakt omdat de opdrachtschrijving in het Afstudeerplan niet meer van toepassing was. Deze wijzigingen zijn in dit hoofdstuk toegelicht.

3.1 De uitgangssituatie

De beschikbare informatie van het CVIS-DDS systeem bestaat uit de volgende twee documenten:

- D.CVIS.3.3 Architecture and System Specifications [SSBA2009] en,
- D.CVIS.3.2 High Level Architecture [SSLO2007].

Deze twee documenten beschrijven het CVIS-DDS systeem, echter komen de omschrijvingen van het CVIS-DDS systeem in de documenten niet volledig overeen met elkaar en zijn de omschrijvingen erg abstract.

Overige (technische) informatie is aanwezig bij de technisch expert, deze beschikt over een aantal e-mail conversaties met technische informatie over het CVIS-DDS systeem.

De organisatie heeft al een CVIS-DDS server draaiende en is staat om de CVIS-DDS cliënt te configureren met een “DDS-configurator” applicatie.

3.2 Wijziging in opdrachtschrijving

Tijdens het inlezen op de literatuur ([SSLO2007] & [SSBA2009]) werd duidelijk dat het CVIS-DDS systeem niet een systeem is zoals dat omschreven is in de aanleiding van het Afstudeerplan.

De volgende zin, die afkomstig is uit de aanleiding van het Afstudeerplan, omschrijft het CVIS-DDS systeem incorrect:

“De verkeersdeelnemer moet in staat gesteld worden om informatie middels het CVIS-DDS systeem over eventuele lokale evenementen, horeca-aangelegenheden, vrijetijdsbesteding en dergelijke te kunnen raadplegen.” [R2009-AP]

Het CVIS-DDS systeem is géén informatiesysteem met informatie over lokale evenementen noch horeca-aangelegenheden, noch vrijetijdsbesteding. Het is een systeem waarin type, eigenschappen en IP-adressen van diensten geregistreerd kunnen worden.

Toen hierover met de opdrachtgever en technisch expert gesprekken gevoerd werd, bleek dat de aanleiding incorrect was. Uit de gesprekken kwam ook naar voren dat de opdrachtschrijving niet was afgestemd tussen de opdrachtgever en technisch expert. De technisch expert had een andere interpretatie van de opdrachtschrijving dan de opdrachtgever.

Omdat de aanleiding van de opdracht gewijzigd is en de omschrijving van het CVIS-DDS systeem niet correct was, is een nieuwe opdrachtschrijving opgesteld. De nieuwe opdrachtschrijving (aanleiding, probleemstelling en doelstelling) staat hieronder.

Aanleiding uit plan van aanpak

Het CVIS-project heeft behoefte aan een dynamisch systeem dat CVIS-objecten, zoals diensten en applicaties, beheert en organiseert; een soort “Gouden Gids”. Dit systeem moet naast registratie ook voorzien in automatische lokalisatie van diensten. Deze eigenschappen zijn benodigd om coöperatieve systemen in een mobiele omgeving mogelijk te maken.

Probleemstelling uit plan van aanpak

De software van het CVIS-DDS systeem is opgeleverd door Technolution BV. De werking van dit systeem en de functionaliteiten zijn bij Logica onbekend waardoor het mogelijk niet optimaal gebruikt wordt; Zo wordt het systeem nu toegepast om de GPS-locatie van de CVIS-vervoersmiddelen te registreren en bij te werken.

Tijdens de implementatie van het CVIS-DDS systeem is men tegen vele problemen aangelopen, onbekend is welke problemen er nu nog bestaan.

Doelstelling uit plan van aanpak

De functionaliteiten en werking van het CVIS-DDS systeem achterhalen en verhelderen binnen de organisatie. Daarbij wordt de werking van het systeem geïllustreerd en worden de tekortkomingen van het systeem verhelderd.

Resultaat uit plan van aanpak

Het resultaat is een demonstrator dat de functionaliteiten van het CVIS-DDS systeem demonstreert. Daarnaast is in de organisatie kennis van werking, de mogelijkheden en de tekortkomingen van het CVIS-DDS systeem.

Producten uit plan van aanpak

- Bevindingenrapportage en
- De demonstrator.

Competenties

Door het veranderen van de opdrachtschrijving kon de competentie “C9 Ontwerpen van een infrastructuur” (Niveau 3) niet meer uitgevoerd worden. Daarom is de competentie in overleg met de examinatoren gewijzigd naar C8, zoals op te maken is uit de opsomming van de nieuwe competenties op de volgende pagina.

De nieuwe competenties:

- A1 Niveau 4 Analyseren van het probleemdomein
In dit afstudeerverslag is beschreven hoe de analyse naar het probleemdomein uitgevoerd is bij het onderzoek naar de mogelijkheden, kwaliteit en problemen van CVIS-DDS het CVIS-DDS systeem.
- C10 Niveau 3 Ontwerpen van een systeemarchitectuur
De systeemarchitectuur van de demonstrator is beschreven in het Scrum werkdocument. In het afstudeerverslag zijn een aantal onderdelen van de architectuur toegelicht en wordt er in gegaan op de aansluiting van de systeemeisen. Ook worden de keuzemogelijkheden wat betreft de alternatieven toegelicht.
- C8 Niveau 3 Ontwerpen van een technisch informatiesysteem
In het Scrum werkdocument zijn de ontwerpen van de demonstrator applicatie en alle onderdelen daarvan geplaatst. Naast de diagrammen worden de ontwerpen ook formeel toegelicht.
- H6 Niveau 3 Professioneel werken: Resultaatgericht werken
In de evaluaties worden kwaliteitseisen en de bewaking hiervan in het evaluatierapport beschreven.

3.3 Strategie

Projectmanagementmethode

Omdat de opdrachtomschrijving gewijzigd is, is de strategie aangepast zodat deze aansluit op de nieuwe opdrachtomschrijving. In het Afstudeerplan is beschreven dat de ASI-methode toegepast wordt. De methode is vervangen door de Prince 2 projectmanagement methode.

De ASI-methode paste niet meer binnen het project na het wijzigen van de opdrachtomschrijving. ASI richt zich op het ontwikkelen van een technische infrastructuur, zie het citaat hieronder:

“... De handleiding is bedoeld voor projectleiders, architecten, en ontwerpers die betrokken zijn de ontwikkeling van een technische infrastructuur.” [KN1995].

In dit project wordt geen technische infrastructuur ontworpen noch ontwikkeld.

Samen met de organisatie zijn de specifieke behoeften voor een projectmanagementmethode opgesteld. Deze behoeften zijn hieronder opgesteld:

- Structureren van het project;
- Beheersen van de voortgang;
- Beheersen van de risico's;
- Bieden van bijstuur-mogelijkheden.

De projectmanagementmethode Prince 2 voldoet aan de behoeften die hiervoor opgesteld zijn. Het beheersen van de project uitvoering en fasenovergangen dat door Prince 2 wordt voorgeschreven, beschrijft een structuur en aanpak dat is gebaseerd op praktijkervaring.

De processen die door Prince 2 beschreven worden waarmee voortgang van het project periodiek is beoordeeld en bijgestuurd (Controlling a Stage) voldoen aan de gestelde behoeften: “Beheersen van de voortgang” en “Bieden van bijstuur-mogelijkheden”.

Met Prince 2 wordt ook de communicatie tussen de uitvoerder en de project board gestructureerd, middels een aantal rapporten. De bekendheid van Prince 2 binnen de organisatie heeft ook voordelen voor wat betreft de communicatie, de Prince 2 rapportages en terminologie zijn namelijk bekend bij de leden van de project board.

De project board bestaat uit de volgende personen:

- De bedrijfsmentor: Marco Stikkelorum;
- De voormalige bedrijfsmentor: Hans de Vreeze;
- De opdrachtgever: Marcel Konijn;
- De architect: Edwin Essenius.

Er bestaat een informele relatie met de hierboven genoemde personen. De personen bevinden zich (op de architect na) op de zelfde verdieping waardoor regelmatig, zowel formeel als informeel, contact plaatsvindt. Deze situatie vraagt dus niet om wekelijkse highlight rapportage.

In dit project is er een aantal contact-momenten met de project board ingebouwd. Dit zijn de bijstuur-mogelijkheden. Zo vindt er eens in de twee weken voortgangsoverleg plaats met een lid van de project board, de bedrijfsmentor. Tijdens de overgang van de fasen wordt de afgeronde fase geëvalueerd en wordt de status van het project besproken.

De volgende Prince 2 processen komen binnen dit project voor:

- Directing a Project;
- Initiating a Project;
- Controlling a Stage;
- Managing stage Boundaries;
- Closing a Project;
- Planning.

De processen “Controlling a Stage” en “Managing Product Delivery” zijn gecombineerd. Volgens Prince 2 worden de projectwerkzaamheden uitgevoerd door de projectmedewerkers. Er is in dit project echter maar één projectmedewerker. Bovendien kunnen de taken uit Managing Product Delivery (acceptatie van de producten en de bewaking van de voortgang en kwaliteit) prima uitgevoerd worden in het proces Controlling a Stage; beide processen hebben overlappende taken.

Directing a Project

Onder dit proces valt het sturen van het project door de project board. De project board wordt benaderd wanneer het project buiten de gestelde grenswaarden (kwaliteit en tijd) valt. Ook maakt de project board de beslissingen over de te ontwikkelen onderdelen van de demonstrator.

De input voor het “Directing a Project Proces” zijn:

- Tweewekelijkse gesprekken met de bedrijfsbegeleider;
- Regelmatig overleg met opdrachtgever;
- Exception rapportages;
- Oplevering van producten.

Het proces Controlling a Stage

Het proces Controlling a Stage proces bestaat in dit project uit de volgende processtappen:

- Sturing van het project;
- Escaleren (indien nodig);
- Kwaliteitsbevordering.

Meer informatie over de stappen 'Sturing van het project' en 'Escaleren' kunt u teruglezen in het Plan van Aanpak. De stap kwaliteitsbevordering is hieronder beschreven.

Om de kwaliteit van de producten op niveau houden en de status ervan te kunnen volgen wordt combinatie van Configuratie – en Change management toegepast. Elk document beschikt over zogeheten versie beheer pagina waarin de status van het document omschreven wordt. In elk document wordt de status en de aangebrachte wijzigingen van het document beschreven en bijgehouden wie het document gereviseerd heeft. De status van de producten wordt ook gevolgd in de detailplanning van de betreffende fase.

Tijdens het proces 'Managing stage Boundaries' worden de eisen van de producten in de opkomende fasen beschreven.

Controlling a Stage verzorgt het opleveren van de producten die opgeleverd worden en is verantwoordelijk voor het realiseren en opleveren van de producten.

Als voorbereiding op het tweewekelijkse gesprek met de opdrachtgever, is de voortgang van het project eerst beoordeeld door mijzelf. Dit gesprek is gevoerd met een lid van de project board. Omdat er al op regelmatige basis contract momenten zijn met de project board worden geen highlight rapporten opgeleverd.

Het proces Managing stage Boundaries

Tijdens het proces “Managing stage Boundaries” vindt overleg plaats met de project board over de behaalde resultaten. De board heeft hierbij de mogelijkheid om bij te sturen. In dit proces wordt de afgeronde fase geëvalueerd en de komende fase wordt voorbereid.

Het proces Closing a Project

Tijdens dit proces worden alle producten opgeleverd bij de opdrachtgever. Eventuele verbeterpunten aan de producten worden verwerkt indien de opdrachtgever dit wenst.

Tevens wordt de afgeronde fase geëvalueerd. Daarbij wordt ook een “lessons learned” opgesteld.

Het proces Planning

Het proces planning wordt tijdens het gehele project uitgevoerd. Echter is het proces het meest actief tijdens het initiëren van de fase en het proces “Managing stage Boundaries”. De globale planning wordt gemaakt tijdens de initiatiefase en wordt daarnaast continu bijgewerkt tijdens de uitvoering van het project. De detailplanningen worden gemaakt tijdens het opstarten van nieuwe fasen (Managing stage Boundaries).

3.4 Fasering

Tijdens het opstellen van het Afstudeerplan is een fasering aan het project aangebracht. De nieuwe fasering is te zien in de rechter zijde van tabel 5. Onder het tabel zelf volgt een toelichting.

Afstudeerplan	Plan van Aanpak
Initiatiefase	De initiatiefase
Definitiefase	
Architectuurfase	De onderzoeksfase
Ontwerpfase	De ontwerpfase
Realisatiefase	De realisatiefase

Tabel 5: Fasering van het project in Afstudeerplan en PvA

De oude fasering sloot niet goed meer aan op de opdrachtoomschrijving. Met name de taken van de architectuurfase, waarin de architectuur van het te ontwikkelen netwerk wordt vastgelegd, zijn overbodig door de nieuwe opdrachtoomschrijving. Er wordt namelijk geen nieuw netwerk ontworpen noch geïmplementeerd. In tabel 5 ziet u de de nieuwe fasering van het project zoals deze opgesteld is in het plan van aanpak. Op de volgende pagina's worden de fasen in het kort toegelicht.

De onderzoeksfase

In deze fase is een literatuurstudie uitgevoerd naar het CVIS-DDS systeem. Het CVIS-DDS systeem wordt namelijk abstract beschreven in de documenten: [SSBA2009] en [SSLO2007]. Het doel van deze literatuurstudie is het achterhalen van de functionaliteiten en mechanismen waar het CVIS-DDS systeem over beschikt.

Onderdeel van de literatuurstudie is een vergelijking met het een reeds bestaande standaard dat overeenkomsten heeft met het CVIS-DDS systeem; dit is de Data Distribution Service standaard van de Object Management Group (OMG-DDS). Het doel van de vergelijking met het CVIS-DDS systeem en de OMG-DDS standaard is het achterhalen van eventuele nieuwe functionaliteiten en mechanismen die niet in de CVIS-literatuur omschreven zijn.

Nadat de literatuurstudie uitgevoerd is, zijn de functionaliteiten geverifieerd met behulp van enkele experimenten.

Het product van de onderzoeksfase is het onderzoeksrapport. Daarnaast wordt een tussentijdse versie van de bevindingenrapportage opgeleverd. Na het afronden van de onderzoeksfase zijn namelijk al een aantal bevindingen te beschrijven.

De ontwerpfase

Tijdens de ontwerpfase wordt de programmatuur van de demonstrator ontworpen. De functionaliteiten en mechanismen die tijdens de onderzoeksfase duidelijk geworden zijn, vormen de input voor de eisen die aan de demonstrator gesteld worden.

Naast het ontwerpen van de demonstrator wordt in deze fase ook het testplan opgesteld.

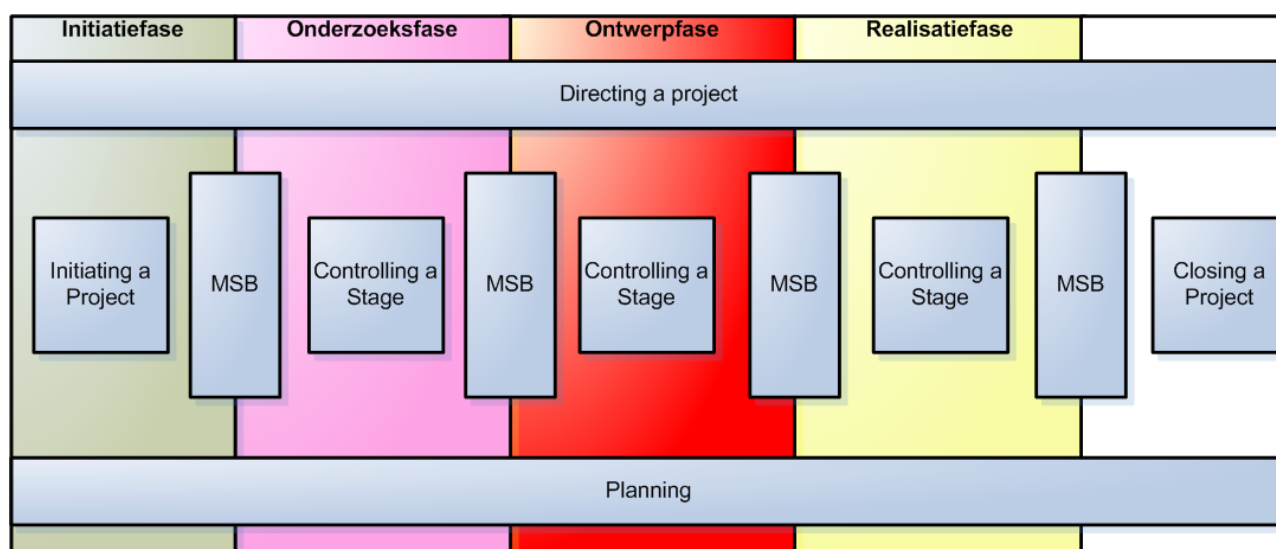
Opmerking:

De fasering van het project is aangepast tijdens de onderzoeksfase. Daarbij zijn de taken van de ontwerpfase verplaatst naar de realisatiefase. Hierover kunt u meer lezen in hoofdstuk 5.

De realisatiefase

Tijdens de realisatiefase worden de ontwerpen van de demonstrator geïmplementeerd en wordt de demonstrator getest volgens het testplan. Na de realisatiefase wordt het project afgesloten middels het proces "Closing a Project".

Op de volgende pagina staat een overzicht van de fasering van het project met de Prince 2 processen, de toelichting volgt na de illustratie.



Figuur 1: Projectfasering van project met Prince 2 processen

In het figuur hierboven staan de vier fasen van het project (initiatiefase, onderzoeksfase, ontwerpfase en realisatiefase).

Tijdens de onderzoeksfase, de ontwerpfase en de realisatiefase wordt het proces “Controlling a Stage” uitgevoerd. Zoals eerder beschreven zijn de Prince 2 processen “Controlling a Stage” en “Managing Product Delivery” gecombineerd. Daarom is het Prince 2 proces “Managing Product Delivery” niet weergegeven in de illustratie.

3.5 De toegepaste technieken

Er is gekozen om de demonstrator te ontwerpen met behulp van de UML-modelleertaal. UML is de de facto standaard waar ook de opdrachtgever bekend mee is. Dat gegeven vind ik erg belangrijk omdat dan goed gecommuniceerd kan worden met de opdrachtgever over de te realiseren software. UML is geschikt voor dit project om de opdrachtgever inzicht te geven in de functionaliteiten die ontwikkeld worden en welke stappen de actor onderneemt (use-cases). Daarnaast kan de “inhoud” en toepassing van de klasse geïllustreerd worden met behulp van de UML-klassediagrammen en beschrijvingen. De interne communicatie van de applicaties is geïllustreerd aan de hand van enkele sequentiediagrammen. De sequentiediagrammen zijn alleen gebruikt bij situaties waar de interne communicatie ingewikkeld is. Voor het ontwikkelen van een betrekkelijk klein systeem zijn in mijn ogen niet meer diagramtechnieken nodig.

De literatuurstudie is uitgevoerd door gebruik te maken van de Mindmapping techniek. Met behulp van deze techniek zijn mindmaps gemaakt van de doorgenomen literatuur. Deze techniek heb ik toegepast omdat ik hiermee informatie kan structureren en ik daarmee de kans verklein dat onderdelen in de vergetelheid raken.

Bij het maken van de mindmaps is gebruik gemaakt van de beschreven techniek in het boek 'Mindmapping in de praktijk', dit boek beschrijft de Mindmapping techniek. Ik vond dit boek toepasbaar omdat er mee geadverteerd wordt dat het toegepast kan worden op "teksten van andere"; iets wat bij de literatuurstudie het geval is.

De mindmap techniek, zoals deze toegepast is, bestaat uit een drietal stappen. Dit zijn:

1. Sleutelwoorden onderstrepen;
2. Plaatsen sleutelwoorden in mindmap;
3. Ordenen van de sleutelwoorden in de mindmaps.

In alle teksten zijn sleutelwoorden onderstreept. Sleutelwoorden zijn door de gehanteerde literatuur als volgt omschreven:

"... Sleutelwoorden zijn die woorden, die het best de inhoud van een uitspraak of een tekst samenvatten. Het zijn die woorden die het geheel het best weergeven en die het beste geheugenbeeld oproepen. ..." [DS1994]

De sleutelwoorden zijn vervolgens in de mindmap geplaatst. Daarna zijn de sleutelwoorden in de mindmaps geordend en gegroepeerd op een dusdanige manier dat voor mij logisch is.

De beschreven Mindmapping techniek in het boek stimuleert het creatieve proces door het verzinnen van extra woorden bij het combineren van de opgeschreven sleutelwoorden; Een soort bottom-up aanpak. Deze stap is niet toegepast, omdat de informatie dan niet meer afkomstig is van de gebruikte literatuur maar ook uit mijn gedachten.

3.6 Het ontwikkelplatform

De demonstrator is ontwikkeld in de FOAM-omgeving. De afkorting "FOAM" staat voor Framework for Open Application Management. De FOAM-omgeving is een in Java ontwikkelde framework waarin alle CVIS-toepassingen, ook wel bundels genoemd, van het CVIS-project draaien. Meer informatie over FOAM staat in

Omdat de demonstrator het nut van het CVIS-DDS systeem moet aantonen is gebruik gemaakt van een demonstrator dat tot de verbeelding spreekt. Om de demonstrator tot de verbeelding te laten spreken moet deze ook draaien op de FOAM-omgeving dat gebruikt wordt binnen CVIS. De FOAM-omgeving beschikt namelijk over het dashboard* dat de automobilist ziet in zijn of haar auto.

* De GUI van de FOAM-omgeving dat de automobilist ziet

Op de Roadside Units (De CVIS computersystemen die langs de kant van de weg staan, meer informatie in) en de Onboard Units (De CVIS computersystemen in de vervoersmiddelen, meer informatie in) draait een realtime besturingssysteem. Dit is Ubuntu 8.04 met een customized kernel. Voor de ontwikkeling van de demonstrator maak ik geen gebruik van dit realtime platform, er wordt gebruik gemaakt van een standaard Ubuntu 9.10 besturingssysteem. Het realtime-platform wordt omdat, volgens het CVIS-team, de stabiliteit veel te wensen overlaat niet gebruikt. Bovendien is niet compatibel met veel hardware. Omdat de demonstrator volledig in Java is ontwikkeld levert het afwijken van besturingssysteem geen compatibiliteitsproblemen op.

3.7 De afbakening

Het was tijdens de initiatiefase onbekend welke functionaliteiten de demonstrator uiteindelijk moet demonstreren. Daarom is er voor gekozen de afbakening open te houden in het plan van aanpak. Hieronder staat de afbakening van het plan van aanpak:

“In dit project wordt onderzoek gedaan naar de functionaliteiten en werking van het CVIS-DDS systeem binnen de kaders van het CVIS project.

De mogelijkheden van het CVIS-DDS systeem worden gedemonstreerd en de opgedane ervaringen bij het ontwikkelen van de demonstrator wordt inzichtelijk gemaakt.

Advisering over een alternatief systeem of aanpassingen aan het bestaande CVIS-DDS systeem is geen onderdeel van dit project. Evenals het aanbrengen van verbeteringen aan het huidige CVIS-DDS systeem.” [R2009-PVA]

3.8 Het opstellen van de risico's

De risico's die voorafgaand aan het project voorzien zijn, en in het Afstudeerplan staan, wijken af ten opzichte van de risico's in het plan van aanpak. Ook dit is het gevolg van de vernieuwde opdrachtomschrijving.

Voorafgaand aan het project is tijdens het opstellen van het Afstudeerplan het volgende risico geïdentificeerd: “IPv6 heeft te veel kwetsbaarheden om in de praktijk te gebruiken”. Omdat er in de nieuwe opdrachtomschrijving geen nieuw netwerk ontworpen noch geïmplementeerd wordt komt dit risico te vervallen.

Tijdens de initiatiefase zijn de volgende risico's geïdentificeerd:

- Niet beschikbaarheid van benodigde personen;
- Benodigde functionaliteiten van het CVIS-DDS systeem ontbreken of zijn defect;
- Weinig beschikbare informatie over CVIS-DDS;
- Onbeschikbaarheid van Central Services center;
- CVIS-DDS systeem complex in gebruik.

Hieronder ziet u het risico “Weinig beschikbare informatie over CVIS-DDS”. Dit risico heeft zich voorgedaan tijdens de uitvoering van het project.

Risico	Weinig beschikbare informatie over CVIS-DDS
Omschrijving	De beschikbare documentatie van het huidige CVIS-DDS systeem is onvolledig of ontbreekt. Waardoor de functionaliteiten van het CVIS-DDS systeem moeilijk te achterhalen zijn.
Kans	Mogelijk.
Impact	Vertragend.
Kans verlagende maatregelen	De ontwikkelaars direct benaderen met vragen over het systeem.
Plan B	Meer tijd besteden aan de experimenten om de functionaliteiten van het systeem te ontdekken en te begrijpen.

Tabel 6: Weinig beschikbare informatie over CVIS-DDS

De kans verlagende maatregel is uitgevoerd. De ontwikkelaars zijn benaderd met vragen, echter gaven zij geen thuis. Voor het “plan B” was helaas geen ruimte in de planning. Er is echter wel een exploratie uitgevoerd waarin geëxperimenteerd niet gedocumenteerde functionaliteiten ontdekt zijn. Meer informatie over de exploratie staat in paragraaf 4.7.

3.9 Kwaliteitsbevordering

De maatregelen waarmee de kwaliteit van de producten die in dit project opgeleverd zijn, zijn in deze paragraaf toegelicht; met producten wordt de rapportage en demonstrator bedoeld. Eerst wordt de totstandkoming van de software kwaliteitseisen toegelicht.

In het plan van aanpak is beschreven hoe de kwaliteit van dit project geborgd wordt en wat de maatregelen zijn om de gewenste kwaliteit te behalen en te verifiëren. De kwaliteitseisen en kwaliteitsverhogende maatregelen van dit project zijn gebaseerd op eigen inzicht. Om te voorkomen dat een aantal kwaliteitsaspecten vergeten worden heb ik de eisen afgeleid van de ISO 9126 standaard. De ISO standaard is niet toegepast in dit project maar is gebruikt als bron om tot kwaliteitseisen en maatregelen te komen.

ISO 9126 beschrijft de volgende karakteristieken waar de kwaliteit van software op gespecificeerd kan worden:

- Functionality (functionaliteit);
- Reliability (betrouwbaarheid);
- Usability (bruikbaarheid);
- Efficiency (efficiëntie);
- Maintainability (onderhoudbaarheid);
- Portability (portabiliteit).

De demonstrator is geen systeem dat toegepast wordt in het CVIS-project, daarom zijn de kwaliteitsaspecten betrouwbaarheid en portabiliteit geen belangrijke aspecten wat betreft de kwaliteit van de demonstrator. De demonstrator wordt een beperkt aantal malen gedemonstreerd aan de project board.

Ook de efficiëntie is niet van belang voor het project. De demonstratie van de demonstrator wordt uitgevoerd op moderne computer systemen. Deze systemen beschikken over voldoende capaciteit om de demonstrator te hosten.

De functionaliteit is wel van belang voor de demonstrator. De demonstrator moet geschikt en juist zijn, het moet dus de juiste functionaliteiten van het CVIS-DDS systeem op een geschikte en volledige wijze aantonen.

De bruikbaarheid is ook van belang voor de demonstrator, deze dient immers duidelijk het nut van het CVIS-DDS systeem aan te tonen waarbij het ook gemakkelijk te bedienen is.

Op het gebied van onderhoudbaarheid worden ook maatregelen genomen. Het doel van dit project is de werking van het CVIS-DDS systeem te illustreren. Om inzichtelijk te maken hoe het systeem op technisch vlak werkt moet het systeem onderhoudbaar zijn. De broncode van de demonstrator moet leesbaar en duidelijk zijn, zo kan de programmatuur hergebruikt worden en kan de werking van het CVIS-DDS systeem afgeleid worden aan de hand van de programmatuur. Een deel van de genomen maatregelen staan in de tabel hieronder

Eisen	Maatregel
Functionaliteit	
Geschiktheid.	De opdrachtgever oefent zijn invloed uit op de keuze voor de geschikte functionaliteiten waardoor alleen de geschikte functionaliteiten gerealiseerd worden. Door de bekende functionaliteiten van het CVIS-DDS systeem te betrekken bij het opstellen van de use-cases, wordt de geschiktheid van de te realiseren functionaliteit van de demonstrator heroverwogen. Daarmee wordt voorkomen dat ongeschikte functionaliteiten gerealiseerd worden. De demonstrator moet spreken tot de verbeelding van de project board. Om tot de verbeelding te spreken wordt er gebruik gemaakt van een voorbeeldapplicatie die tot ieders verbeelding spreekt.
Juistheid en volledigheid.	Vooraf worden testcases opgesteld. Na de implementatie van de programmatuur worden die testcases uitgevoerd. Daardoor wordt de juistheid en volledigheid van de functionaliteiten geverifieerd omdat er vooraf is vastgesteld wat er getest wordt en achteraf wordt vastgesteld of het juist en volledig is.
Bruikbaarheid	
Leerbaarheid (werking van het CVIS-DDS systeem aantonen).	Door de demonstrator duidelijk aan te laten geven wat er gebeurd en welke acties het systeem onderneemt kan er geleerd worden van het systeem. Dit is met name belangrijk om de werking van het CVIS-DDS systeem te kunnen aantonen.

Eisen	Maatregel
Duidelijkheid.	Omdat de project board en het Logica CVIS-team de doelgroep van de demonstrator zijn, worden zij nauw betrokken bij de realisatie van de demonstrator. Tijdens de Scrum demonstraties van de demonstrator wordt de project board en het CVIS-team in staat gesteld om de duidelijkheid te beoordelen en beïnvloeden.
Bedieningsgemak.	Door de mogelijkheden in de bediening te beperken neemt het bedieningsgemak toe. Daarnaast moeten de mogelijkheden duidelijk zijn door kernachtig termen te gebruiken in knoppen en menu's.
Onderhoudbaarheid	
...	

Tabel 7: Deel van kwaliteitseisen en maatregelen [R2009-PVA]

In dit hoofdstuk is beschreven dat de opdrachtomschrijving veranderd is ten opzichte van het Afstudeerplan. Dit had als gevolg dat de strategie en fasering van het project aangepast moest worden.

In dit hoofdstuk zijn de beslissingen van de initiatiefase beschreven, belangrijke beslissingen zijn de wijzigingen in de opdrachtomschrijving en strategie. Daarnaast zijn de keuzen voor de toegepaste technieken toegelicht. Als laatste is in dit hoofdstuk behandeld welke maatregelen er genomen zijn om de kwaliteit van de producten te verhogen. Na het afronden van de Managing Stage Boundaries proces is begonnen aan de onderzoeksfase. Meer informatie over de onderzoeksfase staat in het volgende hoofdstuk.

4 De onderzoeksfase

In dit hoofdstuk is het proces van de onderzoeksfase omschreven. Tijdens deze fase is er een literatuurstudie uitgevoerd naar het CVIS-DDS systeem en de OMG-DDS standaard. Door het systeem en de standaard met elkaar te vergelijken kunnen eventuele nieuwe functionaliteiten van het CVIS-DDS systeem achterhaald worden.

De functionaliteiten die uit de CVIS-DDS literatuurstudie gevloeid zijn, zijn geverifieerd. Hieruit is gebleken dat de implementatie afwijkt van de abstracte beschrijvingen uit de literatuur. Een belangrijk mechanisme van het CVIS-DDS systeem, het “Publish and Subscribe” mechanisme, is niet geïmplementeerd.

4.1 Beschrijven van de netwerk-topologie

In het begin van de onderzoeksfase is de CVIS netwerk-topologie in kaart gebracht. Hierdoor is de netwerkarchitectuur duidelijk geworden. Dit is belangrijk omdat de demonstrator, die gebouwd is tijdens dit project, afhankelijk is van het CVIS-netwerk waar het CVIS-DDS systeem onderdeel van is.

De CVIS-netwerkarchitectuur is beschreven in het document: [R2009-netwerk].

4.2 Literatuurstudie CVIS-DSS

Deze paragraaf beschrijft het proces van de literatuurstudie. Als eerste wordt de studie naar CVIS-DDS toegelicht, daarna wordt de studie naar OMG-DDS beschreven. Bij het literatuuronderzoek naar CVIS-DDS is gebruik gemaakt van de Mindmapping techniek. Deze techniek is eerder toegelicht in paragraaf 3.5.

CVIS-DDS

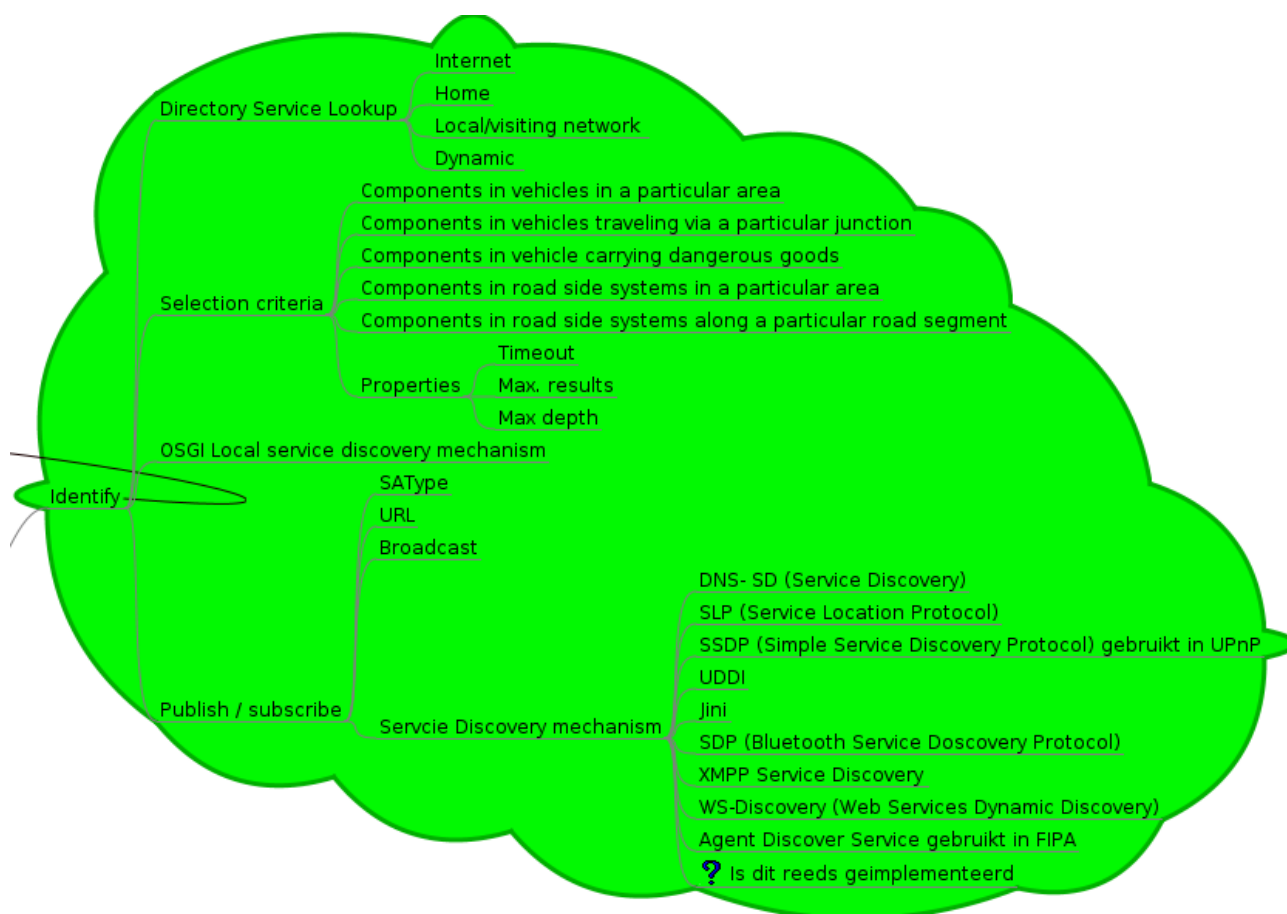
Tijdens de initiatiefase is de literatuur van CVIS doorgenomen om het CVIS-DDS systeem te kunnen omschrijven in het plan van aanpak. Hierbij is alleen gekeken naar de benodigde informatie om het CVIS-DDS systeem te kunnen beschrijven. Daarom was het nodig om tijdens de onderzoeksfase de literatuur diepgaander te bestuderen.

De literatuur is aangeleverd door de organisatie. De beschikbare informatie is helaas erg beperkt. Er zijn maar twee documenten die beide één hoofdstuk wijden aan het CVIS-DDS systeem. Buiten de CVIS organisatie om bestaat er geen kennis van het CVIS-DDS systeem; het CVIS-DDS systeem is geheel specifiek voor CVIS gemaakt.

In de literatuur is het CVIS-DDS systeem op technisch niveau omschreven. Zo zijn er use-cases klasse en sequentie-diagrammen beschreven. Omdat onduidelijk was of de implementatie overeen komt met de literatuur, moesten de functionaliteiten geverifieerd worden.

Zoals in de referentielijst te zien is, komt er één document uit het jaar 2007 ([SSLO2007]) en één uit 2009 ([SSBA2009]). Het laatstgenoemde document is als leidraad genomen, omdat deze het meest recent is. Daarom is er in dit document meer gerefereerd aan [SSBA2009] dan aan [SSLO2007]; dit geldt ook voor de andere rapportages.

Tijdens de literatuurstudie is er een aantal functionaliteiten van het CVIS-DDS systeem geïdentificeerd. Deze functionaliteiten zijn afgeleid uit de mindmap die gemaakt is van de literatuur. Onderstaand in figuur 2 ziet u het meest relevante gedeelte van mindmap. De mindmap is helaas te groot om in zijn geheel te kunnen weergegeven. De complete versie is te zien in bijlage 5.



Figuur 2: Deel mindmap CVIS-DDS

In bovenstaande mindmap is een aantal functionaliteiten van het CVIS-DDS systeem te herleiden. Zo komt het lokaliseren van objecten terug in de mindmap. Het beperken van de zoekresultaten (Selection criteria) en het Publish and Subscribe mechanisme zijn tevens genoemd.

De functionaliteiten die uit de mindmap afgeleid zijn, zijn hieronder opgesomd:

- Registreren en wijzigen van 'properties' (eigenschappen) van een geregistreerd object;
- Lokalisatie van objecten op:
 - Eigenschappen;
 - Internet;
 - Home netwerk (thuis netwerk);
 - Het lokale netwerk;
- Beperken van de zoekresultaten op:
 - Aantal resultaten;
 - Zoektijd;
 - Diepte;
- Publiceren van diensten;
- Inschrijven op diensten;

Tevens zijn er in de mindmap verschillende vormen van architecturen beschreven waarin het CVIS-DDS systeem ingezet kan worden, namelijk:

- CVIS-DDS federatie;
- CVIS-DDS replicatie;
- Gecentraliseerd CVIS-DDS.

De “CVIS-DDS federatie” is een architectuur die overeenkomsten vertoont met de DNS-architectuur. Als een CVIS-DDS server een query zelf niet kan afhandelen, dan wordt de query doorgestuurd naar een hoger liggende CVIS-DDS server, totdat de query al dan niet door een top-level CVIS-DDS server afgehandeld kan worden.

De tweede architectuur is “CVIS-DDS replicatie”. Hierbij beschikken alle CVIS-DDS servers over dezelfde informatie.

De laatste architectuur beschrijft één gecentraliseerde CVIS-DDS server die bekend is bij alle CVIS-DDS cliënten. Het CVIS-DDS systeem is in deze situatie niet gedistribueerd over een netwerk.

4.3 Literatuurstudie OMG-DDS

De literatuurstudie van OMG-DDS (Object Management Group Data Distribution Service) heeft als doel de functionaliteiten van OMG-DDS te kunnen vergelijken met die van het CVIS-DDS systeem. Omdat er meerdere bronnen met informatie over OMG-DDS beschikbaar zijn, zijn er tijdens de selectie van de literatuur eisen gesteld. Er is alleen gebruik gemaakt van literatuur waarin wordt gerefereerd aan officiële OMG DDS standaard documentatie. Daarbij wordt geen informatie gebruikt van bronnen die publiekelijk aan te passen zijn (zoals Wikipedia). Deze eisen zijn voldoende voor het literatuuronderzoek omdat de literatuur alleen de functionaliteiten van OMG-DDS hoeft te omschrijven zodat ik een vergelijking kan maken met de functionaliteiten.

Het Publish and Subscribe mechanisme is een voorbeeld van de overeenkomsten met het CVIS-DDS en OMG-DDS. In de literatuur wordt het Publish and Subscribe mechanisme omschreven als een toevoeging op het message passing model, zoals u hieronder kunt lezen:

"Publish-Subscribe adds a data model to messaging. Publish-Subscribe nodes simply "Subscribe" to data they need and "Publish" information they produce. Messages logically pass directly between the communicating nodes. The fundamental communications model implies both discovery—what data should be sent—and delivery—when and where to send it. This design mirrors time-critical information delivery systems in everyday life including television, radio, magazines, and newspapers. PublishSubscribe systems are good at distributing large quantities of time-critical information quickly, even in the presence of unreliable delivery mechanisms." [SF2006]

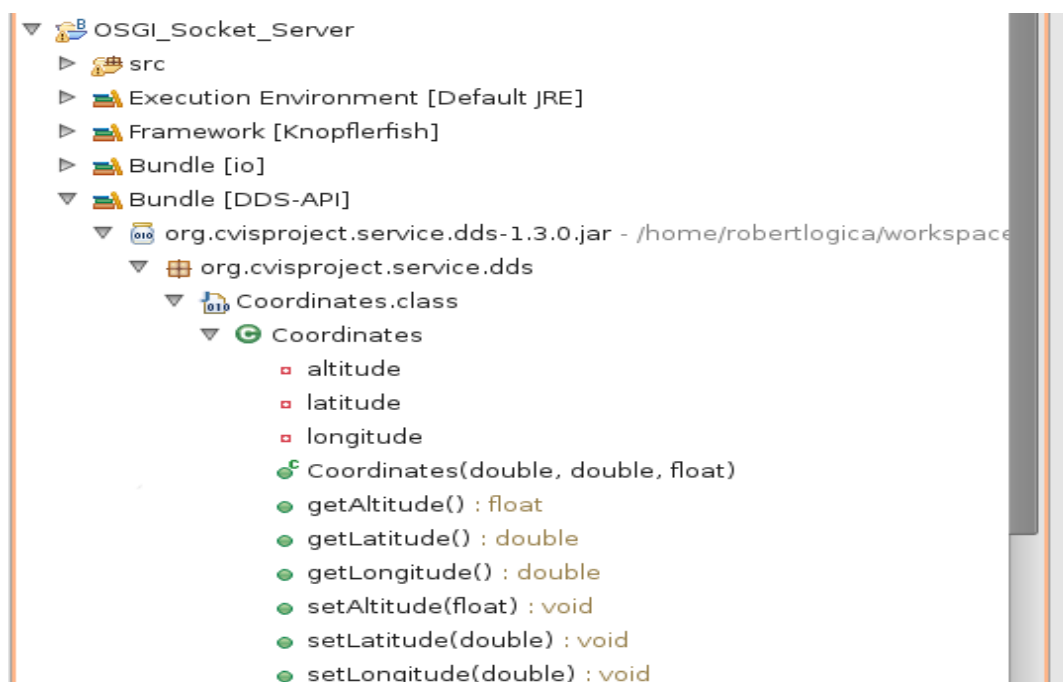
4.4 Vergelijking CVIS-DDS en OMG-DDS

Met de vergelijking tussen CVIS-DDS en OMG-DDS was het de bedoeling om eventuele onbekende functionaliteiten van CVIS-DDS te achterhalen. Er zijn helaas geen nieuwe functionaliteiten of mechanismen van het CVIS-DDS systeem blootgelegd. De overeenkomsten staan hieronder opgesomd:

- Middleware software in een gedistribueerde omgeving;
- Publish and Subscribe – mechanisme voor het publiceren en ontdekken van diensten.

4.5 Klassediagrammen van CVIS-DDS API

Er zijn geen klassediagrammen van de CVIS-DDS API beschikbaar. Om inzicht in het systeem te krijgen zijn er eerst klassediagrammen gemaakt van de CVIS-DDS API. Deze diagrammen heb ik kunnen afleiden uit de Integrated Development Environment (IDE) Eclipse. Door het importeren van de CVIS-DDS API kan door de variabelen en methoden van de API "gebrowse" worden zoals in figuur 3 op de volgende pagina.



Figuur 3: Screenshot Eclipse IDE met CVIS-DDS API

De hierboven weergegeven screenshot geeft de private variabelen altitude, latitude en longitude met de methoden getAltitude tot en met setLongitude weer. Aan de hand van dergelijke informatie kan het hieronder staande UML klassediagram gemaakt worden.

Coordinates
<ul style="list-style-type: none"> - latitude : double - longitude : double - altitude : float
<ul style="list-style-type: none"> + Coordinates(latitude : double, longitude : double, altitude : float) : void + getLatitude() : double + getLongitude() : double + getAltitude() : float + setLatitude(latitude : double) : void + setLongitude(longitude : double) : void + setAltitude(altitude : float) : void

Figuur 4: Coordinates klasse

Figuur 4 op de vorige pagina illustreert het klassediagram van de “Coordinates” klasse uit de CVIS-DDS API weergegeven. In de onderzoeksrapportage kunt u alle gemaakte klassediagrammen van de CVIS-DDS API vinden (zie bijlage X.X). De klassediagrammen zijn gebruikt voor het maken van de applicaties waarmee de functionaliteiten van het CVIS-DDS systeem gevalideerd zijn. Door het maken van de klassediagrammen is meer begrip van het systeem ontstaan. Ook kon de CVIS-DDS API gemakkelijk toegelicht worden in de onderzoeksrapportage door gebruik te maken van de klassediagrammen. Het werd echter snel duidelijk dat de gemaakte klassediagrammen van de CVIS-DDS API niet overeenkwamen met de abstracte omschrijvingen in de literatuur van het CVIS-DDS systeem. In de literatuur komen ook een aantal klassediagrammen voor die het CVIS-DDS systeem illustreren, echter wijken deze af van de daadwerkelijke implementatie. In het volgende paragraaf wordt dit toegelicht.

4.6 Verificatie functionaliteiten CVIS-DDS

In deze paragraaf wordt de opstelling toegelicht waarmee de functionaliteiten geverifieerd zijn. Ook worden de resultaten van de verificatie in deze paragraaf toegelicht.

Gezien de beperkte tijd waarin dit project uitgevoerd diende te worden is het verifiëren van de functionaliteiten en exploreren van de functionaliteiten (zie paragraaf 4.7) uitgevoerd in een timebox van twee weken.

Het valideren van de gedocumenteerde functionaliteiten heeft een hogere prioriteit gekregen. Ik vond het belangrijker om de omschreven functionaliteiten te valideren dan nieuwe functionaliteiten te achterhalen tijdens de exploratie. De kans dat functionaliteiten niet geïmplementeerd of defect zijn wordt groter geacht dan dat er nieuwe functionaliteiten gevonden kunnen worden. Dit ligt overigens ook in de lijn der verwachting van de opdrachtgever.

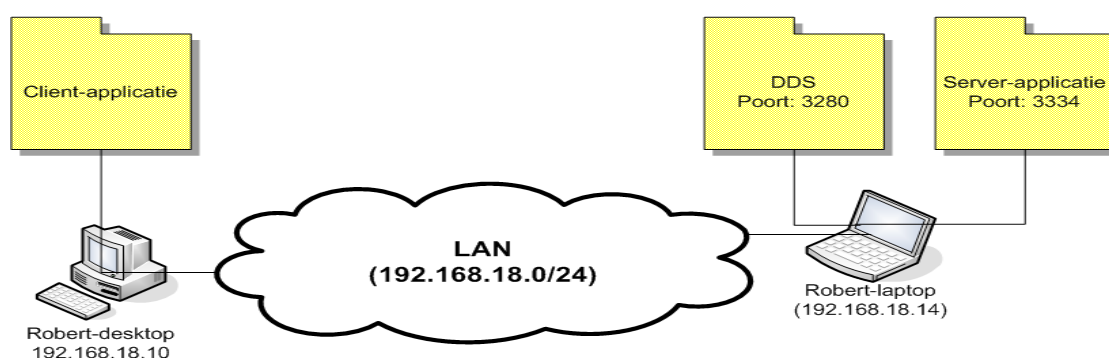
De testopstelling

Met deze testopstelling wordt een aantal functionaliteiten van het CVIS-DDS systeem geverifieerd. De testopstelling is gebouwd naar een gecentraliseerd CVIS-DDS architectuur. Er is bewust gekozen voor deze architectuur omdat de opdrachtgever heeft aangegeven niet geïnteresseerd te zijn in de andere architectuur vormen. Gecentraliseerd CVIS-DDS wordt reeds ook toegepast binnen CVIS.

De testopstelling bestaat uit twee applicaties, elke applicatie draait op een afzonderlijke computer. De eerste applicatie is een cliënt-applicatie (OSGI_Socket_Client genaamd) dat zichzelf registreert bij het CVIS-DDS systeem en een geregistreerde server-applicatie op kan opzoeken in het CVIS-DDS systeem. De cliënt-applicatie draait op de host 'robert-desktop' (192.168.18.10). De tweede ontwikkelde applicatie is de server-applicatie (OSGI_Socket_Server genaamd). De server-applicatie draait op de host "robert-laptop".

Deze applicatie opent een socket op de host waarop de cliënt-applicatie een bericht van de server afhaalt, de cliënt drukt dit bericht af. De server-applicatie registreert zichzelf bij het centrale CVIS-DDS systeem met het betreffende poortnummer en IP-adres van de host.

Onderstaand figuur illustreert de testopstelling. De toelichting van de illustratie volgt na de illustratie zelf.



Figuur 5: Illustratie testopstelling [R2009-onderzoek]

U ziet dat de CVIS-DDS server draait op de host robert-laptop. Omdat de testopstelling opgebouwd is rond gecentraliseerd CVIS-DDS zijn de CVIS-DDS cliënts (deze cliënt applicaties zijn op elke host geïnstalleerd maar niet weergegeven in de illustratie) statisch geconfigureerd met het IP-adres en poortnummer van de server. Het maakt daarom niet uit op welke host de CVIS-DDS server geïnstalleerd is.

Het risico "Onbeschikbaarheid van Central Services center" heeft zich voorgedaan. Door netwerkproblemen was het Central Services center, waar de Logica CVIS-DDS server draait, onbeschikbaar. Ik heb er voor gekozen om plan B van dit risico uit te voeren. Dit betekende dat ik het CVIS-DDS systeem in eigen beheer zou nemen. Ik moest het CVIS-DDS systeem toch zelf een keer installeren om de bevindingen over het "installatiegemak" van het CVIS-DDS systeem te kunnen beschrijven. Het kostte mij dus per saldo geen extra tijd.

De functionaliteiten van het CVIS-DDS systeem zijn gevalideerd. In de tabel hieronder ziet u de uitkomst van deze validatie.

Functionaliteit	Geverifieerd	Resultaat
Registreren en wijzigen 'properties' van een geregistreerd object	Ja.	Succesvol.
Lokalisatie van objecten.	Ja.	Succesvol.
Lokalisatie van objecten op: <ul style="list-style-type: none"> Eigenschappen; Internet; Home netwerk (thuis netwerk); Het lokale netwerk. 	Ja.	Succesvol.
Beperken van de zoekresultaten op: <ul style="list-style-type: none"> Aantal resultaten; Zoektijd; Diepte. 	Ja.	Onsuccesvol, voor meer informatie, zie de "Toelichting: beperken zoekresultaten" dat na deze tabel volgt.
Publiceren van diensten.	Ja.	Niet volledig geïmplementeerd. Voor meer informatie, zie "Toelichting: publiceren van diensten" dat volgt na deze tabel.
Inschrijven op diensten.	Ja.	Niet volledig geïmplementeerd. Voor meer informatie, zie "Toelichting: Inschrijven (Subscriben) op objecten" dat volgt na deze tabel.
Verschillende architecturen vormen: <ul style="list-style-type: none"> Gecentraliseerd CVIS-DDS; CVIS-DDS federatie; CVIS-DDS replicatie. 	Nee.	De opdrachtgever vond deze alternatieve architecturen niet belangrijk. Hij verwacht immers niet dat er een andere architectuur, dan gecentraliseerd CVIS-DDS, nodig zal zijn binnen het CVIS-project. Omdat er een beperkte tijd gereserveerd was voor de iteratieslag naar het onderzoek is een validatie naar het vormen van verschillende architecturen niet uitgevoerd.

Tabel 8: Overzicht verificatie functionaliteiten

Toelichting: beperken zoekresultaten

Zoals u in kunt zien is het beperken van zoekresultaten niet succesvol geverifieerd. Het beperken van de zoekresultaten veroorzaakt een "java.io.NotSerializableException" door een implementatiefout in het CVIS-DDS systeem. Bij het zoeken naar objecten in CVIS-DDS wordt de "search(SAQuery, SearchConstraint)" methode gebruikt. Zolang de SearchConstraint "null" is, werkt de methode prima. Wanneer gebruik gemaakt wordt van de search() methode in combinatie met een SearchConstraint ontstaat er een java.io.NotSerializableException in de "org.cvisproject.service.dds.SearchConstraint" class. De broncode van deze klassen is niet in het bezit van Logica waardoor dit probleem niet opgelost kan worden.

De Logica-technisch expert heeft naar dit probleem gekeken om te verifiëren of het geen programmeerfout door mijn toedoen was. Echter kwam ook hij tot de conclusie dat de SearchConstraint klassen niet “serializable” was. Er is daarbij nog getracht een nieuwe klassen te maken dat de SearchConstraint “extends” en daarbij is java.io.Serializable geïmplementeerd. Helaas liep dit uit op een “remote error”. Samen met de technisch expert is de conclusie getrokken dat het beperken van zoekresultaten defect is.

Er is een e-mail gestuurd naar de makers van het CVIS-DDS systeem (Technolution B.V.). Met de e-mail heb ik getracht meer informatie over het probleem los te krijgen, dit is helaas niet gelukt omdat antwoord is uitgebleven. De e-mail kunt u nalezen in bijlage 4.

Toelichting: publiceren van diensten

Zoals u in tabel 8 kunt zien is het publiceren van diensten niet succesvol geverifieerd. Het Publish and Subscribe mechanisme is helaas niet volledig geïmplementeerd. Het publiceren van de diensten functioneert niet zoals omschreven in de literatuur.

Om een vergelijking tussen de ontwerpen uit [SSBA2009] en de daadwerkelijke implementatie te maken is hieronder de “SADescription” klasse geïllustreerd. Daarna volgt een illustratie van de SADescription volgens de daadwerkelijke implementatie. Na figuur 7 worden de twee illustraties vergeleken.

SADescription	
+	DDS-SAID: Int
+	IsPublish: boolean
+	Mandatory: boolean
-	url: URI

Figuur 6: SADescription [SSBA2009]

SADescription	
-	serialVersionUID : long
-	saType : String
-	url : String
-	properties : Hashtable
+	SADescription(saType : String, url : URL, isPublish : boolean) : void
+	SADescription(saType : String) : void
-	CheckConditions(saType : String) : void
+	getSaType() : String
+	setSaType(saType : String) : void
+	getUrl() : String
+	setUrl(url : URL) : void
+	getUrlString() : String
+	toString() : String
+	getProperties() : Hashtable
+	setProperty(property : Property) : void
+	hashCode() : int
+	equals(obj : Object) : boolean

Figuur 7: SADescription (daadwerkelijke implementatie)

Merk op dat de variabelen niet overeenkomen. De 'isPublish' variabele ontbreekt in de daadwerkelijke implementatie van de CVIS-DDS API. Echter is er wel een constructor aanwezig met de IsPublish (boolean) variabele. Het Subscriben van de OSGI_Socket_Server is getest waarbij gebruik gemaakt is van de SADescription(saType : String, url : URL isPublish : Boolean) constructor.

Het publiceren van een applicatie met de SADescription(saType:String, url:URL, isPublish:Boolean) ging goed; zowel met een “true” als “false” argument voor de “isPublish” variabele. Echter moet het CVIS-DDS systeem continu broadcast berichten versturen op het moment dat het een applicatie gepubliceerd is. Zie het citaat hieronder:

“... The DDS will initiate a continuous broadcast of the presence of the service provided by the registered application (i.e., the identifier and the URI are broadcasted within the transmission range of the corresponding CVIS Unit (eg a Roadside Unit)).”[SSBA2009]

Na het registreren van de applicatie is het netwerk met de netwerk-sniffing applicatie 'Wireshark' gescand op broadcast berichten afkomstig van het CVIS-DDS systeem waarin de applicatie gepubliceerd is. De broadcast berichten zijn niet helaas niet gevonden.

Het ontbreken van de 'isPublish' variabele doet vermoeden dat de functionaliteit niet volledig geïmplementeerd is. En het ontbreken van de broadcast-berichten heeft er toe geleid dat de conclusie getrokken is dat het publiceren van applicaties niet volledig geïmplementeerd is.

Toelichting: Inschrijven (subscriben) op objecten

Objecten kunnen zich bij hun lokale CVIS-DDS cliënt inschrijven op gepubliceerde diensten. De lokale CVIS-DDS cliënt scant dan continu het netwerk naar de gepubliceerde diensten. Zodra een gepubliceerde dienst gelokaliseerd wordt op het netwerk zal de URL doorgegeven worden aan de objecten die zich ingeschreven hebben op de betreffende gelokaliseerde dienst. In de CVIS-DDS API is geen manier gevonden om een applicatie in te schrijven (subscriben).

In de literatuur is omschreven dat het CVIS-DDS systeem gebruik maakt van een onderliggende communicatie infrastructuur voor de “broadcast-and-detect” faciliteit.

“The DDS relies on services provided by the underlying communication infrastructure to provide this broadcast-and-detect facility” [SSBA2009]

Het is onduidelijk op welke functionaliteit gedoeld wordt. In de API van het CVIS-DDS systeem zijn geen referenties gevonden die duiden op de implementatie van de onderliggende communicatie infrastructuur. Uit het niet functioneren van het Publish mechanisme en het ontbreken van een Subscribe mechanisme is de conclusie getrokken dat een dergelijke infrastructuur en het Publish and Subscribe mechanisme niet (volledig) geïmplementeerd zijn.

4.7 Exploratie van nieuwe functionaliteiten CVIS-DDS

In het plan van aanpak is een bottom-up aanpak beschreven waarmee functionaliteiten ontdekt moesten worden. Het doel van deze aanpak is de aanwezigheid en werking van de functionaliteiten van het CVIS-DDS systeem te verifiëren en eventuele nieuwe, niet gedocumenteerde, functionaliteiten te ontdekken. Deze exploratie heeft zich gericht op de CVIS-DDS API. Omdat dit de programmatuur is die gebruikt wordt door de applicaties die gebruik maken van het CVIS-DDS systeem.

Zoals in het vorige paragraaf (paragraaf 4.6) toegelicht is, zijn de verificatie en exploratie in een timebox geplaatst. In de tijd die nog beschikbaar was voor het uitvoeren van de exploratie, is alleen het beperken van de zoekresultaten geëxploreerd. Graag had ik de mogelijkheid van database replicatie op database niveau nader onderzocht. Echter heeft de opdrachtgever aangegeven niet geïnteresseerd te zijn in replicatie.

Zoekresultaten beperken middels de properties

Uit de klassediagrammen die afgeleid zijn van de API viel op dat er veel property constraint klassen waren (PropertyConstraint, PolygonConstraint, LongConstraint, CoordinatesinPolygonConstraint, ProximityConstraint, StringConstraint, BooleanConstraint). De naamgeving geeft al aan dat er iets gelimiteerd kan worden, omdat er in de CVIS-DDS klassen de `search(SAQuery : SAQuery, SearchConstraint: SearchConstraint)` methode voorkomt met een SearchConstraint is het idee gerezen dat de zoekresultaten gefilterd kunnen worden op basis van de properties.

De applicaties die tijdens de validatie gebruikt zijn (OSGI_Socket_Server en OSGI_Socket_Client), zijn aangepast om gebruik te maken van de SearchConstraint; dit werkte zonder problemen. De cliënt-applicatie was in staat om de resultaten te filteren op basis van de TestString (StringProperty) en de TestCoordinates (CoordinatesProperty).

Overzicht functionaliteiten na validatie en exploratie

Hieronder ziet u de functionaliteiten die gevalideerd zijn. De opsomming is inclusief de nieuwe functionaliteiten (dik gedrukt) die bekend zijn geworden tijdens de exploratie. De functionaliteiten zijn:

- Registreren van objecten, met meerdere eigenschappen (properties), dit zijn:
 - String properties;
 - Long properties;
 - Boolean properties;
 - Coordinates properties;
 - Polygon properties;
- Wijzigen 'properties' van een geregistreerd object;
- **Lokalisatie van objecten op eigenschappen (properties), dit zijn:**
 - **String properties;**
 - **Long properties;**
 - **Boolean properties;**
 - **Coordinates properties;**
 - **Polygon properties.**

4.8 Uitloop onderzoeksfase

Tijdens de onderzoeksfase werd duidelijk dat de taken uit de onderzoeksfase niet binnen de gestelde tijd afgerond konden worden.

Er is meer tijd gestoken in het schrijven van het afstudeerverslag dan dat daarvoor gepland was. Ik heb het als erg moeilijk ervaren om de juiste diepgang te beschrijven in dit document, waardoor bepaalde onderdelen veel te diepgaand en te gedetailleerd beschreven zijn.

Er zijn vijf weken voor de onderzoeksfase uitgetrokken, dit was veel te optimistisch. In de onderzoeksfase is de architectuur van het CVIS-netwerk omschreven, een literatuurstudie uitgevoerd en zijn de functionaliteiten van het CVIS-DDS systeem geverifieerd. Voor het verifiëren van de functionaliteiten moest ik kennis en ervaring op doen met de ontwikkelomgeving (de FOAM-omgeving). Achteraf gezien te veel werkzaamheden om uit te voeren in vijf weken tijd.

Nadat duidelijk werd dat de planning niet meer te realiseren was, is een exception rapport naar de project board verstuurd. Dit exception document kunt u terugvinden in bijlage 2. De verbeterde strategie dat in het exception rapport voorgesteld wordt, is een Agile-achtige ontwikkelmethodiek. Agile wordt door mij (en de opdrachtgever) gezien als een geschikte ontwikkelmethodiek om snel en resultaatgericht te werk te gaan. Gezien de beperkte tijd was er behoefte om snel resultaat op te leveren, waardoor een iteratiestrategie in de ontwikkeling van toepassing is. Aan het einde van een iteratie-slag kan immers resultaat getoond worden. Ook de mogelijkheid om goed met veranderende systeemeisen om te gaan heeft bijgedragen aan de keuze voor een Agile-achtige methodiek.

In dit hoofdstuk is het proces van de onderzoeksfase beschreven. In de onderzoeksfase is gebruik gemaakt van de Mindmapping techniek dat gebruikt is om de informatie van de literatuur te structureren.

De functionaliteiten van het CVIS-DDS systeem zijn geverifieerd in dit hoofdstuk daaruit is gebleken dat de implementatie van het CVIS-DDS systeem niet volledig overeenkomt met de literatuur.

Omdat de onderzoeksfase uitgelopen is, ontstond er een behoefte naar een nieuwe strategie, meer over deze strategie in het volgende hoofdstuk.

5 Verbeterde strategie

Er is tijdens de onderzoeksfase voor een nieuwe strategie gekozen om de demonstrator te ontwikkelen. Hier zijn twee redenen voor.

(1) Tijdsdruk, de onderzoeksfase is uitgelopen (zie paragraaf 4.8).

(2) De druk om iets op te leveren, de opdrachtgever heeft zijn zorgen geuit over de oplevering van de demonstrator omdat het project achter liep.

Gezien de tijd en de druk om resultaat op te leveren is in overleg met de architect besloten om de keuze voor een specifieke Agile ontwikkelmethodiek te beperken tussen enkele aspecten van Extreme Programming (XP) en Scrum. De keuzemogelijkheden zijn beperkt tot XP en Scrum omdat dit bekende en gemakkelijk toe te passen methodieken zijn.

Uiteindelijk is besloten om gebruik te maken van de Scrum methode. XP is meer een software ontwikkelmethodiek terwijl Scrum meer een procesbeschrijving is. Zo schrijft XP de programmeertechniek 'pair-programming' voor, waar niet aan voldaan kan worden omdat dit project door één persoon wordt uitgevoerd. Daarbij schrijft XP test-driven programmeren voor. Omdat er geen betrouwbaarheidseisen aan de demonstrator gesteld zijn, wordt het test-driven programmeren zoals XP voorschrijft niet toegepast; ook vind ik test-driven programmeren te omslachtig voor dit project. XP is dus niet geschikt voor dit project, er is dus voor de Scrum methodiek gekozen.

Omdat er veel literatuur bestaat dat Scrum beschrijft, konden eisen aan de literatuur gesteld worden. Die eisen staan hieronder opgesomd:

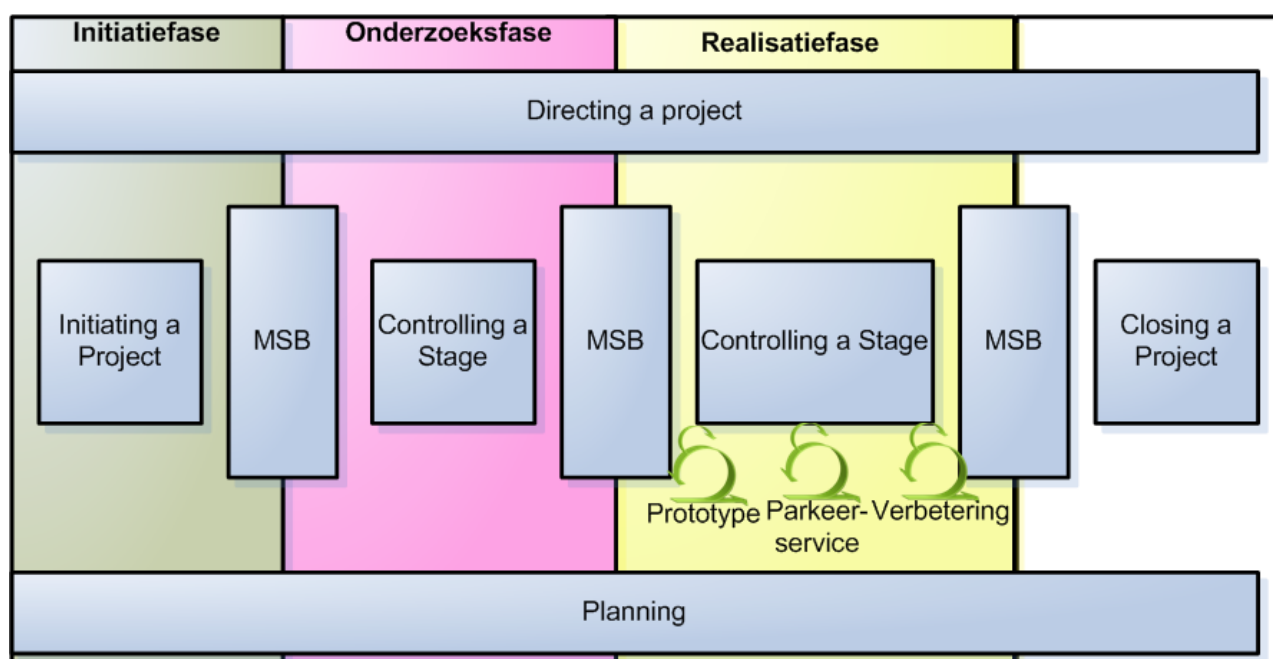
A. De literatuur moet toegankelijk zijn zodat de methodiek gemakkelijk en snel toe te passen is. Dit in verband met de tijdsdruk.

B. De beschrijving van de methode mag niet publiekelijk aan te passen zijn (zoals Wikipedia).

Er wordt ontwikkeld in meerdere iteratieslagen (sprints). Aan het begin van deze sprints worden de ontwerpen van de software gemaakt. De gehanteerde bronnen waarop de methodiek is gebaseerd schrijven voor dat er tijdens de iteratieslagen ontworpen wordt en dat die ontwerpen aangepast worden gedurende het ontwikkel-proces.

"This means keeping the design simple from start and continuously improving it, rather than trying to get it all right from the start and then freezing it." [K2007]

Omdat er ontworpen wordt tijdens de iteratieslagen (realisatiefase), zijn de taken van de ontwerpfase gecombineerd met de realisatiefase zoals u in figuur 8 kunt zien. De ontwerpfase komt in dat figuur niet terug. Een toelichting van de illustratie volgt na het figuur.



Figuur 8: Nieuwe projectfasering met Prince 2 processen

Het figuur hierboven illustreert de nieuwe projectfasering. De ontwerpfase ontbreekt in de illustratie, de taken van deze fase worden uitgevoerd tijdens de sprints. In de realisatiefase ziet u drie scrum icoontjes. Elk icoontje staat voor een scrum sprint die uitgevoerd is tijdens dit project. In paragraaf 5.2 is het doel van de sprints beschreven.

5.1 De aanpak van de sprints

In deze paragraaf is de aanpak van de sprints beschreven. Hier staat beschreven hoe de de sprints ingedeeld zijn. De daadwerkelijk procesverslaggeving is beschreven in hoofdstuk 6.

Product backlog

Tijdens het uitvoeren van de Scrum sprints wordt een werkdocument bijgehouden. Dit document draagt de toepasselijke naam: "Scrum werkdocument". In dit document is alles beschreven wat met het realiseren van de software te maken heeft.

Een belangrijk onderdeel van dit werkdocument is het product backlog. In hoofdstuk twee van het Scrum werkdocument staat de product backlog met daarin de "stories". Stories is een Scrum term voor een beknopte omschrijving van een functionaliteit die gerealiseerd wordt. In het volgende hoofdstuk, waar de realisatiefase beschreven is, zijn delen van de product backlog weergegeven en verder toegelicht.

Ontwerpen

De use-cases zijn gebaseerd op de omschrijving van de functionaliteiten in het product backlog. Hoewel samen met de opdrachtgever de functionaliteiten in het product backlog beschreven zijn vond ik het belangrijk om met de opdrachtgever de te realiseren functionaliteiten voor een tweede maal te bespreken, maar dan met een (voor hem) bekende modelleertaal. Zo werd er nogmaals goed nagedacht over de te realiseren functionaliteiten van de demonstrator.

Zodra de use-cases en use-case diagrammen afgerond zijn, worden er klassediagrammen gemaakt van de te ontwikkelen software. De klassediagrammen worden gemaakt na het maken van use-cases omdat dan enige zicht is op de te realiseren functionaliteiten. Op basis van deze kennis worden de functionaliteiten gegroepeerd worden in verschillende klassen.

In het geval wanneer het interne gedrag van het systeem complex is, is een sequentiediagram opgesteld. Hiermee maak ik het gedrag van het systeem inzichtelijk zodat deze gemakkelijk en snel geïmplementeerd kan worden.

Testcases opstellen

Het doel van de demonstrator is de werking van het CVIS-DDS systeem aantonen. Om aantoonbaar te maken dat de demonstrator haar doel behaalt zijn acceptatietesten gemaakt. Acceptatietesten passen goed bij de gehanteerde ontwikkelstrategie. In de Product Backlog zijn de functionaliteit verhalenderwijs beschreven dit wordt in Scrum termen zogeheten “stories” genoemd. Deze stories zijn gericht op het omschrijven van de functionaliteiten en zijn daarom gemakkelijk naar testcases te vertalen. Zodoende wordt de werking van de functionaliteiten van het CVIS-DDS systeem middels de testcases in de acceptatietest aangetoond. Ook in de literatuur van de Agile methoden wordt regelmatig verwezen naar acceptatietesten. In literatuur van Extreme Programming staat het volgende citaat:

“Acceptance tests are created from user stories. During an iteration the user stories selected during the iteration planning meeting will be translated into acceptance tests. The customer specifies scenarios to test when a user story has been correctly implemented. A story can have one or many acceptance tests, what ever it takes to ensure the functionality works. ”
[WWXP]

Implementatie van de ontwerpen

Zodra de testcases af zijn wordt de software geïmplementeerd op basis van de klasse- en sequentiediagrammen.

Uitvoeren van de testen

Aan het einde van de sprint wordt een demonstratie gegeven van de geïmplementeerde functionaliteiten. Na de demonstratie zullen de acceptatietesten doorgenomen worden. Dat wordt gedaan met de project board, zij moeten de demonstrator immers goedkeuren.

Als extra (blackbox) test is gevraagd aan mijn collega's en de technisch expert om de systemen te onderwerpen aan een "explorerende test". Daarbij wordt het systeem getest door iemand die enigszins begrip heeft van het systeem (de technisch expert). Door het uitvoeren van de blackbox test kunnen meer bugs gevonden worden omdat er met een andere "blik" naar de demonstrator gekeken wordt en een andere insteek wat het testen betreft gehanteerd wordt.

Scrum-meetings

Er zijn een drietal soorten Scrum-meetings. Dit zijn de 'Daily Scrum', 'Scrum of scrums' en Sprint planning meeting.

Omdat dit project niet uit meerdere teams bestaat is de Scrum of Scrums – meeting niet nodig. Om toch het doel van de daily scrum te behalen, het identificeren en uit de weg ruimen van eventuele vertragende factoren, hou ik een dagelijks logboek bij.

In het log behandel ik de volgende vragen:

- Wat is er gebeurd sinds de laatste beschrijving in het logboek?
- Wat wordt er gedaan in de tijd tussen nu en de volgende beschrijving in het logboek?
- Is er iets dat de werkzaamheden kan belemmeren?

Dit zijn de vragen die ook in een Daily Scrum meeting ook behandeld worden.

5.2 De sprints

In deze paragraaf is het doel van de uitgevoerde sprints toegelicht. In hoofdstuk 6 wordt het proces toegelicht. Elke sprint heeft een naam gekregen zoals hieronder is beschreven:

1. Sprint 1 → prototype sprint;
2. Sprint 2 → Parkeer-systeem sprint;
3. Sprint 3 → Parkeer-systeem verbetering sprint.

Het aantal iteratieslagen (sprints) is gebaseerd op het doel van de demonstrator en de eisen die daaraan gesteld zijn.

De demonstrator heeft als doel de functionaliteiten van het CVIS-DDS systeem aan te tonen.

Hoewel dit al aangetoond is met de resultaten uit de onderzoeksfase bestaat er nog geen echte demonstrator.

Daarnaast zijn de eisen gesteld dat het systeem het nut van het CVIS-DDS systeem moest aantonen en tot de verbeelding moest spreken.

Het doel van de demonstrator (aantonen functionaliteiten) heb ik willen realiseren in één sprint met als resultaat een prototype. De eisen (aantonen nut en tot de verbeelding spreken) wilde ik realiseren in een tweede sprint. Daarnaast wilde ik nog één sprint besteden aan het oplossen van eventuele problemen en anders een uitbreiding van de demonstrator.

Ik heb de inschatting gemaakt dat de werkzaamheden van de sprints in ongeveer één week te realiseren zijn.

Sprint 1 (Prototype sprint)

De eerste sprint had als doel een prototype dat de werking van het CVIS-DDS systeem aantoont op te leveren. Met een dienst aanbiedende applicatie (Aanbieder-applicatie) en een diensten afnemende applicatie (Afnemer-applicatie) wordt de belangrijkste werking van het CVIS-DDS systeem aangetoond.

Met de prototype worden de volgende functionaliteiten van het CVIS-DDS systeem aangetoond:

- Registreren van objecten, met meerdere eigenschappen (properties);
- Lokalisatie van objecten, op basis van eigenschappen (properties).

De Aanbieder-applicatie beschikt over de mogelijkheid om een beknopte tekst in te voeren. Deze tekst wordt door de Afnemer-applicatie afgedrukt, zo kan onomstotelijk bewezen worden dat berichten die de Afnemer-applicatie afdruckt in de GUI afkomstig zijn uit de Aanbieder-applicatie. En dat er netwerk-communicatie plaatsvindt tussen de twee applicaties. Deze Aanbieder-applicatie is via CVIS-DDS gelokaliseerd en er is een communicatie-sessie tussen de Afnemer- en Aanbieder-applicatie opgesteld.

Meer informatie over de prototype sprint kunt u vinden in paragraaf 6.1.

Sprint 2 (Parkeer-systeem sprint)

Het doel van deze sprint is het ontwikkelen van de Parkeer-systeem demonstrator bestaande uit een Parkeer-service en Parkeer-applicatie. Deze sprint breidt de functionaliteiten en werking van de Aanbieder en Afnemer applicatie uit. De eerder ontwikkelde prototype verifieert alleen de werking van het CVIS-DDS. Het nut en voordeel van CVIS-DDS wordt niet aangetoond door het prototype; de Parkeer-service doet dit wel omdat het meer tot de verbeelding spreekt. Meer informatie over de Parkeer-service sprint kunt u vinden in paragraaf 6.2.

De demonstrator die opgeleverd is tijdens deze sprint toont aan dat de volgende functionaliteiten van het CVIS-DDS systeem werken:

- Registreren van objecten, met meerdere eigenschappen (properties);
- Wijzigen 'properties' van een geregistreerd object;
- Lokalisatie van objecten, op basis van eigenschappen (properties).

Sprint 3 (Parkeer-systeem verbetering sprint)

Het doel van deze sprint is verbeteringen aan het Parkeer-systeem aan te brengen en deze uit te breiden. De uitbreidingsmogelijkheden zijn het lokaliseren van objecten op basis van andere eigenschappen (properties) en het oplossen van enkele bugs.

In dit hoofdstuk is de vernieuwde strategie waarmee de demonstrator ontwikkeld is toegelicht. De nieuwe strategie is gebaseerd op Scrum en bevat drie iteratieslagen waarbinnen de demonstrator ontwikkeld is. Het volgende hoofdstuk beschrijft het ontwikkel-proces van de demonstrator.

6 De realisatie fase

Dit hoofdstuk beschrijft het proces van de realisatiefase waarin de demonstrator ontwikkeld is. De demonstrator is ontwikkeld in drie iteratieslagen (sprints) dit zijn: De prototype sprint, Parkeer-systeem sprint en de Parkeer-systeem verbetering sprint.

6.1 Prototype sprint

De prototype sprint heeft als doel een demonstrator op te leveren waarin het registreren van objecten in CVIS-DDS wordt aangetoond en dat volgende functionaliteiten van het CVIS-DDS systeem aantoont:

- Registreren van objecten, met meerdere eigenschappen (properties), dit zijn:
 - String properties;
- Lokalisatie van objecten op eigenschappen (properties), dit zijn:
 - String properties;

Om dit doel te bereiken is een tweetal applicaties ontwikkeld tijdens de prototype sprint. Dit is de Aanbieder-applicatie en de Afnemer-applicatie. De Aanbieder-applicatie registreert zichzelf bij CVIS-DDS. De Afnemer-applicatie die neemt een dienst af van de Aanbieder-applicatie nadat deze de Aanbieder-applicatie gelokaliseerd heeft in het CVIS-DDS systeem. Deze twee applicaties zijn gebouwd om aan te tonen dat een service via CVIS-DDS gelokaliseerd kan worden waarbij de applicaties met elkaar communiceren.

Elke applicatie draait op een afzonderlijke computer. De afnemer-applicatie kan beschouwd worden als een zogeheten cliënt-applicatie. De Aanbieder-applicatie kan gezien worden als een server-applicatie. De Aanbieder-applicatie registreert zichzelf bij het CVIS-DDS systeem waardoor deze gelokaliseerd kan worden.

De Aanbieder-applicatie opent ook een socket om communicatie over het netwerk mogelijk te maken. De functionaliteiten dat het systeem aanbiedt is een tekst die (lokaal in de Aanbieder-applicatie zelf) gewijzigd kan worden. Deze tekst wordt door de Afnemer-applicatie afgedrukt. Op deze manier wordt aangetoond dat deze een dienst afneemt van de Afnemer-applicatie en kan bewezen worden dat de tekst die afgedrukt wordt niet hard-coded in de Afnemer-applicatie zit. Voordat de Afnemer-applicatie kan communiceren met de Aanbieder-applicatie moet het de betreffende URL van de Aanbieder-applicatie achterhalen uit het CVIS-DDS systeem.

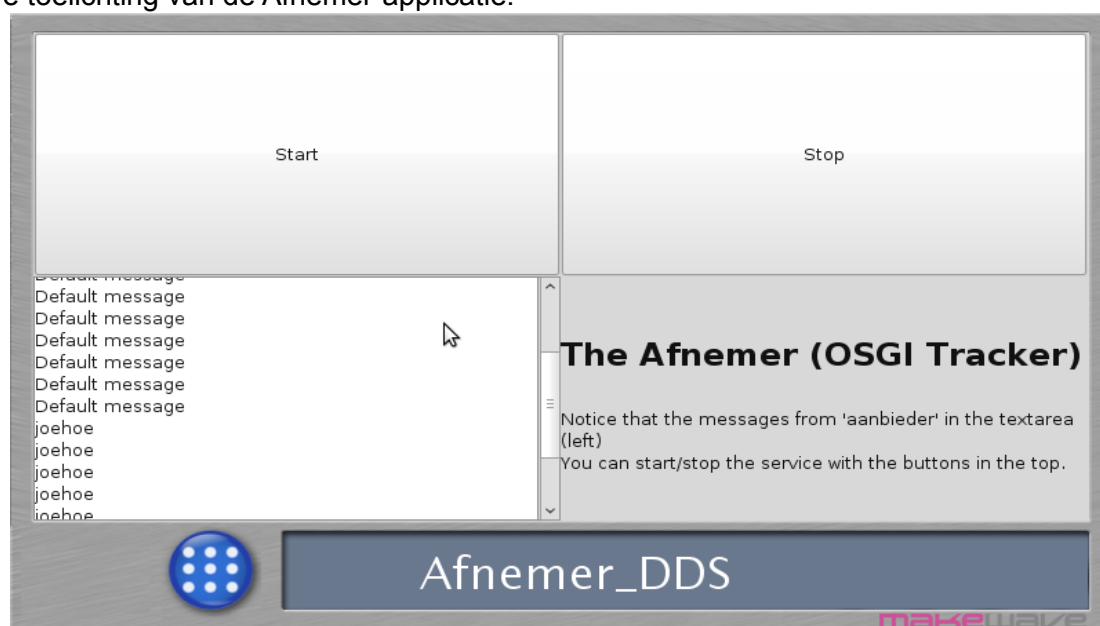
In het figuur hieronder ziet u de GUI van de Aanbieder-applicatie. Onder het figuur is de GUI toegelicht.



Figuur 9: Screenshot Aanbieder-applicatie

In figuur 9 staat de GUI van de Aanbieder-applicatie. In de Aanbieder-applicatie kan het bericht aangepast worden, deze staat standaard op “Default message”. In figuur 9 ziet u dat het bericht is gewijzigd naar “joehoe”. Vanaf het moment dat er op de “Oke”-knop is gedrukt zal de Afemer-applicatie de nieuwe tekst “joehoe” afdrukken.

Na deze alinea, in figuur 10, is de GUI van de Afemer-applicatie weergegeven. Daarin kunt u zien dat de Afemer-applicatie ook daadwerkelijk de tekst “joehoe” afdrukt. Na de illustratie volgt een verdere toelichting van de Afemer-applicatie.



Figuur 10: Screenshot Afemer-applicatie

De taken die tijdens deze sprint uitgevoerd zijn, zijn afkomstig uit het Scrum product backlog. In de tabel hieronder ziet u de eerste twee regels van de product backlog uit het Scrum werkdocument.

ID	Naam	Prio	Tijd Schatting	Hoe te demonstreren
1	Registreren van objecten in gecentraliseerde CVIS-DDS-omgeving.	10	2 dagen	Start bundel: 'Aanbieder_DDS_CENTRAL' op de roadside unit. Open de GUI van Aanbieder-applicatie. Controleer vervolgens of deze de melding geeft dat deze zichzelf geregistreerd heeft.
2	Lokalisatie van objecten in gecentraliseerde CVIS-DDS omgeving.	10	2 dagen	De Afnemer_DDS_CENTRAL bundel in de auto starten en controleren of deze de tekst van de Aanbieder_DDS_CENTRAL weergeeft terwijl de auto stil naast de roadside unit staat. Verander vervolgens de tekst in de Aanbieder_DDS_CENTRAL en controleer of de Afnemer_DDS_CENTRAL deze nieuwe tekst afdruckt.

Tabel 9: Deel de product backlog [R2010-SWD]

De twee taken die in tabel 9 staan, zijn uitgevoerd tijdens deze sprint. De taken hebben de hoogste prioriteit en zijn daarom als eerste geïmplementeerd. Hoewel deze functionaliteiten van het CVIS-DDS systeem al gedemonstreerd waren in een testopstelling tijdens de onderzoeksfase, zijn ze tijdens deze sprint beter ontworpen en is er een GUI gemaakt.

Op de volgende pagina staat het sprint backlog, dit is de detail planning van de sprint. De werkzaamheden die in het sprint backlog beschreven zijn, zijn afkomstig uit de product backlog. Daarnaast zijn de aanvullende taken, zoals het ontwerpen en opstellen van het testplan opgenomen in het sprint backlog. De dagen zijn opgesplitst in twee dagdelen, de ochtend en de middag. In tabel 10 (op de volgende pagina) zijn de taken verspreid over de dagdelen.

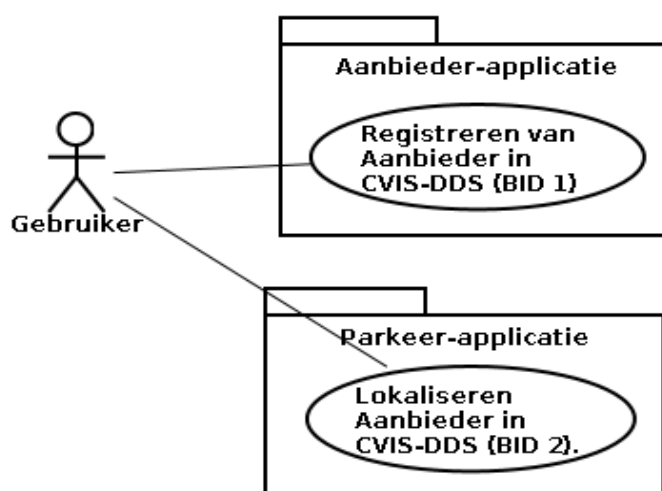
Dagdeel	Te verrichte taken	Status
Dag 1 ochtend	Ontwerpen Aanbieder en Afnemer-applicatie Testcases opstellen	Afgerond
Dag 1 middag	Ontwerpen Aanbieder en Afnemer-applicatie Testcases opstellen	Afgerond
Dag 2 ochtend	Activator klasse bouwen (Aanbieder-applicatie)	Afgerond
Dag 2 middag	Netwerk interface klasse bouwen waar het lokale IP mee achterhaald wordt	Afgerond
Dag 3 ochtend	IOServer klasse bouwen (Aanbieder)	Afgerond
Dag 3 middag	Protocol klasse bouwen (Aanbieder) DDSTracker klassen bouwen (Aanbieder)	Afgerond
Dag 4 ochtend	View klasse bouwen	Afgerond
Dag 4 middag	View klasse bouwen	Afgerond
Dag 5 ochtend	IOClient klasse bouwen	Afgerond
Dag 5 middag	View klasse (Afnemer) ombouwen Testen en demonstreren systeem	Afgerond

Tabel 10: Detailplanning prototype sprint [R2010-SWD]

Ontwerpen

Voorafgaand aan het implementeren van de programmatuur zijn tijdens de sprint eerst ontwerpen gemaakt. Deze ontwerpen bestaan uit use-cases, use-case diagrammen, klassediagrammen en enkele sequentiediagrammen. Hierna worden enkele resultaten van het ontwerpen behandeld. Om de grootte van dit verslag te beperken zijn niet alle gemaakte onderdelen van de ontwerpen in dit verslag beschreven.

Hieronder ziet u het use-case-diagram van de functionaliteiten die ontwikkeld worden tijdens de demonstrator sprint. In het Scrum werkdocument staan de use-cases.



Figuur 11: Use-case diagram Prototype sprint

In de demonstrator is gebruik gemaakt van de `org.osgi.service.io.ConnectorService` interface voor het openen van sockets en connecties tussen de cliënt en server applicaties, er is dus geen gebruik gemaakt van de standaard Java Unix sockets. Door gebruik te maken van de `ConnectorService` interface kan gemakkelijk een socket opgebouwd worden. Het OSGi-platform zoekt automatisch voor een geschikte Connection Factory service om een socket te openen of een connectie te starten.

Door gebruik te maken van de `ConnectorService` hoeft de programmatuur niet aangepast te worden om te draaien op het CVIS FOAM-platform. FOAM maakt namelijk gebruik van de CALM manager (Continuous Air-interface Long and Medium range). Een systeem dat verschillende soorten netwerk verbindingen zoals Wifi en M5 beheert. Om hiervan gebruik te kunnen maken moet de `org.osgi.service.io.ConnectorService` interface gebruikt worden.

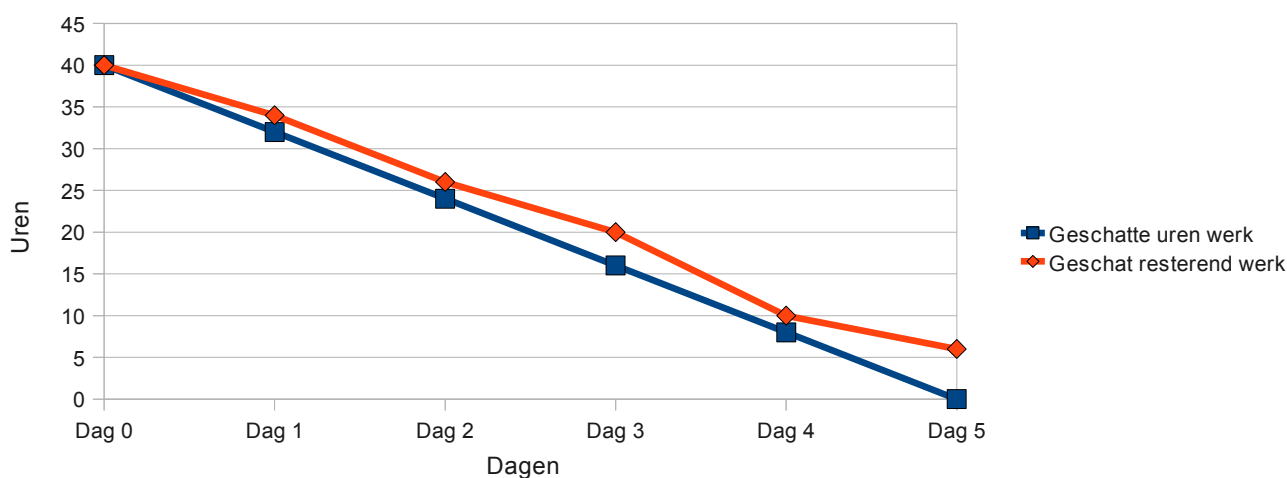
Testresultaten prototype sprint

Na het uitvoeren van de testen heb ik de conclusie getrokken dat de testcases niet specifiek en reproduceerbaar waren. In de hierop volgende sprints heb ik meer aandacht besteed aan het specifiek maken reproduceerbaar maken van de testen.

Tijdens de explorerende test kwam de technisch expert het volgende probleem tegen: Wanneer de Aanbieder-applicatie gestopt wordt terwijl de Afnemer-applicatie draait, dan gaat de Afnemer-applicatie hier niet goed mee om en ontstaat er een "IOClientConnection Error" waardoor de applicatie crasht. Dit is een probleem dat tijdens de volgende sprint is opgelost.

Burndown grafiek

De grafiek hieronder toont middels de donker blauwe lijn het vooraf geschatte aantal uren werk aan en de rode lijn toont het geschatte aantal uren resterende werk aan op het moment dat een daily sprint afgerond is.



Figuur 12: Burndown chart prototype sprint

Tijdens het plannen van deze eerste sprint werd al duidelijk dat er weinig tijd beschikbaar was voor het afronden van de taken. Uit de burndown grafiek uit figuur 12 kunt u afleiden dat aan het einde van de sprint nog een aantal uur werk over was. De werkzaamheden van deze sprint zijn alsnog afgerond. De afspraak om de resultaten van de prototype sprint te demonstreren is door de opdrachtgever verplaatst. Omdat ik erg gemotiveerd was om de demonstrator af te ronden heb ik de resterende werkzaamheden afgerond door over te werken.

Na de demonstratie van de demonstrator is met de opdrachtgever ook een acceptatie- en exploratie-test uitgevoerd. De resultaten zijn na deze alinea beschreven.

6.2 Parkeer-systeem sprint

Het doel van deze sprint is het nut van gecentraliseerd CVIS-DDS aan te tonen middels een systeem dat tot de verbeelding van de opdrachtgever spreekt. Dit gebeurt middels een geautomatiseerd parkeersysteem. Dit parkeersysteem wijst de automobilist een parkeerplaats toe.

Met de resultaten van de Parkeer-systeem sprint is de werking van de volgende functionaliteiten van het CVIS-DDS systeem aangetoond:

- Registreren van objecten, met meerdere eigenschappen (properties), dit zijn:
 - String properties;
 - Long properties;
 - Boolean properties;
 - Coordinates properties;
- Wijzigen 'properties' van een geregistreerd object;
- Lokalisatie van objecten op eigenschappen (properties), dit zijn:
 - Long properties;
 - Boolean properties;
 - Coordinates properties.

Er is sprake van een cliënt-server architectuur. Een cliënt-applicatie in de auto om een parkeerplaats aan te vragen bij de Parkeer-service en een Parkeer-service dat een parkeerplaats toewijst. De cliënten en servers hebben een veel op veel relatie; er zijn immers meerdere parkeergarages en meerdere automobilisten.

Het CVIS-DDS systeem biedt de mogelijkheid voor parkeergarages om hun Parkeer-service (server) te registreren in het CVIS-DDS systeem. Door de services te registreren in het CVIS-DDS systeem kunnen deze services gelokaliseerd worden door de Parkeer-applicaties (cliënten) van de automobilisten. Hiermee wordt het lokaliseren van parkeergarages een geautomatiseerd proces.

Het CVIS-DDS systeem geeft de parkeergarages de mogelijkheid om eigenschappen van de parkeergarage te geven en bij te werken. Bij eigenschappen kunt u denken aan het aantal vrije parkeerplaatsen (Long), locatie (Coordinates), geopend/gesloten (Boolean), Bedrijfsnaam (String) enzovoorts. Op basis van dergelijke eigenschappen kan een automobilist de gewenste parkeergarage selecteren.

In het figuur 13 hieronder staat een screenshot van de Parkeer-service server applicatie. De toelichting volgt na het figuur.



The screenshot shows the ParkeerService application interface. It is divided into two main sections: 'Garage info:' and 'Systeem info:'. The 'Garage info:' section contains several input fields: 'Bedrijfsnaam:' (a dropdown menu showing 'Q-park Laakhaven'), 'Garagenaam:', 'Straat:', 'Huisnummer:', 'Postcode:', 'Stad:', 'Aantal parkeerplaatsen:', 'Bezette parkeerplaatsen:', and 'Parkeergarage open:' (a checkbox that is checked). The 'Systeem info:' section displays system details: 'URL: 192.168.18.10:3334', 'Coordinaten: 52.07 4.32', and 'Vrije parkeerplaatsen: 1398'. At the bottom, there are three buttons: 'Stop', 'De-registreer', and 'Update'. Below the buttons is a blue circular icon with white dots and a large blue banner with the text 'ParkeerService'. The 'makewave' logo is visible in the bottom right corner.

Figuur 13: Screenshot Parkeer-service versie 0.1

In de GUI van de Parkeer-service kan de Q-park medewerker het systeem registreren in het CVIS-DDS systeem. De Parkeer-service die afgebeeld is in figuur 13 is reeds geregistreerd. De status kan afgeleid worden van de middelste knop "De-registreer". De tekst van de knop verandert van "Registreer" naar De-registreer op het moment dat de applicatie zichzelf geregistreerd heeft bij het CVIS-DDS systeem.

Er is geen verplichting om de Parkeer-service te registreren in het CVIS-DDS systeem. Het systeem kan ook draaien zonder de registratie, echter kunnen dan alleen Parkeer-applicaties de betreffende Parkeer-service benaderen die over een "hard-coded" URL van de Parkeer-service beschikken.

De Q-park medewerker kan de betreffende garage kiezen middels een pulldown menu. Dit vermindert de kans op invoer-fouten. Alle adresgegevens, het totaal aantal parkeerplaatsen, de URL en de coördinaten zijn geregistreerd in het CVIS-DDS systeem.

De automobilisten beschikken over een Parkeer-applicatie. Een screenshot van deze applicatie ziet u hieronder. De toelichting volgt na de afbeelding.

Zoek:

DDS gebruiken: ☒

Ik ben in:

Groningen
▼

Dichtstbijzijnde garage

Verlaat garage

Garage:

Bedrijfsnaam: Q-park

Garagenaam: Laakhaven

Adres

plaats: Den Haag

Straat: Laakhaven

huisnummer: 27

Postcode: 2516ED

Open/gesloten: Geopend

Parkeerplaats: 1



ParkeerApp

makewave

Figuur 14: Screenshot Parkeer-applicatie versie 0.1

In het figuur hierboven ziet u de Parkeer-applicatie zoals de automobilist deze ziet in de auto. Middels het CVIS-DDS systeem is de parkeergarage “Q-park” in Laakhaven gelokaliseerd. De garage-gegevens staan aan de rechterzijde van de applicatie onder “Garage.” De automobilist heeft een parkeerplaats toegewezen gekregen; Dit is parkeerplaats “1”.

Omdat de demonstrator een GPS-module ontbeert moet de gebruiker zelf zijn of haar locatie opgeven voor het zoeken van de dichtstbijzijnde parkeergarage. Daarbij heeft de automobilist de keuze om wél of géén gebruik van het CVIS-DDS systeem te maken. Wanneer er geen gebruik gemaakt wordt van het CVIS-DDS systeem maakt de Parkeer-applicatie zelf een keuze uit één van de drie “hard-coded” parkeergarages.

Wanneer er wél gebruik van het CVIS-DDS systeem wordt gemaakt, lokaliseert het CVIS-DDS de dichtstbijzijnde parkeergarage. Het CVIS-DDS systeem kiest dan voor de dichtstbijzijnde parkeergarage uit de geregistreerde Parkeer-services in de CVIS-DDS database.

De taken die tijdens deze Parkeer-service sprint uitgevoerd zijn, zijn afkomstig uit het product backlog van het Scrum-werkdocument. Hieronder ziet u in tabel 11 de betreffende taken. De toelichting volgt na de tabel.

ID	Naam	Prio	Tijd Schatting	Hoe te demonstreren
...				
19	Bug: Crash in Afnemer-applicatie bij afsluiten Aanbieder	10	1 dag	Stop de Aanbieder-applicatie terwijl de Afnemer-applicatie nog draait en gebruik maakt van de dienst. Controleer of Afnemer stopt met het afdrukken van de tekst uit Aanbieder-applicatie en in het CVIS-DDS systeem opnieuw opzoek gaat naar een Aanbieder-applicatie.
9	Configureren van Parkeer-service	10	½ dag	De Q-park medewerker opent de Parkeer-service server op de roadside unit. Vervolgens configureert de medewerker de volgende parameters: <ul style="list-style-type: none"> • Bedrijfsnaam parkeergarage; • Naam parkeergarage; • Locatie parkeergarage; • Parkeergarage geopend/gesloten; • Totaal aantal parkeervakken; • Aantal bezette parkeerplaatsen wijzigen. Zodra bovenstaande parameters volledig geconfigureerd zijn bevestigt de medewerker de configuratie door op de "OKE" knop te drukken.
10	Registeren Parkeer-applicatie in CVIS-DDS	10	½ dag	(1) De Q-Park medewerker opent de GUI van de Parkeer-service. (2) De Q-Free parkeergarage medewerker klikt op de knop: "registreer service" en (3) controleert in de melding of de registratie succesvol verlopen is.
11	Dichtstbijzijnde parkeergarage opzoeken	9	1 dag	De automobilist start de Parkeer-applicatie in zijn auto. De gebruiker kiest voor een zoekmethode, dichtstbijzijnde parkeergarage, in dit geval. Vervolgens zoekt het CVIS-DDS systeem een aantal dichtstbijzijnde parkeergarages. Uit de resultaten kiest de Parkeer-applicatie de gewenste parkeergarage. Parkeergarages zonder vrije plaatsen worden niet weergegeven.
12	Wijzigen aantal beschikbare parkeerplaatsen	8	½ dag	Een V.I.P (Madonna) inclusief lijfwachten brengt een bezoek aan de Haagse Hogeschool. Een aantal parkeerplaatsen moeten gereserveerd worden zodat de V.I.P en lijfwachten probleemloos kunnen parkeren. De Q-Park medewerker opent de GUI en reduceert het aantal beschikbare parkeerplaatsen. De Q-Park medewerker controleert of het systeem bevestigt dat het de eigenschappen (properties) van zijn registratie in CVIS-DDS bijgewerkt heeft.
...				

Tabel 11: Deel product backlog [R2010-SWD]

Het totaal aantal geschatte dagen geschatte werk komt uit op 3 ½ dag. In de sprint zijn ook nog use-cases, sequentie- en klassediagrammen gemaakt. Deze diagrammen kunt u terugvinden in het Scrum werkdocument. De ontwerp-werkzaamheden en het maken van een testplan zijn niet meegenomen in de tijd van de taken voor de sprint. Het totale aantal dagen werk voor deze sprint komt naar mijn schatting uit op vijf.

De bug “Crash in Afnemer-applicatie bij afsluiten Aanbieder”, die gevonden is in de vorige sprint, is tijdens de hierop volgende sprint opgelost. Het oplossen van de bugs met een hoge prioriteit tijdens de volgende sprint lijkt mij de beste oplossing omdat er dan geen momentum verloren gaat in de overgang van sprint één naar sprint twee. Bovendien valt het aantal bugs mee. In de gebruikte Scrum literatuur wordt de aanpak van bugs, zoals in het citaat hieronder is geciteerd, ook toegepast.

“This is our preferred approach. Right now at least.

Basically, when we finish a sprint we move on to the next one. But we expect to be spending some time in the next sprint fixing bugs from the last sprint. If the next sprint gets severely damaged because we had to spend so much time fixing bugs from the previous sprint, we evaluate why this happened and how we can improve quality ” [K2007]

Testresultaten Parkeer-systeem sprint

Hieronder staat één van de testcases die voorafgaand aan het implementeren van de demonstrator gemaakt zijn. Na de testcase volgen de testresultaten.

“Test 1: Configureren van Parkeer-service

(1) Open het programma Parkeer-service in het dashboard van de road-side unit.

(2) Configureer de volgende velden met de informatie:

- **Bedrijfsnaam parkeergarage:** Q-park;
- **Naam parkeergarage:** Laakhaven;
- **Locatie parkeergarage:**
 - **Straat:** Rijswijkseweg;
 - **Huisnummer:** 27;
 - **Postcode:** 2516ED;
 - **Plaats:** Den Haag;
- **Parkeergarage geopend/gesloten:** Geopend;
- **Totaal aantal parkeervakken:** 100;
- **Aantal bezette parkeerplaatsen:** 1398.

(3) Druk op de “OKE” knop en controleer of de melding gegeven wordt dat het systeem connecties met de Parkeer-applicatie accepteert.” [R2010-SWD]

In het Scrum werkdocument staan de overige 3 testen van de Parkeer-systeem sprint. Hieronder staat de tabel met testresultaten uit het Scrum werkdocument.

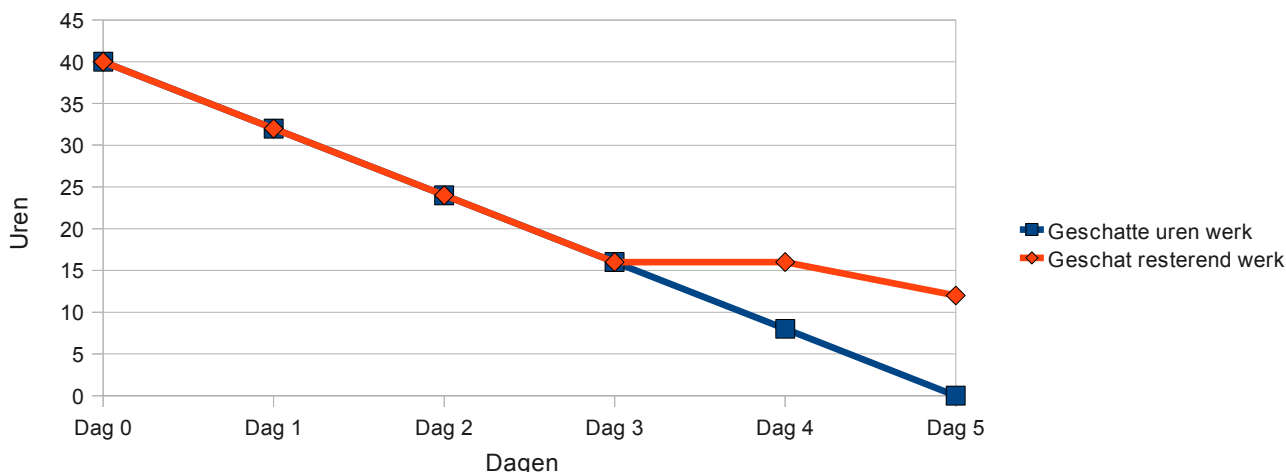
Test	Verwachte resultaat	Resultaat
Test 1: Configureren van Parkeer-service	De volgende melding wordt weergegeven: Het systeem is geconfigureerd met de volgende informatie: Bedrijfsnaam: Q-park Garagenaam: Laakhaven Straat: Laakhaven. Nummer: 27 Postcode: 2516ED Stad: Den Haag Aantal Parkeerplaatsen: 1398. Aantal bezette parkeerplaatsen: 0. Open/gesloten: Open De Parkeer-applicatie accepteert nu connecties van de Parkeer-applicatie.	De melding in het pop-up venster komt volledig overeen met de tekst hiernaast.
Test: 2: Registeren Q-park Laakhaven Parkeer-service in CVIS-DDS	De volgende melding wordt weergegeven: "De Parkeer-service" heeft zichzelf succesvol geregistreerd bij het CVIS-DDS systeem. Daarbij zijn de volgende eigenschappen geregistreerd: Bedrijfsnaam: Q-park Naam: Laakhaven Straat: Laakhaven Nummer: 27 Postcode: 2516ED Plaats: Den Haag Aantal beschikbare parkeerplaatsen: 1398 Open/gesloten: Open Coördaten: 52.063012,4.315245	De melding in het pop-up venster komt volledig overeen met de tekst hiernaast.
Test:3: Dichtstbijzijnde parkeergarage opzoeken	In de GUI van de Parkeer-applicatie staat het adres van de parkeergarage Laakhaven met de toegewezen parkeerplaats.	De garage Laakhaven is gevonden. Maar er is geen parkeergarage toegewezen.
Test 4: Wijzigen aantal beschikbare parkeerplaatsen	Het aantal beschikbare parkeerplaatsen staat op 0 De Parkeer-applicatie toont het adres van de parkeergarage "Rademarkt". Daarbij wordt het toegewezen parkeervak weergegeven	Er wordt een "IndexOutOfBoundsException" gegenereerd.

Tabel 12: Testresultaten Parkeer-service sprint [R2010-SWD]

In het testresultaat staat dat niet alle testen met goed gevolg uitgevoerd zijn. Dit is het gevolg van het ontstane probleem in de ontwikkeling van de Parkeer-service. De Parkeer-service opende een socket maar handelde communicatie niet goed af. Omdat ik in de overtuiging was dat het probleem zich voordeed bij de Parkeer-applicatie duurde het lang voor de juiste diagnose gesteld was en het probleem aan de Parkeer-service verholpen kon worden. Dit probleem heeft het ontwikkel-proces gefrustreerd waardoor ik niet aan alle werkzaamheden van deze sprint ben toegekomen.

Burndown grafiek

In het figuur hieronder ziet u de burndown chart van deze fase.



Figuur 15: Burndown chart parkeersysteem sprint

Het werd op dag drie duidelijk dat de sprint buiten de planning zou treden zoals dat te zien is in de burndown chart hierboven. Tijdens dag drie is de programmatuur van de Parkeer-applicatie gebouwd. Daar werd duidelijk dat de communicatie met de Parkeer-service niet goed werkte. Omdat de oorzaak van het probleem niet direct werd gevonden, liepen de te verrichten werkzaamheden uit zoals u ziet in figuur 15.

Het Prince 2 proces 'Controlling a Stage' schrijft voor dat er contact met de project board opgenomen moest worden in deze situatie. Dat is echter niet gedaan omdat op dag vijf al contact plaatsvond met de leden van de project board i.v.m. de aanstaande demonstratie over de resultaten van de sprint.

De Scrum methodiek schrijft voor dat aan het eind van de sprint de resultaten gedemonstreerd moeten worden, ongeacht de status van het product. Daarom is de demonstratie alsnog gehouden ook al waren nog niet alle functionaliteiten volledig geïmplementeerd.

6.3 Parkeer-systeem verbetering sprint

Tijdens de Parkeer-systeem sprint is het niet gelukt om alle functionaliteiten te bouwen in de demonstrator. Dit was het gevolg van een probleem in de netwerk-communicatie tussen de Parkeer-applicatie en de Parkeer-service. De project board heeft besloten dat de functionaliteiten die tijdens de parkeer-systeem sprint geïmplementeerd moesten worden alsnog tijdens deze sprint geïmplementeerd moest worden. In de tabel op de volgende pagina staat een overzicht van de uit te voeren taken tijdens de Parkeer-systeem verbetering sprint. Deze tabel is afkomstig van het product backlog uit het Scrum werkdocument. Na de tabel volgt een toelichting.

ID	Naam	Prio	Tijd geschat	Hoe te demonstreren
20	Bug: geen Netwerk-communicatie	10	1 ½ dag	De Parkeer-applicatie vraagt met succes een parkeerplaats aan bij de Parkeer-service. Dit gebeurt via netwerk-communicatie waardoor aangetoond wordt dat de netwerk-communicatie werkt.
21	Bug "Wijzigen aantal beschikbare parkeerplaatsen" genereert exception	8	½ dag	Een V.I.P (Madonna) inclusief lijfwachten brengt een bezoek aan de Haagse Hogeschool. Een aantal parkeerplaatsen moeten gereserveerd worden zodat de V.I.P en lijfwachten probleemloos kunnen parkeren. De Q-Park medewerker opent de GUI en reduceert het aantal beschikbare parkeerplaatsen. De Q-Park medewerker controleert of het systeem bevestigt dat het de eigenschappen (properties) van zijn registratie in CVIS-DDS bijgewerkt heeft.
11	Dichtstbijzijnde parkeergarage opzoeken	9	1 dag	De automobilist start de Parkeer-applicatie in zijn auto. De gebruiker kiest voor een zoekmethode, dichtstbijzijnde parkeergarage, in dit geval. Vervolgens zoekt het CVIS-DDS systeem een aantal dichtstbijzijnde parkeergarages. Uit de resultaten kiest de Parkeer-applicatie de gewenste parkeergarage. Parkeergarages zonder vrije plaatsen worden niet weergegeven.
14	Parkeergarage opzoeken op plaatsnaam	4	½ dag	De automobilist start de Parkeer-applicatie in zijn auto. En zoekt de parkeergarage op plaatsnaam. Uit de resultaten kiest de Parkeer-applicatie de gewenste parkeergarage. Parkeergarages zonder vrije plaatsen worden niet weergegeven.
15	Parkeergarage opzoeken op straatnaam	4	½ dag	De automobilist start de Parkeer-applicatie in zijn auto. En zoekt de parkeergarage op straatnaam. Uit de resultaten kiest de Parkeer-applicatie de gewenste parkeergarage. Parkeergarages zonder vrije plaatsen worden niet weergegeven.

Tabel 13: Deel product backlog [R2010-SWD]

Zoals u in tabel 13 kunt zien zijn er twee dagen uitgetrokken voor het oplossen van de bugs en is backlog ID 11 item overgenomen uit de vorige sprint. In de vorige sprint is Backlog ID 11 niet afgerond in verband met de bug dat zich heeft voorgedaan.

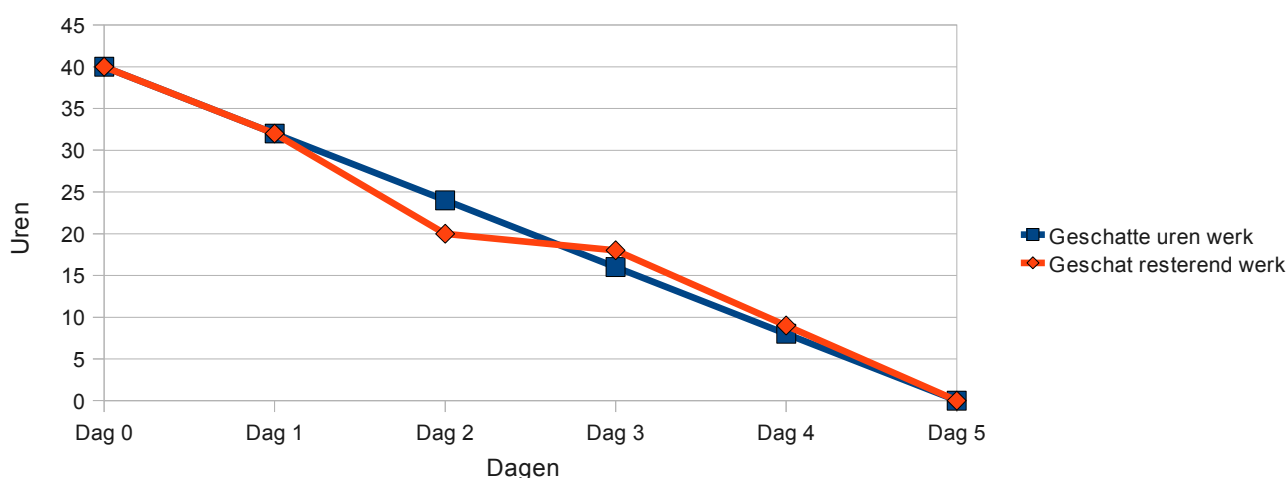
De totale tijd voor het oplossen van de bugs en het ontwikkelen van de functionaliteiten komt neer op vier dagen. Er is een halve dag uitgetrokken voor de voorbereidende werkzaamheden zoals het maken van de use-cases en de ontwerpen. Een halve dag voor deze taken is niet veel, echter hoeven er maar twee use-cases gemaakt te worden (voor Backlog ID 13 & 14). het zelfde geldt voor het uitvoeren van de demonstratie en acceptatietesten. Daar is ook een halve dag voor uitgetrokken. De bestaande ontwerpen hoeven voor beide taken niet veel aangepast te worden. De duur van deze werkzaamheden wordt geschat op een halve dag.

Met de resultaten van de Parkeer-systeem verbetering sprint is de werking van de volgende functionaliteiten van het CVIS-DDS systeem aangetoond:

- Registreren van objecten, met meerdere eigenschappen (properties), dit zijn:
 - String properties;
 - Long properties;
 - Boolean properties;
 - Coordinates properties;
- Wijzigen 'properties' van een geregistreerd object;
- Lokalisatie van objecten op eigenschappen (properties), dit zijn:
 - Long properties;
 - Boolean properties;
 - Coordinates properties;
 - String properties.

Burndown grafiek

De burndown chart van de derde en laatste sprint staat hieronder. Na de illustratie volgt een toelichting.



Figuur 16: Burndown chart parkeer-systeem verbeter sprint

Uit de grafiek kunt u afleiden dat er minder tijd besteed is aan de werkzaamheden die gepland zijn voor de tweede dag. Op de eerste en tweede dag zou gewerkt worden aan het netwerk-communicatie probleem. Dit was echter gemakkelijker op te lossen waardoor de rode lijn in de grafiek sneller daalt dan de blauwe lijn. Echter liepen de voorbereidende werkzaamheden, het maken van use-cases en diagrammen uit waardoor de rode en blauwe lijn in de grafiek ongeveer gelijk lopen. Naar schatting is er in totaliteit vier dagen tijd gestoken om het netwerk-communicatie probleem op te lossen.

Alleen de Parkeer-service applicatie is tijdens deze sprint aangepast. De Parkeer-applicatie heeft een tweetal extra functionaliteiten gekregen ten opzichte van de vorige sprint. Deze nieuwe functionaliteiten maken lokalisatie van garages mogelijk op basis van plaatsnaam en straatnaam. (Zie backlog ID 14 & 15). Hieronder ziet u de nieuwe GUI van de Parkeer-applicatie. Na de illustratie volgt een toelichting.



The screenshot shows the ParkeerApp interface. On the left, under the heading 'Zoek:', there are search options. A checkbox 'DDS gebruiken:' is checked. Below it, a dropdown menu 'Ik ben in:' is set to 'Groningen'. There is a button 'Dichtstbijzijnde garage'. Further down, a search section for 'Zoek op plaatsnaam:' has a dropdown set to 'Zwartewaal' and a button 'Zoek op plaatsnaam'. Below that, a search section for 'Zoek op Straatnaam' has a dropdown set to 'Dorpsstraat' and a button 'Zoek op straatnaam'. At the bottom of the search section is a button 'Verlaat garage'. On the right, under the heading 'Garage:', there are input fields for 'Bedrijfsnaam:' (Zwartewaal Parking), 'Garagenaam:' (-), 'Adres' (with sub-fields for 'plaats:' Zwartewaal, 'Straat:' Dorpsstraat, and 'huisnummer:' 3), 'Postcode:' 3238XV, 'Open/gesloten:' Geopend, and 'Parkeerplaats:' 1. At the bottom of the app window is a blue bar with a circular icon on the left and the text 'ParkeerApp' in the center. The 'makewave' logo is visible in the bottom right corner.

Figuur 17: Screenshot Parkeer-applicatie versie 1.0

In de GUI van de Parkeer-applicatie ziet u een tweetal nieuwe knoppen ten opzichte van de versie die tijdens de parkeersysteem sprint gemaakt is. Met deze nieuwe knoppen kan de automobilist zoeken op plaatsnaam en straatnaam.

6.4 Conclusie

De functionaliteiten van het CVIS-DDS systeem die in de literatuur beschreven zijn, zijn niet volledig geïmplementeerd. Het Publish and Subscribe mechanisme ontbreekt namelijk en het is niet mogelijk om het maximaal aantal resultaten te beperken.

De functionaliteiten die het CVIS-DDS systeem wél aanbiedt hebben toegevoegde waarde voor CVIS. De mogelijkheid om dynamisch en automatisch diensten te registreren, neemt de noodzaak weg om zowel server als cliënt te moeten configureren met netwerk-adres configuratie. Dit reduceert het aantal configuratie-handelingen, evenals de kans op fouten.

Helaas moet er wel aangemerkt worden dat er beperkingen zijn wat de functionaliteiten betreft. Het CVIS-DDS systeem genereert maximaal één resultaat, bij het zoeken op eigenschappen moeten de eigenschappen exact overeenkomen. Daarnaast is er geen mechanisme dat het illegaal overschrijven van bestaande registraties voorkomt. Er komt dus veel verantwoordelijkheid te liggen bij de ontwikkelaar van diensten om de integriteit van het CVIS-DDS systeem intact te houden.

De beschikbaarheid van het CVIS-DDS systeem is hoog, de ervaringen met de stabiliteit van het CVIS-DDS systeem zijn positief. Het systeem is niet “gecrashed” tijdens de ontwikkeling en de demonstraties van de demonstrator.

Helaas zijn er wel problemen voorgekomen bij het beïnvloeden van de zoekresultaten (Java.io.NotSerializableException bij een zoekoperatie waar een SearchConstraint-object bij gebruikt werd). Deze fout werd netjes getoond in de console terwijl het CVIS-DDS systeem zonder verdere problemen bleef functioneren.

Het CVIS-DDS systeem is gemakkelijk te implementeren in de toepassingen van CVIS-DDS. Echter moet de ontwikkelaar zich dan wel eerst verdiepen in de literatuur en de klasse-diagrammen die gemaakt zijn tijdens de onderzoeksfase. Er is dus geen officiële documentatie van het CVIS-DDS systeem. Echter kunt u in de bijlage van dit document de zelfgemaakte klasse-diagrammen van de CVIS-DDS API vinden.

In de huidige vorm biedt het CVIS-DDS systeem niet de mogelijkheden om op grote schaal ingezet te worden. Het centrale kenmerk van het CVIS-DDS systeem belemmert dit.

6.5 Afsluiting van het project

De Prince 2 processen “managing stage boundaries”, “closing a project” en de scrum sprint demonstratie zijn gecombineerd. Er is een laatste demonstratie gehouden van het parkeer-systeem. De resultaten van de acceptatietest staan in het Scrum werkdocument. Omdat de opdrachtgever vond dat de demonstratie duidelijk het nut van het CVIS-DDS systeem aantoonde heeft hij het idee geopperd om de demonstratie ook te geven voor een ander project binnen Logica. Dit is het Strategic Platform for Intelligent Traffic Systems (Spits) project waar een CVIS-DDS systeem van toepassing zou kunnen zijn.

De evaluaties die tijdens de afsluiting van het project beschreven zijn kunt u vinden in het volgende hoofdstuk.

De producten zijn overhandigd tijdens de afsluiting van het project (demonstrator, netwerkarchitectuur rapportage, en de bevindingenrapportage).

In dit hoofdstuk is het proces van de realisatiefase beschreven. De realisatiefase is opgesplitst in een drietal Scrum sprints waarin de demonstrator afgerond is. De demonstrator, dat meerdere malen tijdens het proces gedemonstreerd is aan de project board, illustreert de werking en noodzaak van het CVIS-DDS systeem. Er is gebruik gemaakt van een toepassing dat tot de verbeelding spreekt waar het nut van het CVIS-DDS systeem getoond wordt. Na dit hoofdstuk volgen de evaluaties.

7 Evaluatie

Dit hoofdstuk beschrijft zowel de product als de proces-evaluaties.

7.1 Productevaluatie

In deze paragraaf staan de evaluaties van de producten die opgeleverd zijn. Onderdeel van deze evaluaties is het valideren van kwaliteitsaspecten waaraan de producten moeten voldoen.

Productevaluatie plan van aanpak

In het plan van aanpak heb ik de Prince 2 projectmanagement methode een aantal maal te “strikt” toegepast. Zo is er een project board samengesteld; dit lijkt mij achteraf overdreven omdat de project board maar uit een beperkt aantal personen bestaat. De project board bestaat uit de Logica begeleider en opdrachtgever. Ook zit ik op de zelfde afdeling en bestaat er een informele verhouding met de leden van de project board wat een dergelijke formaliteit overbodig maakt.

De spelling en grammatica van het plan van aanpak zijn meerdere malen door de project board gecontroleerd, waardoor er een hogere kwaliteit is behaald.

Het plan van aanpak vind ik op enkele punten te gefragmenteerd. Onderdelen met gerelateerde informatie zijn op verschillende plaatsen in het document beschreven. Zo is de globale planning te vinden in hoofdstuk drie en kan een grafische weergave van de globale planning gevonden worden in bijlage één van het plan van aanpak. De detailplanning van de onderzoeksfase is weer terug te vinden in bijlage twee van het plan van aanpak.

Onderzoeksrapportage

De onderzoeksrapportage is gebaseerd op bestaande literatuur van het CVIS-project. Aan de literatuurstudie naar CVIS-DDS konden geen eisen gesteld worden, omdat de beschikbare literatuur zeer beperkt was.

Bij het literatuuronderzoek naar het OMG-DDS systeem is gebruik gemaakt van literatuur waarin wordt gerefereerd naar de officiële OMG-DDS documentatie. Helaas was deze officiële documentatie zelf niet vrij beschikbaar. Aan de beschikbare literatuur is de eis gesteld dat er geen informatie gebruikt wordt van bronnen die publiekelijk aan te passen zijn (denk hierbij aan informatiebronnen als Wikipedia).

Aan de onderzoeksrapportage is een aantal eisen opgesteld. Deze eisen staan hieronder in de tabel. De maatregelen en kwaliteitseisen die betrekking hebben op de bevindingenrapportage kunt u vinden in het plan van aanpak. In paragraaf 2.12 zijn de generieke kwaliteitseisen en maatregelen van de rapportages beschreven. In het onderzoeksfase plan (zie bijlage plan van aanpak) staan een aantal specifieke eisen die gesteld zijn aan de bevindingenrapportage.

Eisen	Voldaan (J/N)	Verantwoording
Generieke kwaliteitseisen		
Nette en zakelijke tekst.	J	Het document is door meerdere personen gecontroleerd.
Spelling en grammatica.	J	Het document is door meerdere personen gecontroleerd.
Juistheid en volledigheid.	J	In het onderzoeksrapport zijn er meerdere citaten en diagrammen overgenomen uit de literatuur. Er is gestructureerd te werk gegaan door gebruik te maken van de "Mindmapping" techniek. Daarnaast zijn kwaliteitseisen gesteld aan de gehanteerde literatuur zodat niet de verkeerde conclusies getrokken kunnen worden door slechte bronnen.
Tijdig opleveren van producten.	N	Het onderzoeksrapport is later opgeleverd dan gepland door het uitlopen van de onderzoeksfase. Zie paragraaf 4.9.
Specifieke kwaliteitseisen		
Overzicht van de gebruikte literatuur.	J	Zie hoofdstuk 6.
Vergelijking van de concepten OMG-DDS en CVIS-DDS.	J	Zie hoofdstuk 4.
Beschrijving van de bekende functionaliteiten van het CVIS-DDS systeem.	J	Zie hoofdstuk 2.
De resultaten van de experimenten.	J	Zie paragraaf 2.3.

Tabel 14: Overzicht kwaliteitseisen onderzoeksrapportage

Bevindingenrapportage

De bevindingenrapportage is een rapport dat de kwaliteit van het CVIS-DDS systeem beschrijft op een aantal punten.

Ik vind de bevindingenrapportage een zinnig document, omdat er duidelijk in aangegeven is welke functionaliteiten er werken en hoe tot deze conclusie gekomen is. Zo worden de conclusies controleerbaar gemaakt.

Aan de bevindingenrapportage is een aantal eisen opgesteld. Deze eisen staan hieronder in de tabel. De maatregelen en kwaliteitseisen die betrekking hebben op de bevindingenrapportage kunt u vinden in het plan van aanpak. In paragraaf 2.12 zijn de generieke kwaliteitseisen en maatregelen van de rapportages beschreven. In het onderzoeksfase plan (zie bijlage plan van aanpak) staat een aantal specifieke eisen die gesteld zijn aan de bevindingenrapportage.

Eisen	Verantwoording
Generieke kwaliteitseisen	
Nette en zakelijke tekst.	Het document is door meerdere personen gecontroleerd.
Spelling en grammatica.	Het document is door meerdere personen gecontroleerd.
Juistheid en volledigheid.	Er is een aantal maal tekst geciteerd uit de CVIS-literatuur en programmatuur geciteerd uit de demonstrator, waarmee de bevindingen onderbouwd en controleerbaar gemaakt zijn. Verder zijn er specifieke kwaliteitseisen gesteld die hieronder gedefinieerd staan.
Product specifieke kwaliteitseisen	
Overzicht van de gebruikte literatuur.	Zie hoofdstuk 7
Beschrijving van de plaats van dit document in de inleiding.	Zie hoofdstuk 1
Afsluiten en samenvatten van hoofdstukken.	Gehele document
Een beschrijving van het installatiegemak, gebaseerd op de ervaringen	Zie paragraaf 5.6
Een beschrijving van de stabiliteit, gebaseerd op de ervaringen.	Zie paragraaf 5.2
Klassediagrammen van de demonstrator	Bijlage
Beschrijving van de toepassing van de API	Zie hoofdstuk 4
Beschrijving van de kwaliteit van het CVIS-DDS systeem	Zie hoofdstuk 5
Beschrijving van de problemen met het CVIS-DDS systeem	Zie paragraaf 4.5

Tabel 15: Overzicht kwaliteitseisen bevindingenrapportage

Zoals te zien is in tabel 15, is voldaan aan alle kwaliteitseisen van de bevindingenrapportage.

Demonstrator

De demonstrator is een product waar de project board zichtbaar enthousiast van werd. Er is zelfs sprake geweest om de demonstrator te demonstreren bij een ander project van Logica.

De demonstrator ziet er goed en simpel uit. De toegewezen parkeerplaats in de GUI van de parkeer-applicatie had ik echter beter in beeld kunnen brengen door deze een prominentere plaats te geven in de GUI of het lettertype te vergroten.

In de tabel hieronder zijn de eisen die aan de demonstrator gesteld zijn beschreven. Daarnaast is aangegeven of er voldaan is aan de betreffende eis. Ook is er een verantwoording beschreven waarom er voldaan is aan de eis en waar eventueel bewijslast gevonden kan worden. De kwaliteitseisen zijn afkomstig van het plan van aanpak, uit paragraaf 2.12 om precies te zijn. Daar kunt u de maatregelen, die genomen zijn om de kwaliteit te verhogen, nalezen.

Eisen	Verantwoording
Functionaliteit	
Geschiktheid.	Bij het opstellen van de stories in de product backlog en de use-cases is de opdrachtgever nauw betrokken geweest. Deze heeft kunnen bijsturen op de geschiktheid van de functionaliteiten. Daarnaast zijn de functionaliteiten van het CVIS-DDS systeem betrokken bij het opstellen van de use-cases en de testcases, waardoor de geschiktheid van een betreffende functionaliteit heroverwogen wordt. De functionaliteiten van de demonstrator spreken tot de verbeelding. Een parkeersysteem dat automatisch een parkeerplaats toewijst is erg begrijpelijk.
Juistheid en volledigheid.	Voorafgaand aan het implementeren van de programmatuur zijn er testcases opgesteld. Daardoor wordt voorafgaand nagedacht over de juistheid en volledigheid van de nog te realiseren functionaliteiten van de demonstrator. In hoofdstuk 6 van dit afstudeerverslag is er een deel van de testresultaten beschreven.
Bruikbaarheid	
Leerbaarheid (werking van het CVIS-DDS systeem aantonen).	In de use-cases en de testplannen (zie tabel 12 in hoofdstuk 6) is een aantal keer beschreven dat er door de demonstrator een pop-up bericht getoond wordt met berichtgeving over wat er gaat gebeuren of reeds gebeurd is. Daar kan de werking van het CVIS-DDS systeem afgeleid van worden. Ook is de GUI dusdanig ontworpen dat de functionaliteiten duidelijk zijn, omdat er kernachtige termen gebruikt zijn en er een beperkt aantal bedieningsmogelijkheden is.
Duidelijkheid.	Het CVIS-team van de Logica organisatie is de doelgroep van de demonstrator. Aan hen en de project board zijn de demonstraties van de demonstrator gegeven. Omdat de ontwikkelingsmethodiek de project board en het CVIS-team betrokken heeft bij het ontwikkelingsproces is de duidelijkheid van de demonstrator regelmatig geverifieerd.
Bedieningsgemak.	De mogelijkheden wat betreft de bediening van de demonstrator is beperkt. Zo zijn er geen invoervelden. (Alle velden worden automatisch ingevuld zodra een keuze gemaakt is met een pulldown menu.) Verder is er een beperkt aantal knoppen, waardoor de interface kernachtig en overzichtelijk is voor de gebruiker.
Onderhoudbaarheid	
Wijzigbaarheid.	De "Java naming conventions" zijn aangehouden bij het ontwikkelen van de

Eisen	Verantwoording
	<p>programmatuur. Er is daarnaast ook gebruik gemaakt van de Eclipse automatische “code formatting functionaliteit” om de programmatuur op een juiste manier te structureren.</p> <p>In de bevindingenrapportage is er een toelichting op de programmatuur van de demonstrator gegeven. Daarbij is er gebruik gemaakt van UML-diagrammen en bijhorende beschrijvingen. Op deze manier wordt de code begrijpelijk gemaakt en dus ook wijzigbaar. Uiteraard is de programmatuur ook voor zien van commentaar.</p>
Herbruikbaarheid.	Idem.

Tabel 16: Overzicht kwaliteitseisen bevindingenrapportage

7.2 Procesevaluatie

Initiatiefase

Het opstellen van de opdrachtschrijving had ik kunnen bespoedigen door de Logica begeleider (opdrachtgever) en technisch begeleider eerder op “één lijn” te proberen krijgen. Ik heb namelijk te lang geprobeerd de verschillende interpretaties van de bedrijfsbegeleider en technisch expert in het plan van aanpak te verwerken. Ik had in een vroeg stadium met beide heren gezamenlijk moeten overleggen en niet individueel.

De grote wijzigingen in de opdrachtschrijving hadden beter duidelijk gemaakt moeten worden aan de examinatoren van de Haagse Hogeschool. De wijzigingen in de opdrachtschrijving zijn wel door mij gecommuniceerd maar naar mijn mening niet voldoende opgemerkt omdat er pas later tijdens het afstudeertraject ingegrepen is door de competenties aan te passen.

De fasering die gemaakt is tijdens de initiatiefase is niet correct gebleken. Er is tijdens de initiatiefase te weinig nagedacht over de uitvoering van de realisatiefase. Hoe de realisatiefase er uit zou zien en welke strategieën er toegepast zouden worden is niet voldoende toegelicht.

Hoewel de opdrachtschrijving gewijzigd is, zijn enkele “resten” van het Afstudeerplan herkenbaar in de op te leveren producten. De netwerkbeschrijving is tijdens de initiatiefase niet geschrapt omdat op dat moment kennis van het CVIS-netwerk als nodig werd geacht. Echter ben ik tegen het einde van het project van mening dat dit rapport geen toegevoegde waarde heeft en beter geschrapt had kunnen worden.

In dit project heb ik te veel tijd en energie verspild aan het implementeren en uitvoeren van de Prince 2 methodiek. Het opstellen van een project board en het schrijven van een exception rapport zijn daar goede voorbeelden van.

Tijdens de initiatiefase is gekozen om gebruik te maken van Prince 2. Terugblikkend op dit project denk ik dat het beter was om RUP toe te passen. Dit klinkt wellicht verbazingwekkend in oren omdat ik de strategie van dit project tijdens de onderzoeksfase heb aan gepast, toen is de keuze gemaakt voor Scrum en niet voor RUP. Scrum was gezien de beperkte leercurve en de aanwezige tijdsdruk de beste keuze op dat moment. Het was beter geweest als ik bij aanvang van dit project meteen gekozen had voor RUP. De iteratieve eigenschappen van RUP hadden goed gepast bij dit project. Een andere kenmerkende eigenschap van RUP is het proces waarin de requirements continu gemanaged worden. Deze eigenschap zou goed passen bij dit project. De requirements zijn in een R&D-project namelijk moeilijk te definiëren en kunnen gezien het innoverende karakter wijzigen (omdat er altijd een mate van onzekerheid is aan de haalbaarheid van de requirements). De literatuurstudie had, wanneer RUP toegepast zou worden, uitgevoerd kunnen worden tijdens de inception fase of er had een aparte fase hiervoor ingevoerd kunnen worden.

Onderzoeksfase

De aanpak van de onderzoeksfase is succesvol gebleken. Uit de CVIS-literatuur zijn meerdere functionaliteiten en mechanismen van het CVIS-DDS systeem gekomen. Het valideren van deze functionaliteiten en mechanismen is zinvol geweest. Enkele mechanismen bleken niet volledig geïmplementeerd te zijn of zelfs volledig te ontbreken. Denk hierbij aan het Publish and Subscribe mechanisme.

De vergelijking met het CVIS-DDS systeem en de OMG-DDS standaard heeft geen nieuwe functionaliteiten opgeleverd. Achteraf gezien had ik de vergelijking beter kunnen maken met CVIS-DDS en één of meerdere geïmplementeerde OMG-DDS systemen die overeenkomsten hebben met het CVIS-DDS systeem om zo tot meer functionaliteiten te komen.

De methode die toegepast is om informatie te structureren is Mindmapping. Deze techniek is toegepast volgens de omschreven techniek in het boek: "Mindmapping in de praktijk". Zoals beschreven in paragraaf 3.5 heb ik de techniek toegepast volgens de beschrijvingen uit het boek [DS1994].

Het Managing stage Boundaries proces is na de onderzoeksfase uitgevoerd. De werkzaamheden van de volgende fasen zijn uitgewerkt; er is een Scrum werkdocument opgesteld. De eisen aan de producten (demonstrator en de bevindingenrapportage) zijn op dat moment vastgesteld.

Het proces planning is minder goed uitgevoerd. Het moment waarop het exception rapport is verstuurd naar de project board, was te laat. Het uitlopen van de onderzoeksfase had eerder geconstateerd moeten worden. Echter is het project wel op de juiste manier geëscaleerd en is er gerichte en ingrijpende actie ondernomen op het uitlopen van de onderzoeksfase.

Realisatiefase

Tijdens de realisatiefase heb ik Scrum goed toegepast. Kenmerkende onderdelen van de Scrum methodiek komen duidelijk naar voren in de beschrijving van het proces. Denk hierbij aan de Scrum backlog, de sprint backlog en de timeboxing. Deze aspecten zijn allemaal in hoofdstuk 6 behandeld.

Tijdens de realisatiefase ben ik erg gestructureerd te werk gegaan. Eerst is het product backlog opgemaakt en vervolgens zijn daar use-cases met use-case diagrammen van gemaakt evenals de klassediagrammen. Daarna zijn de testen opgesteld en is de programmatuur geïmplementeerd. De resultaten zijn vervolgens getest en gedemonstreerd.

De methodiek heeft het ontwikkelingsproces naar mijn gevoel versneld. De toegepaste ontwikkelmethodiek maakte de werkzaamheden afwisselend waardoor tunnelvisie voorkomen werd. Het opsplitsen de werkzaamheden in verschillende sprints maakten de werkzaamheden “behapbaar” en overzichtelijk.

Er is te lang doorgewerkt op de netwerk-communicatie bug die zich tijdens de tweede sprint heeft voorgedaan. Ik had een maximum aan tijd moeten stellen waarbinnen de bug opgelost moest worden. Als de bug niet opgelost kon worden dan moest daar tijdens de volgende sprint tijd voor gereserveerd worden.

7.3 Lessons learned

De grote wijzigingen in de opdrachtomschrijving hebben geleid tot onzekerheden en een behoorlijke portie stress. Hoewel de wijzigingen aangegeven zijn aan de Haagse Hogeschool was ik ongerust over de haalbaarheid van dit project. De tijd die verloren is gegaan aan het opnieuw formuleren van de opdrachtomschrijving heeft de nodige spanning veroorzaakt. Mocht ik opnieuw moeten afstuderen, dan zou ik in een vergelijkbare situatie stoppen met de opdracht en weer opnieuw een Afstudeerplan inleveren met de vernieuwde opdrachtomschrijving.

In het vervolg moet ik niet te lang op één bug blijven hangen, hiermee doel ik het netwerk-communicatie probleem. Het frustreerde mij enorm dat ik deze bug niet kon oplossen waardoor ik obsessief ben blijven proberen het probleem op te lossen. Daardoor is de tweede sprint, de Parkeer-service sprint uitgelopen. Het probleem bleek uiteindelijk erg klein te zijn en was gemakkelijk en veel sneller op te lossen als ik er met een frisse blik naar gekeken had.

In een vervolg project moet ik de voortgang beter monitoren en moet ik bij problemen eerder “aan de bel trekken”.

In een betrekkelijk klein project als deze zou ik Prince 2 niet nogmaals gebruiken. De opzet van een ander project zou ik mogelijk wel laten inspireren op Prince 2 zodat de “abstracte kenmerken” gebruikt worden. Wanneer de term Prince 2 “gevoerd” wordt, moet er veel tijd besteed worden aan de correcte implementatie van de methode.

7.4 Probleem opgelost?

De probleemstelling:

“De software van het CVIS-DDS systeem is opgeleverd door Technolution B.V. De werking van dit systeem en de functionaliteiten zijn bij Logica onbekend waardoor het mogelijk niet optimaal gebruikt wordt; Zo wordt het systeem nu toegepast om de GPS-locatie van de CVIS-vervoersmiddelen te registreren en bij te werken.

Tijdens de implementatie van het CVIS-DDS systeem is men tegen vele problemen aangelopen, onbekend is welke problemen er nu nog bestaan.” [R2009-PVA]

Het probleem is niet geheel opgelost; het CVIS-DDS systeem wordt momenteel nog steeds verkeerd toegepast. Wel beschikt het Logica CVIS-team over de informatie om het CVIS-DDS systeem correct toe te passen in andere informatiesystemen.

De problemen van het CVIS-DDS systeem zijn verhelderd in de bevindingenrapportage.

7.5 Doelstelling behaald?

De doelstelling van dit project is opgesplitst in een tweetal delen:

Het eerste deel van de doelstelling:

“De functionaliteiten en werking van het CVIS-DDS systeem achterhalen en verhelderen binnen de organisatie.” [R2009-PVA]

De functionaliteiten van het CVIS-DDS systeem zijn tijdens dit project achterhaald, deze zijn zowel in de onderzoeks- als in de bevindingenrapportage beschreven.

Het tweede deel van de doelstelling:

“... Daarbij wordt de werking van het systeem geïllustreerd en worden de tekortkomingen van het systeem verhelderd.” [R2009-PVA]

De demonstrator illustreert de werking van het CVIS-DDS systeem. Op technisch vlak zijn de CVIS-DDS API en de stukken programmatuur van de demonstrator toegelicht. Verder zijn de tekortkomingen van het systeem verhelderd in de bevindingenrapportage.

8 Literatuurlijst

Netwerk:

- [DWPT2002] V. Devarapalli (Nokia), R. Wakikawa (Keio University), A. Petrescu (Motorola), P. Thubert (Cisco Systems);
Network Mobility (NEMO) Basic Support Protocol (RFC 3963);
- [B2009]L. Bilderbeek
Test-Site Deployment v1.0.
2009

OMG-DDS

- [SF2006]
Stan Schneider, Bert Farabaugh.
Is DDS for You? A Whitepaper by Real-Time Innovations.
- [XPEN2007]
Ming Xiong, Jeff Parsons, James Edmondson, Hieu Nguyen, en Douglas Schmidt
Vanderbilt University
Evaluating Technologies for Tactical Information management in Net-Centric Systems.
- [JP-C2006]
Rajive Joshi, Gerardo Pardo-Castellote
OMG Data-Distribution Service: Architectural Overview

CVIS-DDS:

- [SSBA2009]
Arnor Solberg (QFREE), Andreas Schmid (PTV), Mick Baggen(Technolution), Francesco Alesiani (Mizar), Hannes Stratil (Efkon), Richard Bossom (Siemens), Silke Forkert (PTV), Marius Schlingelhof (DLR), Axel Burkert (PTV), Sjoerd Haverkamp (PEEK), Paul Mathias (Siemens), Zeljko Jeftic (ERTICO).
D.CVIS.3.3 Architecture and System Specifications;
- [SSLO2007]
Andreas Schmid, Arnor Solberg, Christer Larsson, Erik Olsen, Frank Tuijnman, Gino Franco, Hans Hadderingh, Jean-François Gaillet, Josef Kaltwasser, Knut Evensen, Marcel Konijn, Mick Baggen, Richard Bossom, Siebe Turksma, Thierry Ernst, Zeljko Jeftic.
D.CVIS.3.2 High Level Architecture.
- [WWWCVIS] (laatst bezocht op: 8-02-2010)
http://www.cvisproject.org/en/cvis_project/objectives/

Methodieken

- [OK2005]
Mark van Onna, Ans Koning
De kleine Prince 2
- [KN1995]
KIVI & NGI
ASI-Rapport
- [K2007]
Henrik Kniberg. Scrum And XP from the Trenches;
Scrum in Five Minutes;
- [WWWXP] (laatst bezocht op: 8-02-2010)
<http://www.extremeprogramming.org/rules/functionaltests.html>

Technieken

- [DS1994]
Nelleke Dorrestijn en Ingemar Svantesson.
Mindmapping in de praktijk.
- [WK2004]
Jos Warmer en Anneke Kleppe
Praktisch UML

Producten van dit project

- [R2009-AP]
Robert van Reems
Afstudeerplan
- [R2009-PVA]
Robert van Reems
Plan van Aanpak
- [R2009-netwerk]
Robert van Reems
Netwerkarchitectuur beschrijving
- [R2009-onderzoek]
Robert van Reems
Onderzoeksrapportage
- [R2009-netwerk]
Robert van Reems
Bevindingenrapportage
- [R2010-SWD]
Robert van Reems
Scrum werkdocument

9 Bijlagen

Bijlage 1: Competenties

Bijlage 2: Exception rapport

Bijlage 3: Omschrijving CVIS

Bijlage 4: Brief aan medewerkers Technolution B.V.

Externe bijlage:

1. Plan van aanpak
2. Netwerkarchitectuur beschrijving
3. Onderzoeksrapportage
4. Bevindingenrapportage

Bijlage 1: Competenties

- A1 Niveau 4 Analyseren van het probleemdomain
In dit afstudeerverslag, de onderzoeksrapportage en bevindingenrapportage is beschreven hoe de analyse naar het probleemdomain uitgevoerd is. Onder de analyse valt de analyse naar de mogelijkheden, kwaliteit en problemen van CVIS-DDS het CVIS-DDS systeem.
- C10 Niveau 3 Ontwerpen van een systeemarchitectuur
De systeemarchitectuur van de demonstrator is beschreven in het afstudeerverslag daar wordt ingegaan op de aansluiting van de systeemeisen. Ook zijn de keuzemogelijkheden wat betreft de alternatieven toegelicht.
- C8 Niveau 3 Ontwerpen van een technisch informatiesysteem
In het Scrum werkdocument zijn de ontwerpen en beschrijvingen van de demonstrator geplaatst. De ontwerpen zijn gebaseerd op UML. De functionaliteiten van het CVIS-DDS systeem zijn aangetoond in de demonstrator met behulp van de acceptatietesten en verwerkt in de ontwerpen.
- H6 Niveau 3 Professioneel werken: Resultaatgericht werken
In het afstudeerverslag is verantwoording afgelegd waarom niet alle producten tijdig opgeleverd zijn. In het Scrum werkdocument staat een geprioriteerde lijst met functionaliteiten (product backlog) en een planning (sprint backlog) dat gebaseerd is op het product backlog. In de evaluatie is de kwaliteit van de producten aangetoond.

Bijlage 2: Exception rapport

Inleiding

Het project loopt achter. Om de vertraging toe te lichten is hieronder een deel van de globale planning van het project hieronder weergegeven:

Processen en taken	Kalender week	Initiatiefase					Onderzoeksfase								Ontwerpfase				Realisatiefase
		DagerWeek					1 2 3 4 5 6 7 8 9 10 11 12 13												
Beoordelen project status, evalueren en kwaliteitbevordering	8					10,5	0,5	0,5	0,5	0,5					10,5	0,5	0,5		
Fase afsluiten en opstarten	3,5						1								0,5	0,5	0,5		
Schrijven afstudeerverslag	12							0,5	10,5	2,5	0,5	0,5	0,5		2				
Opzetten van het project	20,5		3	5	5	4	3,5												
De architectuur van de huidige communicatie infrastructuur bes	1							1											
Literatuurstudie naar concept van DDS	6,5							3	3,5										
Bezoek begeleider/examinator HHS	0,5							0,5											
Bepalen mogelijkheden van DDS	7									4	2	1							
Concept bespreking HHS	0,5												0,5						
Vaststellen systeemeisen van DDS en de demonstrator	0,5												0,5						
Schrijven van concept rapportage bevindingen DDS-systeem	2,5												1						
Ontwerpen demonstrator	7														3,5	3,5			
Gereedmaken test site	2																2		
Bouwen demonstrator	4,5																		
Tussentijdse assessment	0,5																		
Testen demonstrator	2																		
Demonstreren van werkende demonstrator	1																		
Eerste overdracht van de producten	0,5																		
Nazorg	3																		
Deadline	0,5																		
Dagen			3	5	5	5	5	5	5	5	5	5	5	5	5	5	5		

Figuur 1: Deel originele globale projectplanning

Zoals u hieruit kunt opmaken moet het project in week 2 van de ontwerpfase zitten. Echter is dit helaas niet het geval.

De vertraging is opgelopen doordat veel tijd gestoken is in het afstudeerverslag. Het project loopt bijna drie weken achter op de planning. Nader inzicht heeft mij doen inzien dat een Agile-achtige strategie beter past bij deze opdracht.

De strategie

Een Agile-achtige ontwikkelmethodiek wordt voorgesteld omdat deze beter past bij dit project en het opleveren van een product "zeker stelt". Een Agile-strategie past om sneller resultaat te bereiken en te voorkomen dat dit project niets oplevert wordt het gebruik van de Agile ontwikkelmethodiek aangeraden. Middels korte iteraties wordt getracht de functionaliteiten van het CVIS-DDS systeem aan te tonen en uit te breiden. Hierbij kan de MoSCoW-techniek toegepast worden.

Bijlage 3: Omschrijving CVIS

Deze paragraaf belicht de context van het project. Daarbij wordt de technische architectuur van het CVIS-project op hoog niveau beschreven.

CVIS is een consortium van meer dan zestig Europese organisaties die in opdracht van de Europese Commissie aan coöperatieve verkeerssystemen werken. De doelstellingen van het CVIS-DDS systeem zijn op de volgende pagina geciteerd.

- *“to create a unified technical solution allowing all vehicles and infrastructure elements to communicate with each other in a continuous and transparent way using a variety of media and with enhanced localisation;*
- *to enable a wide range of potential cooperative services to run on an open application framework in the vehicle and roadside equipment;*
- *to define and validate an open architecture and system concept for a number of cooperative system applications, and develop common core components to support cooperation models in real-life applications and services for drivers, operators, industry and other key stakeholders;*
- *to address issues such as user acceptance, data privacy and security, system openness and interoperability, risk and liability, public policy needs, cost/benefit and business models, and roll-out plans for implementation”. [MWWCVIS]*

De coöperatieve systemen van CVIS zijn slimme, communicerende en samenwerkende systemen waarmee allerlei verkeersinformatie verzameld en verspreid wordt. In CVIS wordt een open-application-framework voor een breed spectrum aan coöperatieve diensten ontwikkeld. Deze diensten worden aangeboden aan verkeersdeelnemers, beheerders, de industrie en andere belanghebbende. De samenwerking tussen de systemen kan de volgende vormen aannemen Vehicle-2-Vehicle, Vehicle-2-Infrastructure en Infrastructure-2-Infrastructure. Bij Vehicle-2-Vehicle vindt samenwerking plaats tussen systemen in de verschillende vervoersmiddelen (Ook wel In-Vehicle-Units genoemd), bij Vehicle-2-Infrastructure vindt samenwerking plaats tussen de systemen van de infrastructuur (ook wel Roadside-Units genoemd) en de vervoersmiddelen.

Logica is een belangrijke spil in de CVIS-organisatie. De producten die opgeleverd worden door partners van CVIS, worden door Logica geïntegreerd in één omgeving. Daarnaast is Logica verantwoordelijk voor het opzetten van de Nederlandse en Belgische testopstellingen en het onderhouden van de netwerkarchitectuur.

De In-Vehicle-Units en Roadside-Units bestaan uit verschillende onderdelen. De units bestaan op hun beurt weer uit een zogeheten “router” en een “host”. De In-Vehicle-Units bieden de toepassingen aan de gebruiker aan en verzorgen de communicatie met de verschillende netwerken via de infrastructuur en/of het 3G-netwerk. De Roadside-Units bieden diensten en communicatie faciliteiten aan. Op het internet zijn verschillende centrale systemen die diensten aanbieden aan de In-Vehicle- en de Roadside-Units.

CVIS Netwerk-componenten

Het mobiele karakter van CVIS maakt de communicatie architectuur van CVIS complex. De communicatie architectuur vraagt daarom om verschillende specifieke systemen en communicatiemedia.

De volgende systemen zijn voor de communicatie architectuur vereist:

- Home Agent;
- Host Management Centre;
- Distributed Directory Service.

Deze systemen worden hieronder toegelicht.

Home Agent

Binnen het Logica Central service center draait een Home Agent. De taak van de Home Agent is unieke identificatie van de In-Vehicle routers mogelijk te maken. De communicatie in het vervoersmiddel wordt verzorgd door een In-Vehicle router. Doordat deze router middels verschillende media verbinding kan maken met het internet wijzigt het IP-adres continu. Om deze reden is een In-Vehicle router extern niet te identificeren.

Binnen CVIS is het echter vereist dat de systemen extern te benaderen en te identificeren zijn, dit probleem wordt opgelost door Home Agent. De In-Vehicle router bouwt een NEMO-tunnel op met de Home Agent. Via deze tunnel kan er bi-directioneel gecommuniceerd worden met de router. Meer informatie over de NEMO-tunnel kunt u vinden in: [R2009-netwerk].

Host Management Center

Het Host Management Center (HMC) distribueert software, en daarbij ook updates aan de (In-Vehicle) cliënt-systemen. Een dergelijk systeem is nodig omdat CVIS een groot aantal diensten (service) aan de weggebruiker biedt. De software van deze diensten moet door CVIS zelf beheerd worden. Een goed voorbeeld van één van de vele diensten zijn de vele verschillende tol-systemen. De software van de diensten zijn niet vooraf geïnstalleerd op de In-Vehicle systemen maar worden automatisch geïnstalleerd wanneer deze nodig zijn. Het systeem dat hiervoor verantwoordelijk is het Host Management Centre. Het HMC distribueert de cliëntsoftware en is verantwoordelijk voor het up-to-date houden van deze software bij de cliënts.

Distributed Directory Service

Distributed Directory Service CVIS-DDS is een soort dynamische “Gouden Gids” van alle lokale diensten (services) die aangeboden worden. Het dienstenaanbod verschilt per locatie en het maken van dataverbindingen vormt een probleem omdat het vervoersmiddel zich snel verplaatst en daardoor regelmatig switcht tussen netwerken. CVIS-DDS maakt snelle dynamische lokalisatie van diensten mogelijk en zet de communicatie verbindingen op.

Binnen de EU verschilt het aanbod en type aan diensten sterk per regio en land.

Een parkeersysteem is een goed voorbeeld van een dienst dat de noodzaak van het CVIS-DDS aantoont. Verschillende parkeergarages verspreid over de gehele EU bieden een Parkeer-service aan op het open application framework. De service houdt in dat automobilisten middels een Parkeer-applicatie een parkeerplaats toegewezen krijgen zodra de automobilist de auto wil parkeren.

Automobilisten kunnen parkeergarages gemakkelijk vinden waarbij deze geselecteerd kunnen worden op basis van eigenschappen zoals locatie en het aantal beschikbare parkeerplaatsen. De Parkeer-applicaties in de auto moeten contact kunnen maken met de Parkeer-service om een parkeerplaats toegewezen te kunnen krijgen.

Zonder het CVIS-DDS systeem is het niet mogelijk om automatisch verbinding met de service tot stand te kunnen brengen; de Parkeer-applicaties kunnen gezien de hoeveelheid aan parkeergarages onmogelijk over een recent overzicht van parkeer-garages beschikken. Ook het selecteren van een parkeergarage op basis van beschikbare parkeerplaatsen kan zonder het CVIS-DDS systeem niet automatisch plaatsvinden.

Hierboven is het CVIS-DDS systeem omschreven als een centraal systeem. Het CVIS-DDS systeem biedt volgens de literatuur ook een alternatief mechanisme aan, met dit mechanisme kunnen applicaties elkaar op dynamische wijze lokaliseren. Dit mechanisme wordt het Publish and Subscribe mechanisme genoemd.

Het Publish and Subscribe mechanisme stelt applicaties in staat om zichzelf te publiceren (Publish). Een applicatie maakt zich middels het publiceren vindbaar op het netwerk. Tegenover het publiceren staat het abonneren (Subscribe) op een applicatie. Dit abonneren doet het bij het lokale CVIS-DDS cliënt-systeem dat aanwezig is in de auto, de lokale CVIS-DDS cliënt “speurt” het netwerk af op zoek naar een gepubliceerde applicatie. Omdat de systemen in de auto's zelf gepubliceerde services kunnen lokaliseren raakt het CVIS-DDS systeem meer gedistribueerd dan in het hiervoor geschetste voorbeeld.

Ook het Publish and Subscribe mechanisme kan toegelicht worden aan de hand van een voorbeeld met het Parkeer-systeem. In dit voorbeeld wordt de automobilist een parkeerplaats toegewezen, echter rijdt deze zelfstandig naar een bekende parkeergarage zonder een garage te selecteren met behulp van de Parkeer-applicatie. De gebruiker start de Parkeer-applicatie dus niet handmatig maar wacht tot de Parkeer-applicatie volautomatisch contact zoekt met een Parkeer-service in de buurt. Dit gebeurt middels het Publish and Subscribe mechanisme. De Parkeer-service is gepubliceerd op het draadloze netwerk van de parkeergarage.

Wanneer een automobilist zijn of haar auto parkeert in een willekeurige parkeergarage, dan wijst de Parkeer-service van de parkeergarage de automobilist automatisch een parkeerplaats toe. Om dit te kunnen bewerkstelligen moet de Parkeer-applicatie verbinding met de Parkeer-service opbouwen. De Parkeer-applicatie moet zichzelf dus als het waren inschrijven op de Parkeer-service. Dit inschrijven doet het bij het CVIS-DDS systeem. Het CVIS-DDS systeem heeft een overzicht van alle diensten die op het lokale netwerk gepubliceerd zijn. Zodra de automobilist het terrein van de parkeergarage betreedt, komt het ook in bereik van het draadloze netwerk van de garage waar een gepubliceerde Parkeer-service operationeel is. Het CVIS-DDS systeem merkt de gepubliceerde Parkeer-service op in het draadloze netwerk van de parkeergarage, waarop deze een signaal geeft aan de Parkeer-applicatie, de Parkeer-applicatie had zich immers ingeschreven op een Parkeer-service. Omdat de Parkeer-service bekend is bij het lokale CVIS-DDS systeem kan de Parkeer-applicatie verbinding maken met de Parkeer-service en zodoende volautomatisch een parkeerplaats toegewezen krijgen.

De CVIS FOAM-omgeving

De afkorting “FOAM” staat voor Framework for Open Application Management. De FOAM-omgeving is de in Java ontwikkelde framework waarin alle CVIS-toepassingen, ook wel bundels genoemd, van het CVIS-project draaien. Dit framework is gebaseerd op de OSGi-standaard, deze OSGi-standaard specificeert een service platform dat op afstand beheerd kan worden. De OSGi-standaard biedt een methode aan om de bundels dynamisch te kunnen installeren, starten, stoppen en de-installeren. Ook is er een service registry gedefinieerd waarin de verschillende toepassingen hun services aan andere toepassingen kenbaar kunnen maken (publiceren), zodat andere toepassingen hierbij kunnen registreren om vervolgens gebruik te kunnen maken van de gepubliceerde services.

De FOAM-omgeving maakt gebruik van de Knoplerfish implementatie van deze standaard. Deze is voor het CVIS-project uitgebreid met het HMC voor het op afstand beheren van de software, de CVIS-DDS bundel, de CALM connection manager, voor het maken van verbindingen via de CVIS implementatie van de CALM ISO-standaard. Ook is er een Human Machine Interface HMI bundel waarin bundels de eigen GUI kunnen registreren zodat interactie met de gebruiker mogelijk wordt gemaakt.

In dit hoofdstuk is de context van de opdracht omschreven. Na het lezen van dit hoofdstuk is duidelijk dat de afstudeeropdracht plaatsvindt binnen Logica onder de practice Technical Software Engineering. Het CVIS-project is in het kort toegelicht evenals het CVIS-DDS systeem dat op conceptueel is beschreven.

Nu de context van dit project duidelijk is kan het proces van dit afstudeerproject beschreven worden. In het volgende hoofdstuk wordt de initiatiefase beschreven.

Bijlage 4: Brief aan medewerkers Technolution B.V.

Geachte heren Gribnau en Baggen,

Voor ik u overval, zal ik mijzelf eerst even voorstellen. Ik ben een TI-student van de Haagse Hogeschool. In het kader van mijn afstuderen voer ik een opdracht uit voor het CVIS-project. Deze opdracht houdt in het kort in dat ik de functionaliteiten en werking van het DDS-systeem onderzoek.

De heer Konijn is mijn begeleider van Logica. Hij staat in de CC-lijst van deze e-mail.

Tijdens mijn onderzoek heb ik een aantal conclusies getrokken. Hierover heb ik een aantal vragen die ik u bij deze voorleg.

Vraag 1:

Ik heb gepoogd door met behulp van de DDS-API een test-applicatie te publiceren. Daarbij heb ik gebruik gemaakt van de `SADescription(saType, URL, isPublish)` constructor. Vervolgens heb ik middels de netwerk-sniffer (Wireshark) gekeken of het DDS-systeem berichten over het netwerk broadcast; Dit gebeurt helaas niet. Dit zou wél moeten doen volgens de documentatie:

“The DDS will initiate a continuous broadcast of the presence of the service provided by the registered application (i.e., the identifier and the URI are broadcasted within the transmission range of the corresponding CVIS Unit (eg a Roadside Unit)).” Bron: D.CVIS.3.3 Architecture and System Specifications.

Op basis van het bovenstaande heb ik de conclusie getrokken dat het publiceren niet werkt. Kunt u dit bevestigen of heb ik een fout gemaakt met mijn test-applicatie?

Vraag 2:

In de DDS-API heb ik geen manier gevonden om een applicatie te “Subscriben” op een dienst. Volgens de literatuur zou de DDS-API over een `Subscribe(sad)`-methode moeten beschikken. Deze heb ik niet terug gevonden in de betreffende DDS-klassen noch in een andere klassen.

Kunt u bevestigen dat het Subscribe mechanisme niet volledig geïmplementeerd is?

Vraag 3:

Ik ben tegen een probleem gelopen bij het aanroepen van de “search(SAQuery, SearchConstraint)” methode. Zodra er gebruik gemaakt wordt van de SearchConstraint ontstaat er een “Java.io.NotSerializableException”.

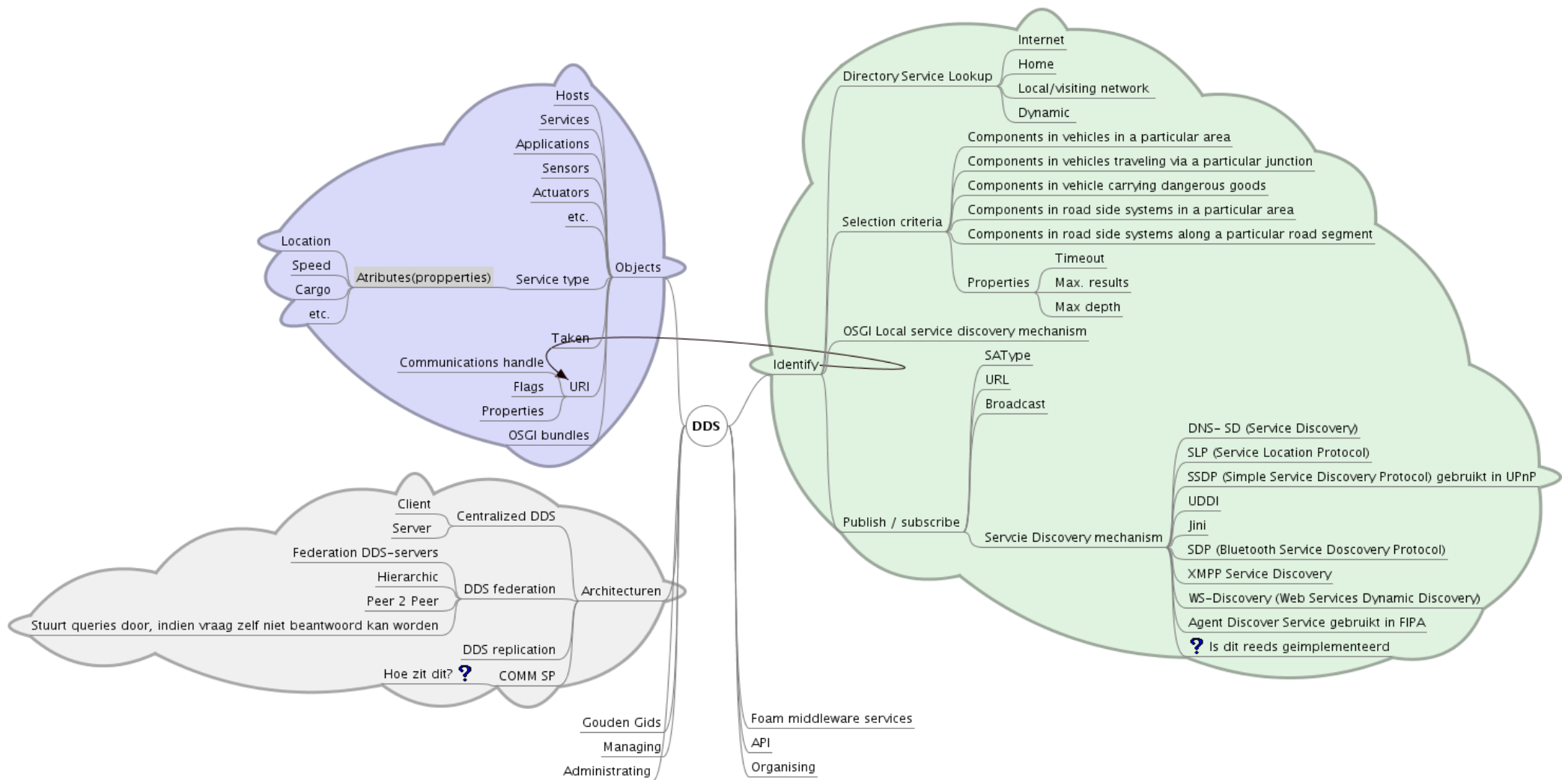
Zijn jullie hiervan op de hoogte en bestaat er een versie waarbij dit probleem verholpen is?

Vriendelijke groet,

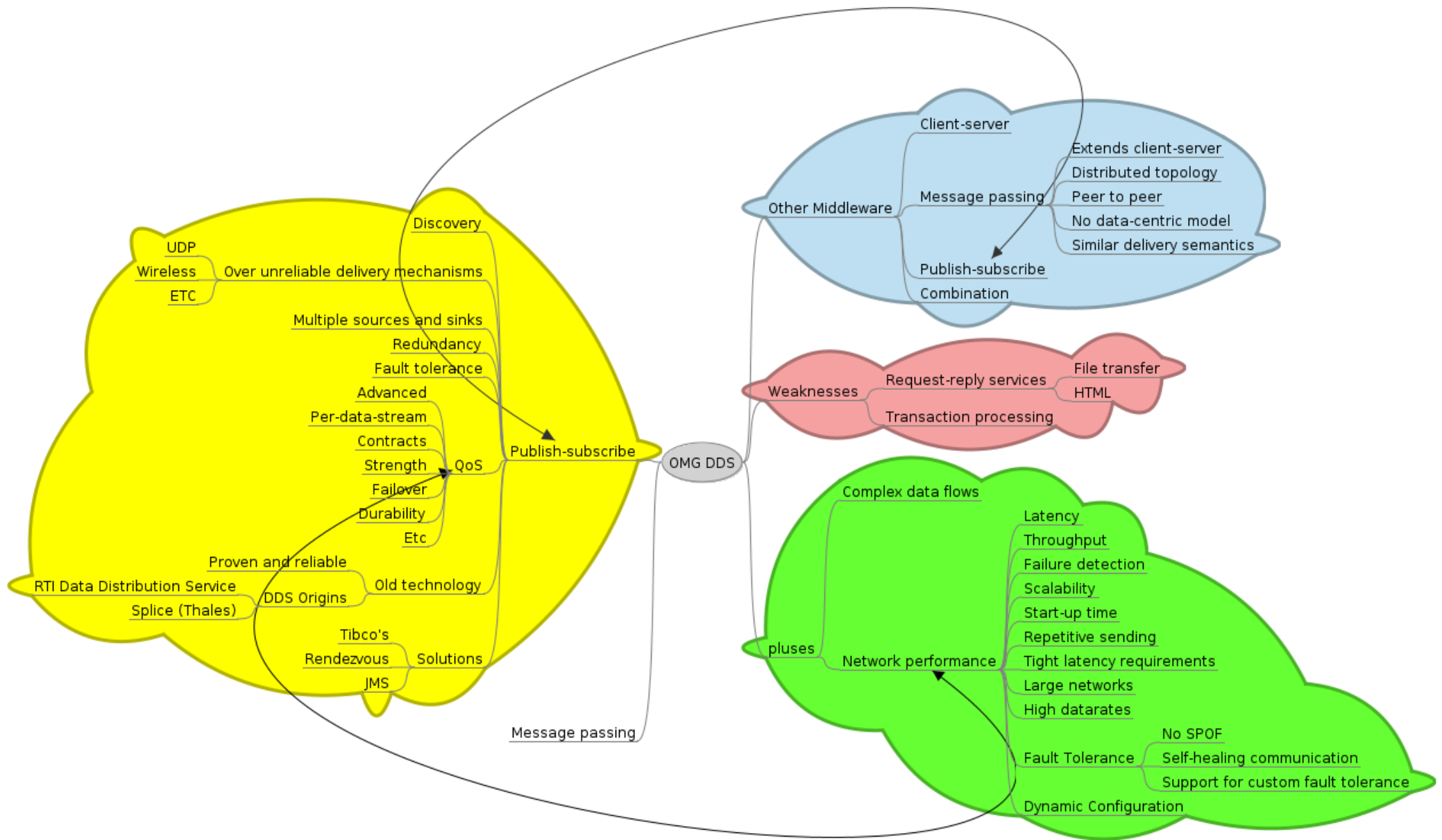
Robert van Reems

Bijlage 5: Mindmaps

Op de volgende pagina ziet u de mindmaps die gemaakt zijn tijdens dit project. Als eerste is de Mindmap van de literatuurstudie naar CVIS-DDS geplaatst. Daarna volgt de Mindmap van de literatuurstudie naar de OMG-DDS standaard.



Figuur 1: Mindmap CVIS-DDS systeem



Figuur 2: Mindmap OMG-DDS