

AFSTUDEERVERSLAG



Titel	: Pelargos Training Coach
Afstudeerblok	: 2004-2.2
Opdrachtgever	: Haagse Studenten Roeivereniging Pelargos
Bedrijfsmentor	: de heer N. Learentveld
Examinatoren	: de heer H. Witkamp & de heer J. van Peski
Afstudeerder	: Geert Jan de Boer
Versie	: 1.0
Plaats	: Den Haag
Datum	: 24 maart 2005

REFERAAT

G.J. de Boer

Afstudeerverslag, de Pelargos Training Coach

Haagse Studenten Roeivereniging Pelargos, Den Haag 24 maart 2005

Dit verslag beschrijft de werkzaamheden die zijn verricht tijdens het afstuderen van Geert Jan de Boer bij de Haagse Studenten Roeivereniging Pelargos te Den Haag. In de periode van 15 november 2004 tot en met 24 maart 2005 heeft het afstudeerproject met de titel “De Pelargos Training Coach” plaatsgevonden.

Het project hield in een overzicht te verkrijgen van de trainingsopbouw tijdens het roeien opdat een coach kan benaderen hoe efficiënt een roeiteam hun energie en kracht gebruikt. Voor het project is gebruikt gemaakt van GPS dat in verbinding stond met een Java-telefoon door middel van een Bluetooth GPS-ontvanger

Project	
Titel	De Pelargos Training Coach
Periode	15 november 2004 – 24 maart 2005
Afstudeerder	
Naam	Geert Jan de Boer
Bedrijf	
Naam	Haagse Studenten Roeivereniging Pelargos
Bedrijfsmentor	De heer N.E. Leerentveld
School	
Naam	Haagse Hogeschool
Sector	Informatica & Informatiekunde
Opleiding	CTIS [Communicatieve en Technische Informatiesystemen]
Examinatoren	De heer H. Witkamp en de heer J. van Peski

INHOUDSOPGAVE

Samenvatting

Voorwoord

1 Inleiding	8
2 GPS uitgelegd	9
2.1 softwarematige nauwkeurigheidssystemen	11
2.2 Gangbare protocollen	12
3 De roeivereniging Pelargos toegelicht	13
3.1 Structuur vereniging	13
3.2 Historie	13
3.3 Plaats afstudeerder binnen Pelargos	14
4 Omschrijving opdracht	15
4.1 Inleiding	15
4.2 Vaststellen probleemstelling	15
4.3 Vaststellen doelstelling	15
4.4 Vaststellen uitgangssituatie	15
4.5 Vaststellen producten	16
4.6 Vaststellen werkzaamheden	16
4.7 Vaststellen planning	17
5 Opstellen plan van aanpak	18
5.1 Inleiding	18
5.2 Projectorganisatie	18
5.3 Afbakening	18
5.4 Te gebruiken methoden en technieken	19
5.5 Risicofactoren en randvoorwaarden	19
6 Definitiefase	20
6.1 Inleiding	20
6.2 Alternatieven	20
6.3 Keuze programmeertaal	21
6.4 Keuze platform	22
6.5 Keuze GPS-ontvanger	23
6.6 Keuze GPS protocol	25
6.7 Keuze softwareprogramma	25
6.8 Gebruik Bluetooth API	26
6.9 BCC (Bluetooth Control Center) uitgelegd	27
7 Architectuurfase	28
7.1 Inleiding	28
7.2 Beeldschermontwerp	28
7.3 Klassediagram	30
7.4 Databaseontwerp	31
8 Ontwikkelfase	33
8.1 Inleiding	33
8.2 Beeldschermontwerp	33
8.3 Opslag GPS data	34
8.4 Verbinding maken met Bluetooth apparaat	34
8.5 Ontvangen GPS informatie van Bluetooth apparaat	37
8.6 Bepalen afgelegde afstand	42
9 Pilot 1 Benaderen nauwkeurigheid van GPS	44
10 Pilot 2 Verifiëren bruikbaarheid programma door de coaches	47
11 Evaluatie	48
11.1 Proces evaluatie	48
11.2 Product evaluatie	48
12 Vervolgstudies	50

VERKLARENDE WOORDENLIJST
GERAADPLEEGDE LITERATUUR

BIJLAGEN

1. Geschikte telefoons.
2. Roeiwater op de Bosbaan in Amsterdam.
3. Roeiwater op de Vliet in Voorburg / Rijswijk / Delft.
4. JAVA support on Pocket PC.
5. GPS en Galileo.
6. Test PDA's door Consumentenbond d.d. september 2004.

SAMENVATTING

Hieronder volgt een korte samenvatting van het proces zoals doorlopen is in dit project. Allereerst is de opdracht vastgesteld in een opdrachtoomschrijving waarbij twee iteraties zijn opgenomen om te voorkomen dat het project direct zou stoppen indien de gestelde eisen niet haalbaar bleken te zijn.

- Eerste iteratie: de nauwkeurigheidseis vanuit de opdrachtgever aan het GPS. Deze moest minimaal 99% zijn bij de wedstrijdafstand (2.000 meter).
- Tweede iteratie : verificatie van het programma op bruikbaarheid door de coaches. Hierdoor zouden mogelijk andere eisen worden gesteld aan het programma.

Bij het vaststellen van de opdrachtoomschrijving bleek uit een eerste verkenning dat de nauwkeurigheidseis van de opdrachtgever aan het programma waarschijnlijk gehaald zou worden. Deze eerste verkenning vond plaats aan de hand van de gegevens die verstrekt worden door de producenten van de verschillende GPS ontvangers.

Uitgaande van de opdrachtoomschrijving is de definitiestudie gehouden waarin de opdracht verder werd uitgewerkt. Daarnaast is in deze fase de architectuur van het programma bepaald door het kiezen van een programmeertaal, een type GPS-ontvanger, het te gebruiken GPS-protocol en het te gebruiken softwareprogramma. Voor meer informatie zie hoofdstuk 6. Meest vermeldenswaardige conclusie uit deze fase was de wijziging van het type platform dat gebruikt zou gaan worden. In plaats van een PDA zou gebruik gemaakt gaan worden van een GSM vanwege de lagere prijs ondanks dat het dezelfde functionaliteiten heeft aangezien gebruik gemaakt wordt van het platformafhankelijk Java.

Daarnaast is in de definitiestudie gekeken naar hoe de afgelegde afstand gemeten zou moeten worden. Indien er namelijk bochten in de afgelegde route liggen dient de afgelegde afstand anders berekend te worden dan als er een rechte route wordt afgelegd. Uit analyse van het roeiwater waarop Pelargos roeit bleek dat deze over het algemeen zeer recht waren. Hierdoor wordt er bij het programma uitgegaan dat er een rechte lijn wordt afgelegd.

Vervolgens kwam de architectuurfase waarin de werking en opbouw van het programma werd bedacht. Tijdens deze fase bleek aan de hand van het eerste beeldschermontwerp dat de bedrijfsmentor tevreden was over de bruikbaarheid van het programma. De bedrijfsmentor is oud-commissaris roeien bij Pelargos en in het bezit van coachcursus A van de Koninklijke Nederlandse Roeibond (KNRB).

Hieruit bleek in de architectuurfase dat de tweede iteratie waarschijnlijk gehaald zou kunnen worden.

Hierna kwam de ontwikkelfase waarin drie testprogramma's zijn gemaakt, aangezien de werking van sommige onderdelen van het programma voor mij nog onbekend waren.

Te weten:

- gebruik van Bluetooth;
- het Record Management System (RMS) dat gebruikt wordt voor de dataopslag. Deze dataopslag is op een GSM anders dan op een computer vanwege de beperkte geheugencapaciteit en processorkracht van een GSM.

Daarnaast is er een testprogramma gemaakt voor het testen van het beeldschermontwerp onder andere vanwege de tweede iteratie.

Met name het testprogramma voor het verbinding maken met een Bluetooth GPS-ontvanger en het uitlezen van de GPS-data kostte meer tijd dan verwacht. Dit kwam onder andere door het gebruik van de mobiele telefoon om het programma te testen: dit was een traag proces. Daarnaast waren er obstakels bij het verbinding maken met en uitlezen van de Bluetooth GPS-ontvanger. Toen na het proberen van meerdere soorten codes van verschillende bronnen (o.a. Sun) de data nog steeds incorrect werd weergegeven op het scherm, is er voor gekozen om via een work around verder te kunnen met het ontwikkelen aan de rest van het programma. In de tussentijd waren verschillende berichten op verschillende fora geplaatst met de vraag op een antwoord.

Hierna is dankzij een tip van een kennis gebruik gemaakt van een Bluetooth dongle voor op de computer. Hiermee kon via de computer direct contact worden gelegd met de GPS-ontvanger in plaats van via de GSM. Op de computer ging het debuggen van het programma veel gemakkelijker aangezien hier meer foutmeldingen worden weergegeven in tegenstelling tot de GSM. Hierdoor konden de fouten tijdens het debuggen boven water worden gehaald, waardoor na aanpassing van de broncode de GPS data correct en continue werden weergegeven.

Vervolgens is gekeken of de eerste iteratie haalbaar was door over een afstand van 2.000 meter de gemiddelde nauwkeurigheid te berekenen. Hieruit bleek dat de nauwkeurigheidseis vanuit de opdrachtgever gehaald werd en dat de definitiestudie niet nogmaals diende te worden uitgevoerd.

Voor de tweede iteratie is het programma niet in de praktijk getest door de coaches aangezien het programma nog niet af is. Het prototype is door de bedrijfsmentor doorgenomen op de mobiele telefoon. Daarnaast is in de architectuurfase een schets van het beeldschermontwerp tijdens één van de eerste meetings met de bedrijfsmentor overlegd en goedgekeurd.

VOORWOORD

Voor u ligt het afstudeerverslag van Geert Jan de Boer. In het kader van het afstuderen heb ik 16 weken een project uitgevoerd bij de Haagse Studenten Roeivereniging Pelargos in Den Haag.

Mede vanuit mijn opleiding CTIS (Communicatieve en Technische Informatie Systemen), het huidige TI (Technische Informatica), vond ik het een zeer interessant project. Met name vanwege het innovatieve karakter van deze opdracht.

Dit onder andere door het gebruik van het GPS niet enkel voor plaatsbepaling maar ook om een beeld te vormen via de GPS data van de verplaatsing van een roeiboot gemeten over een bepaalde tijd.

Allereerst wil ik mijn dank uitspreken voor de begeleiding vanuit de H.S.R.V. Pelargos. Ik wil met name de heer Leerentveld en het bestuur van de H.S.R.V. Pelargos bedanken voor het beschikbaar stellen van hun tijd en diensten. Ik waardeer dit des te meer aangezien dit altijd op vrijwillige basis is geweest.

Daarnaast wil ik ook de beide examinatoren van de Haagse Hogeschool, te weten de heren Witkamp en Van Peski, bedanken voor de begeleiding en feedback bij het schrijven van dit eindverslag.

Den Haag, maart 2005

HOOFDSTUK 1. INLEIDING

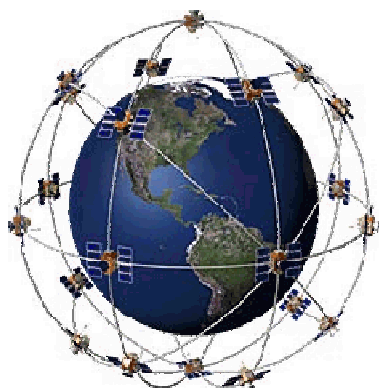
Dit verslag heeft als doel de lezer een duidelijk beeld te geven van het proces zoals doorlopen in het afstudeerproject. In dit verslag komt naar voren wat het project inhoudt, hoe het project is aangepakt, wat de problemen waren, welke oplossingsrichtingen er waren en welke keuzes aan de hand hiervan tijdens het project gemaakt zijn.

In hoofdstuk 2 wordt uitgelegd hoe het GPS in elkaar zit, welke nauwkeurigheidssystemen er hiervoor beschikbaar zijn en via welke protocollen een GPS-ontvanger kan communiceren. Vervolgens wordt in hoofdstuk 3 een beeld gegeven van de H.S.R.V. Pelargos en de plaats van de afstudeerder in deze vereniging.

Hierna zal in hoofdstuk 4 de opdracht worden omschreven aan de hand waarvan een plan van aanpak is opgesteld, welke besproken wordt in hoofdstuk 5. De verschillende fases zoals ze doorlopen zijn bij het uitvoeren van het project zullen worden besproken in de hoofdstukken 6 tot en met 8.

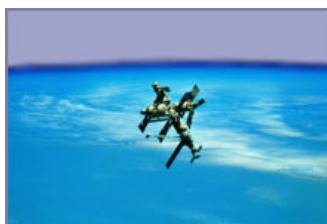
In hoofdstuk 9 en 10 zullen de pilots besproken worden. Vervolgens zal in de evaluatie in hoofdstuk 11 het oordeel over mijn aanpak van dit project worden besproken. Dit zal worden opgesplitst in twee delen, het proces en het product.

HOOFDSTUK 2. GPS UITGELEGD



GPS staat voor **Global Positioning System** en is in het leven geroepen en wordt onderhouden door de Amerikaanse overheid. Het was in eerste instantie hoofdzakelijk bedoeld voor militaire doeleinden.

Het systeem bestaat uit een netwerk van 24 satellieten die rond de aarde draaien. Ieder van deze satellieten draait twee keer per dag in een vaste baan om de aarde en zendt een uniek signaal uit. Met een GPS-ontvanger kunnen deze signalen worden opgevangen en verwerkt. Door op een bepaald moment de positie van de satellieten (die zij door de ruimte zenden) te vergelijken met de tijd op het moment van aflezen, kan de ontvanger exact bepalen waar hij zich op dat moment bevindt. Door immers het snijpunt te nemen van minimaal drie satellieten krijg je een bepaald punt op de aardbol, en dat is je huidige positie. Hoe meer satellieten gebruikt worden bij deze berekening, hoe beter en nauwkeuriger de bepaling van de locatie van de ontvanger.



De GPS-ontvanger ziet alleen de satellieten die op een bepaald moment boven de horizon staan: dat zijn er normaal minimaal 6. Deze satellieten zijn overal op de wereld te ontvangen. GPS vertelt je dus altijd tot op enkele meters nauwkeurig waar je op aarde bent. Steeds wordt de positie opnieuw berekend waardoor het mogelijk is om snelheid, richting en hoogte exact te bepalen. GPS kan je helpen bij het vinden van je locatie en de

richting die je zou moeten volgen om bij een bepaald doel te komen. Als een GPS-ontvanger weet waar je bent en waar je naartoe wilt, dan kan hij (mits je beweegt!) bepalen of je in de goede richting verplaatst.

Het GPS-netwerk is een uitstekend systeem voor wereldwijde plaatsbepaling. De standaard configuratie levert sinds het moment dat de Amerikaanse overheid de zogenaamde SA (Selective Availability) heeft uitgeschakeld een nauwkeurigheid van enkele tientallen meters. Om een hogere nauwkeurigheid te bereiken zijn al sinds het ontstaan van GPS verschillende methodes bedacht om de gemeten positie te corrigeren. Deze methodes worden hieronder besproken. Meer informatie over de nauwkeurigheid & ontwikkeling van GPS zijn te vinden in bijlage 5.

DGPS

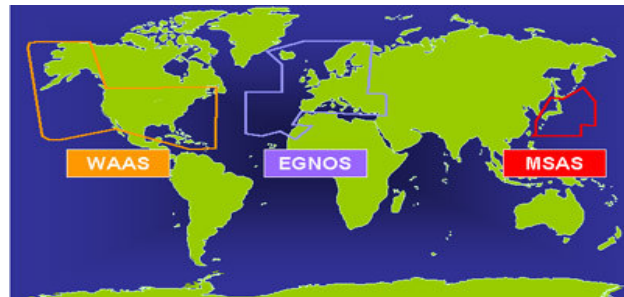
DGPS (Differential Global Positioning System) wordt gebruikt om de positie van de GPS-ontvanger nog nauwkeuriger weer te kunnen geven. Hierbij wordt gebruikt gemaakt van de referentie van verschillende grondstations. Om contact te kunnen maken met deze grondstations moet echter een beacon ontvanger aan de GPS worden toegevoegd. Daarnaast zijn er GPS-ontvangers beschikbaar waar een beacon receiver in geïntegreerd zit. Een voorbeeld hiervan ziet u hiernaast. Kosten van deze beacon receivers loopt echter in de vele honderden euro's.



WAAS/ EGNOS

WAAS is het Wide Area Augmentation System en is een soort softwarematig ingebouwde DGPS uitvoering. Door middel van enkele grondstations en 'multi-purpose'-satellieten wordt de positie van de GPS-ontvanger veel nauwkeuriger weergegeven dan men tot voor kort gewend was. Voorlopig werkt WAAS alleen in de VS en worden Europese bezitters van een GPS-ontvanger die WAAS ondersteunt aangeraden om de WAAS-functie niet aan te zetten. Het WAAS is gefundeerd op het FAA¹ om de nauwkeurigheid en sterkte van het signaal te bewaren om te vliegen in minder goeie meteorologische omstandigheden, vooral bij landingsfases. Dit systeem zendt informatie en gecorrigeerde data door vanaf de grond. De ontvangers op de grond kunnen de info satellietinfo makkelijk interpreteren en verbeteren indien nodig. Op dit ogenblik is dit systeem nog in volle opbouw, tests van de FAA hebben uitgewezen dat een nauwkeurigheid van 3 meter haalbaar is.²

EGNOS is het European Geostationary Navigation Overlay Service en is de Europese tegenhanger van het Amerikaanse WAAS-systeem dat dient om de nauwkeurigheid van GPS-signalen te verhogen en de integriteit van het systeem. Het wordt ontwikkeld door ESA (European Space Agency).



Beide netwerken verkeren nog in een testfase en zijn nog niet operationeel. Verwacht wordt dat ze in (de loop van) 2005 operationeel worden.³

Nauwkeurigheid

De nauwkeurigheid van het oorspronkelijke GPS-systeem, was onderworpen aan de nauwkeurigheid degradatie opgelegd door de Amerikaanse regering. Deze Selective Availability (SA) heeft een nauwkeurigheid van 100 meter.

De typische GPS positie nauwkeurigheid zonder SA is 15 meter. Typische differentie GPS (DGPS) positie nauwkeurigheid is 3-5 meter. De typische WAAS en EGNOS nauwkeurigheid is ongeveer 7 meter. Hieruit blijkt dat het softwarematige EGNOS dan wel WAAS niet onderdoen voor DGPS, waarvoor een aparte beacon ontvanger moet worden aangeschaft. Als bron hiervoor zijn de gegevens gebruikt zoals deze verstrekt zijn door verscheidene producten van GPS-ontvangers.

¹ Federal Aviation Administration (FAA), Amerikaanse overheidsdienst met als missie "to provide the safest, most efficient aerospace system in the world." Meer informatie op: <http://www.faa.gov/index.cfm>

² Voor meer info: <http://GPS.faa.gov/Programs/WAAS/waas.htm>

³ Meer info over EGNOS is te vinden

op: http://www.esa.int/export/esaSA/GGG63950NDC_navigation_0.html

2.1 GPS softwarematige nauwkeurigheidssystemen

Naast het hardwarematige nauwkeurigheidssysteem DPGS bestaan er ook softwarematige nauwkeurigheidssystemen die ingebouwd zijn in de chipset van een GPS-ontvanger. In elke GPS-ontvanger zit een Sirf⁴ hoofd chipset welke bepaalt hoe de signalen van een satelliet worden ontvangen. Ook bepaalt het of er naar alle satellieten wordt geluisterd die in het bereik zijn of dat alleen de sterkste signalen worden gebruikt en de rest genegeerd wordt. Hiervoor dienen echter wel minstens drie signalen van satellieten gebruikt te worden waarbij geldt dat hoe meer signalen beschikbaar zijn, des te beter de positie kan worden bepaald. Van deze hoofd chipset zijn in de loop van de tijd verschillende, nieuwere versies ontwikkeld. In dit hoofdstuk worden de versies van de chipset besproken die momenteel gangbaar zijn, waardoor een keuze kan worden gemaakt welk type chipset het best gebruikt kan worden.⁵

Sirfstar IIe: de standaard in prestatie en flexibiliteit voor hoge volumes.

De Sirfstar IIe was het eerste product om de Sirfstar II architectuur te gebruiken – wat de standaard werd voor GPS prestatie. Het heeft geïntegreerde WAAS, EGNOS. Dit helpt gebruikers met snelle, nauwkeurige GPS die soepel genoeg is om gebruikerstaken naar de bewerker te integreren.

Sirfstar IIe/LP: de nieuwe meer energiezuinige van de Sirfstar IIe standaard GPS.

Indien een apparaat vervaardigd of ontworpen wordt voor een GPS die batterijen gebruikt, is de Sirfstar IIe/LP de oplossing vanwege het veel lagere stroomverbruik.

Sirfstar IIIt: de host-based GPS chipset.

De Sirfstar IIIt oplossing staat vooral voor lagere kosten aan componenten door het delen van de systeem processor en geheugen bronnen. De chipset verwerft en spoort satellieten op, stuurt dan ruwe maten naar de Sirfnav software. De Sirfnav softwaremodule berekent positie tijd en snelheid. Voor betere nauwkeurigheid en navigatie betrouwbaarheid kunnen WAAS correctie door Sirfdrive software op de GPS coördinaten toegepast worden.

Sirf Xtrac

Door middel van zeer gevoelige software vergroot de Sirf Xtrac het bereik en nauwkeurigheid van de GPS. Eerst worden de satellieten met de sterke signalen gebruikt om de GPS positie te bepalen, waarna met behulp van de zwakkere signalen deze positie wordt verfijnd.

Zoals reeds eerder verteld betekent het gebruik van meerdere signalen bij het berekenen van de GPS positie dat de plaatsbepaling nauwkeuriger kan geschieden. Des te meer signalen des te beter de plaatsbepaling.

De nauwkeurigheid van een GPS-ontvanger met Sirf Xtrac ligt tussen de 5 tot 25 meter, afhankelijk van het aantal GPS-satellieten dat gebruik kan worden bij de berekening van de positie. De nauwkeurigheid van 25 meter zal met name gelden bij dichtbebouwde gebieden waarbij het zicht op de lucht beperkt is. Deze onnauwkeurigheid van 25 meter lijkt veel, maar er dient bij te worden gezegd dat een GPS-ontvanger met EGNOS of WAAS een kleiner bereik heeft. Dit geldt met name in dichtbebouwde gebieden of bij weinig zicht op de open lucht (bijvoorbeeld achter het raam van een huis of in een broekzak). XTrac heeft eerder bereik in deze gebieden, echter de nauwkeurigheid is minder goed.

Aangezien het programma zal worden gebruikt op open roeiwater zal de nauwkeurigheid van XTrac eerder bij de 5 meter liggen dan bij 25 meter aangezien de nauwkeurigheid aan de hand van meer satellieten kan worden berekend.

⁴ Bedrijf dat GPS chipsets produceert. Meer informatie op : www.sirf.com/about.html

⁵ Bron : <http://www.pocketGPS.co.uk/xtracornoxtrac.php>

2.2 GPS Protocollen

Om een beeld te krijgen van de protocollen waar GPS gebruik van maakt, is hieronder een overzicht samengesteld:

NMEA

NMEA staat voor National Marine Electronics Association een Amerikaanse belangenorganisatie die opereert op het gebied van maritieme elektronische apparatuur.

NMEA-0183 is een industriestandaard om verbindingen tussen verschillende (maritieme) apparaten, waaronder GPS, op goede wijze te laten verlopen. Onder de NMEA-0183 standaard bestaan alle gebruikte tekens uit printbare ASCII tekst (plus carriage return en linefeed). NMEA-0183 wordt verzonden op 4800 baud. De data worden verzonden in de vorm van "zinnen". Elke zin begint met een "\$", een twee letterige "zender ID", een drie letterige "zins ID", gevolgd door een aantal data velden, gescheiden door komma's, en afgesloten door een optionele checksum, alsmede een carriage return annex linefeed.

Een zin kan tot 82 tekens bevatten, inclusief de "\$" en de CR/LF. Als de data voor een veld niet beschikbaar zijn, wordt het veld gewoon weggelaten, maar de komma's die hem begrenzen worden wel verzonden, echter zonder iets daar tussen.

Omdat sommige velden een variabele omvang hebben, of weggelaten mogen worden (zoals hierboven vermeld staat), moet de ontvanger de gewenste data velden localiseren door de komma's te tellen, en niet via de positie van de letter in de zin.

De standaard staat fabrikanten toe om eigen 'zin' formaten te definiëren. Deze zinnen beginnen met "\$P", dan een 3 letterige fabrikanten ID, gevolgd door de data die de fabrikant wil, volgens de algemene regels voor de standaard zinnen.

Enkele bekende zender ID's zijn:

GP	Global Positioning System ontvanger
LC	Loran-C ontvanger
OM	Omega Navigatie ontvanger
II	Integrated Instrumentation (b.v. een AutoHelm Seataalk systeem)

Sirf Binary

Protocol ontworpen door de producent van de GPS chipsets, Sirf. Sirf Binary wordt door vele GPS-ontvangers ondersteund maar niet door alle. Een overzicht hiervan is te vinden in hoofdstuk 6.5.

Garmin

Protocol gebruikt door producent van GPS-ontvangers Garmin en wordt alleen gebruikt voor hun eigen ontvangers.

HOOFDSTUK 3. DE ROEIVERENIGING PELARGOS

Dit hoofdstuk is bedoeld om een beeld te geven van de vereniging waarbij de afstudeeropdracht is uitgevoerd.

3.1 Structuur vereniging

De interne structuur van de H.S.R.V. Pelargos bestaat momenteel uit het bestuur en de commissies.

Bestuur

Per 30 september 2004 is het derde bestuur van de H.S.R.V. Pelargos geïnstalleerd, bestaande uit de voorzitter, secretaris, penningmeester, commissaris roeien en commissaris sponsoring.

Commissies

Er zijn bij de H.S.R.V. Pelargos vier commissies geïnstalleerd die het bestuur ondersteunen bij de uitvoer van het beleid. Het gaat om de publicatiecommissie, sponsorcommissie, roeicommissie en de activiteitencommissie.

3.2 Historie

In **2000** werd er gesproken over de mogelijkheid van een studenten roeivereniging in Den Haag. Tot dan toe was het via het Bureau Hogeschoolsport mogelijk een vijftal instructielessen te volgen: twee maal in het najaar en tweemaal in het voorjaar.

Op initiatief van Peter in den Bosch, medewerker Hogeschool bureau sport, kwamen in september 2000 een aantal studenten bijeen om te praten over een studentenroeivereniging. Tot ongeveer juni 2001 vonden er ongeveer om de maand een aantal bijeenkomsten plaats. Er werd over velerlei zaken gesproken variërend van structuur, statuten, administratie, ledenbinding, roeitechnische zaken, accommodatie op korte en lange termijn, PR en Communicatie, financiën, sponsoring en subsidies. Dit alles werd gebundeld in een operationeel plan.

In de **zomer van 2001** ging de volgende fase van start. Het operationeel plan moest nu ook daadwerkelijk uitgevoerd gaan worden. De "praatgroep" werd een werkgroep. Er werd een aantal taakgebieden aangewezen en er werd vaker en regelmatig vergaderd.

De instructielessen werden vanaf dit moment ook uitgevoerd door de H.S.R.V. Pelargos zelf in overleg met bureau Hogeschoolsport. Er werden borrels georganiseerd voor deze nieuwe groep toekomstige Pelargianen. Naast de instructielessen werden er ook ploegen gevormd. In het roeiseizoen 2001-2002 namen zij voor het eerst (succesvol) deel aan nationale competitieve roeiwedstrijden. Hiermee kreeg de H.S.R.V. Pelargos het karakter van een echte studenten roeivereniging.

Na een training werd op **2 mei 2002** de eerste Algemene Ledenvergadering van de H.S.R.V. Pelargos gehouden. Hier werd een gedeeltelijk bestuur benoemd en werden Gerard Bakker, Joep de Groot en Niels Leerentveld aangewezen om de dan vastgestelde statuten te deponeren bij de notaris. Vanaf 2 mei 2002 was de H.S.R.V. Pelargos dus een studenten roeivereniging in oprichting.

Op **14 augustus 2002** werden de statuten gedeponneerd en was de H.S.R.V. Pelargos officieel opgericht. Het eerste bestuur bestond uit: Gerard Bakker (voorzitter extern), Joep de Groot (voorzitter intern), Frédérique Custers (penningmeester), Ilona Mikkers (secretaris), Sander Haaksma (commissaris PR & sponsoring) en Niels Leerentveld (commissaris roeien).

Het eerste jaar stond in het kader van het vormen van de vereniging. Er werd een verenigingsweekend gehouden, ploegenborrels en theorieavonden georganiseerd. Er werden ploegen opgericht waaronder: Fortuna, de Phoenixians, Blicksem, MEENUS, Scylla. Er zat een zeer stijgende lijn in het aantal wedstrijden waaraan deel werd genomen.

Tijdens de oprichtingsreceptie in het najaar ontving de H.S.R.V. Pelargos van de Gemeente een boot in "eeuwige bruikleen". Deze wordt naar Floris IV vernoemd, de "stichter" van de stad Den Haag. Met deze boot werd begin september 2003 het officieuze Haagse Kampioenschap gewonnen in een strijd tegen De Laak.

Op **30 september 2003** vond voor het eerst een wisseling van het bestuur van de H.S.R.V. Pelargos plaats. Het nieuwe bestuur bestond uit Jort de Bosch Kemper (voorzitter), Roeland Wijkkel (vice-voorzitter), Steven van Westenbrugge (penningmeester), Anouk de Hoog (secretaris) en Dinand Stufkens (commissaris roeien).

Het begin van het verenigingsjaar 2003-2004 stond in het teken van structuur en cultuur. Er werden meerdere commissies gevormd waarin de meeste Pelargianen actief deelnamen.

Op **30 september 2004** vind de tweede bestuurswisseling van de H.S.R.V. Pelargos plaats. Het nieuwe bestuur bestaat uit Johannes Lijnse (voorzitter), Nelleke Aantjes (secretaris/vice-voorzitter), Geert Jan de Boer (penningmeester), Remco Boucherie (commissaris roeien) en Mark Groenewoud (commissaris sponsoring). Het verenigingsjaar 2004-2005 staat in het teken van verdere structuur en groei. Met name de interne communicatie en structuur krijgt dit jaar de nodige aandacht, daarnaast worden er nog meer roeiteams gevormd.

3.3 Plaats afstudeerder binnen Pelargos

Ik ben zelf sinds meerdere jaren betrokken bij de H.S.R.V. Pelargos. Naast het roeien ben ik actief geweest in verschillende commissies; momenteel zit ik in het bestuur.

HOOFDSTUK 4. OMSCHRIJVING OPDRACHT

Dit hoofdstuk bevat de opdrachtoomschrijving die als basis heeft gediend voor dit project.

4.1 Inleiding

Roeien is geen sport waarbij enkel (brute) kracht vereist is om tot een goede eindtijd te komen op een wedstrijd. Een goede techniek is zelfs veel essentiëler!

Om een goed beeld te krijgen of er volgens een goede techniek geroeid wordt kun je de coach vragen of hij langs de kant mee wil fietsen tijdens een training om te kijken of de roeier(s) een goede roeihaal en beweging maken. Daarnaast kun je met een tempoteller en eventueel een 'low drag' impellertje (soort propeller) aan de boot installeren om te kijken wat resp. het tempo en de snelheid is waarmee geroeid wordt.

4.2 Vaststellen probleemstelling

Wat dan echter niet gemeten/weergegeven (kan) worden, is hoe de opbouw van de gehele training of wedstrijd gegaan is. Dit is voor een coach van belang om te zien of een wedstrijd goed is opgebouwd. Hij kan dan benaderen hoe efficiënt het roeiteam hun energie & kracht heeft gebruikt en kan aan de hand hiervan verbeterpunten doorgeven aan het roeiteam. Daarnaast wil je een overzicht hebben van het tempoverloop waarmee de roeier heeft geroeid. Tot slot wil een coach graag tijden meten over (wedstrijd)-afstanden. Op een open roeiwater kun je echter niet bepalen wanneer de wedstrijdafstand is overbrugd.

4.3 Vaststellen doelstelling

De Pelargos Training Coach zal zich moeten gaan onderscheiden van andere hulpmiddelen bij het roeien door via het gebruik van een PDA (Pocket PC) met GPS;

- een overzicht te geven van de trainingsopbouw (via snelheden) opdat een coach kan benaderen hoe efficiënt het roeiteam hun energie en kracht heeft gebruikt.
- de tijden over een bepaalde (wedstrijd)-afstand te bepalen. Je kunt met een "normale" timer namelijk niet bepalen of je een bepaalde afstand al hebt afgelegd.
- (optioneel) tellen van tempo door middel van een extra sensor in een sliding/bankje

De verzamelde data moeten op het eind van de training / wedstrijd kunnen worden aangevuld met extra gegevens zoals de weercondities e.d.. Vervolgens moeten de data geëxporteerd kunnen worden naar een PC en dienen per team/ coach te onderscheiden zijn.

4.4 Vaststellen Ausgangssituatie

Welke (onderzoeks-)resultaten zijn bekend ?
haalbaarheidsstudie

Door een studente geodesie aan de faculteit Civiele Techniek en Geowetenschappen van de TU Delft is een haalbaarheidsstudie "GPS&Sport" uitgevoerd.

Hierin is gekeken of bij gebruikmaking van een draagbare GPS-ontvanger de positie en snelheid van (buiten)sporters met een dusdanige nauwkeurigheid te bepalen zijn dat de verrichte sportprestaties succesvol kunnen worden geanalyseerd.

nauwkeurigheid

Op een website (www.vierzonder.nl/GPS_informatie.htm) staat een verslag van een klein onderzoek naar de nauwkeurigheid van GPS voor bepalen van afstand en snelheid bij roeien.

4.5 Vaststellen producten

Voor de verschillende fases zullen de volgende producten worden opgeleverd:

- Definitierapport
- Ontwerprapport
- 2 pilot-rapporten (incl. testplan & testrapport)
- Softwareprogramma voor een PDA
- Documentatie over product

4.6 Vaststellen werkzaamheden

De beschrijving van de werkzaamheden is opgesplitst in twee delen te weten de voorbereiding en het ontwikkelen van het programma op de PDA.

Voorbereiding

1. Werking GPS duidelijk krijgen; hoe kun je de plaats berekenen en hoe kun je benaderen hoe groot de onnauwkeurigheden van de metingen zijn;
2. Werking EGNOS duidelijk krijgen;
3. Werking DGPS duidelijk krijgen;
4. Ontwikkelomgeving bepalen;
5. Gebruikersomgeving bepalen (i.s.m. met coaches & opdrachtgever);
6. Benaderen van nauwkeurigheid van beschikbare GPS-ontvanger aan de hand van andere meetresultaten (zie punt 6: uitgangssituatie).

Ontwikkelen programma op de PDA

7. Bepalen hoe gemeten data moet worden opgeslagen in het geheugen van de PDA. N.B. Er zit een automatisch log-functie op de meeste PDA's. Nadeel hiervan is echter dat de data naar de seriële poort worden verstuurd.
8. Communicatie tussen PDA en (een programma op de) pc voor het uitlezen en archiveren van gemeten data.
9. Implementeren van timer welke gekoppeld is aan de meetdata van de GPS, opdat tijden over een wedstrijdafstand gemeten kunnen worden.

In de loop van het project is bij de definitiefase gekozen voor gebruik van een ander platform dan de PDA. Meer informatie hierover is te vinden in hoofdstuk 6.4. Hierdoor gewijzigde werkzaamheden zijn in dit zelfde hoofdstuk beschreven.

4.7 Planning

Aan de hand van de werkzaamheden zoals besproken in de vorige paragraaf is de volgende planning tot stand gekomen.

Begin : 15 november 2004

Inleveren verslag : 24 maart 2005

- Week 1 : Plan van aanpak en opdrachtoomschrijving
- Week 2 : Plan van aanpak en opdrachtoomschrijving
- Week 3 : Werking GPS duidelijk krijgen; hoe kun je de plaats berekenen en hoe kun je benaderen hoe groot de onnauwkeurigheden van de metingen zijn
- Week 4 : Werking EGNOS en WAAS duidelijk krijgen
- Week 5 : Ontwikkel, - en gebruikersomgeving bepalen
- Week 6 : Benaderen van nauwkeurigheid van beschikbare GPS-ontvanger
- Week 7 : Bepalen hoe gemeten data moet worden opgeslagen in het geheugen van de PDA.
- Week 8 : Communicatie tussen PDA en (een programma op de) pc voor het uitlezen en archiveren van gemeten data.
- Week 9 : Implementeren Programma
- Week 10 : Implementeren Programma
- Week 11 : Testen en verbeteren programma
- Week 12 : Schrijven documentatie
- Week 13 : Schrijven eindverslag
- Week 14 : Schrijven eindverslag
- Week 15 : Afronden Verslag
- Week 16 : Uitloop

HOOFDSTUK 5. OPSTELLEN PLAN VAN AANPAK

5.1 Inleiding

Voor het afstudeerproject is een plan van aanpak opgesteld aan de hand van de vastgestelde opdracht zoals gedocumenteerd in hoofdstuk 4. Het proces van het opstellen van het plan van aanpak zal in dit hoofdstuk worden toegelicht.

5.2 Projectorganisatie

De afstudeerder zal het project individueel uitvoeren waarbij hij zal worden begeleid door de bedrijfsmentor, de heer Leerentveld.

5.3 Afbakening

Er zal een programma ontworpen en ontwikkeld worden waarmee de opbouw van een roeitraining of -wedstrijd gemeten en afgelezen kan worden. Daarnaast moeten de tijden van een roeiwedstrijd ermee geklokt kunnen worden. Wat ik niet zal doen, is het implementeren van een extra input sensor welke het tempo meet van de roeiers door middel van het aanbrengen van een sensor aan de bankjes van de slag (de roeier die het tempo bepaald).

Eisen en wensen

Het product zal aan de volgende eisen en wensen moeten voldoen:

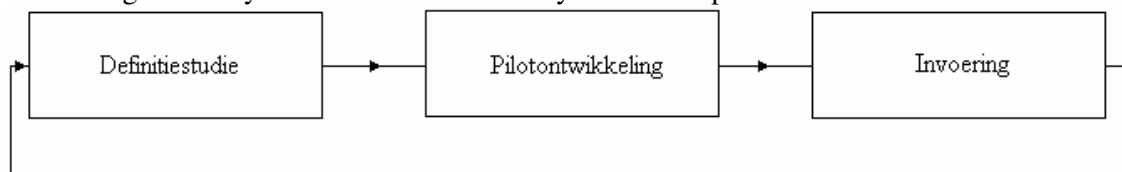
- Het programma dient te worden ontworpen en ontwikkeld voor een PDA.
- Metingen over de snelheid van de boot en de afgelegde afstand zal gebeuren met een GPS-module, die gekoppeld is aan de PDA.
- Tijdens het meten dient er al een real-time weergave afgebeeld te worden van het verloop van de training of de wedstrijd.
- Een overzicht van de metingen dient direct na de training te kunnen worden opgevraagd.
- De input die tijdens een training is verzameld, dient geëxporteerd te kunnen worden naar een PC. Onduidelijk of de input direct in bestaand formaat bezet kan worden (bijv. Excel) of dat er een speciaal programma voor op de pc moet worden gemaakt
- De verkregen input van een training dient aangevuld te kunnen worden met informatie over: de weersomstandigheden, instruerende coach, teamnaam en het type boot waarin is geroeid.
- Het moet mogelijk zijn om tijdens een training een wedstrijdafstand af te leggen. Hierbij dienen dan zowel de wedstrijd,- als de trainingsgegevens bewaard te kunnen blijven. De data van verschillende wedstrijden of opzetjes moeten bij de weergave achteraf van elkaar te onderscheiden zijn.
- Voor de meting van de nauwkeurigheid van GPS zoals bedoeld bij de eerste iteratie in paragraaf geldt de wetenschappelijke nauwkeurigheid (99%) als eis. Hierbij uitgaande van de wedstrijdafstand (2.000 meter).
- Het programma zal gemaakt worden voor studenten, daarom zal moeten worden getracht de kostprijs van gebruik van het programma zo laag mogelijk te houden. Het gaat hier dan met name om de aanschaf van apparatuur.
- Bij het kiezen van de architectuur (programmeertaal, platform en GPS-ontvanger) dient rekening te worden gehouden met het mogelijk breed inzetbaarheid van het programma.
- Het beeldschermontwerp (GUI) dient eenvoudig te bedienen zijn.
- De eerste versie van het programma dient door de coaches te worden geverifieerd op grond van bruikbaarheid onder andere vanwege nauwkeurigheid van het GPS

5.4 Te gebruiken methoden en technieken

Aangezien tijdens de opleiding van de afstudeerder geen methoden als RUP en IAD zijn behandeld is, op advies de vakdocent, een afspraak gemaakt met de heer De Vreught. Hij is bekend met verschillende soorten methoden en technieken. Na overleg van de eisen van het project (zie hoofdstuk 5.3) is uitgekomen bij IAD als leidraad voor met name de iteratiestrategie evolutionair ontwikkelen. Gekozen is voor IAD vanwege de iteratieve methode. Daarnaast kon op de bedrijfsmentor worden terug gevallen aangezien hij vanuit zijn opleiding bekend is met deze methode.

De heer De Vreught adviseerde om twee iteraties in te voegen. De eerste vanwege de nauwkeurigheidseis met betrekking tot GPS. De tweede vanwege mogelijke verandering van de eisen aan het programma nadat de coaches het programma hebben geverifieerd op bruikbaarheid.

De keuze voor evolutionair ontwikkelen als iteratiestrategie is vanwege de twee iteraties die bij de pilotontwikkeling of bij de invoering zouden kunnen leiden tot bijstelling of verdere uitbreiding van de systeemeisen dan wel het systeemconcept.



* Proces zoals doorlopen wordt bij evolutionair ontwikkelen van IAD. Indien bij een iteratie blijkt dat de vastgestelde eisen van hoofdstuk 5.4 moeten worden aangepast, kan via deze methode teruggekeerd worden naar de definitiestudie.

Door evolutionair ontwikkeling als iteratiestrategie te kiezen zou voorkomen kunnen worden dat het project direct zou stoppen indien verder in de cyclus zou blijken dat de in eerste instantie gestelde eisen niet haalbaar zouden zijn. Het is dan echter mogelijk om, in overleg met de opdrachtgever, te kijken of het mogelijk is om de eisen te versoepelen indien nodig. Deze mogelijkheid is vooral vanwege de eerste iteratie; het benaderen van de nauwkeurigheid van GPS tijdens roeien. Indien zou blijken dat de nauwkeurigheid onvoldoende is, dan is het mogelijk om terug te keren naar de definitiestudie. In mindere mate is het ook voor de tweede iteratie. Indien zou blijken dat het GPS niet nauwkeurig genoeg zou zijn dan is het mogelijk om de eisen aan de werking van het programma aan te passen. Daarnaast kunnen in de tweede iteratie functionele eisen aan het programma worden aangepast. Een coach kan namelijk bij het proberen van het programma dusdanige opmerkingen hebben die kunnen leiden tot aanpassing van de eisen zoals gesteld in definitiestudie.

Vanwege deze korte pilots was de iteratiestrategie van IAD geschikt voor dit project en is, in overleg met de bedrijfsmentor, gekozen voor deze methode.

5.5 Risicofactoren en randvoorwaarden

Het is een innovatief programma, welke gebruik zal gaan maken van GPS. Ondanks het feit dat enkele jaren geleden de Amerikaanse overheid het SA (Selective Availability)-systeem van GPS heeft uitgeschakeld, is het niet zeker of GPS nauwkeurig genoeg is voor gebruik bij dit project. Zie voor meer informatie over SA bijlage 5.

Daarnaast is er binnen Pelargos weinig technische begeleiding qua programmeren aangezien geen van de leden een ICT-opleiding volgt anders dan VIA of IVIT, welke meer op vormgeving zijn gericht. Indien er tijdens het programmeren technische problemen naar boven komen is er binnen de vereniging dus niemand om op terug te vallen.

De laatste risicofactor is de iteratiestrategie evolutionair ontwikkeling van IAD. Het is goed mogelijk dan al na de eerste pilot tot stopzetting van het project wordt besloten, bijvoorbeeld omdat de verwachtingen niet realistisch blijken of omdat het risico te hoog wordt geacht.

HOOFDSTUK 6. DEFINITIEFASE

6.1 Inleiding

In deze fase zal het programma van eisen worden opgesteld aan de hand van de vastgestelde opdracht in het vorige hoofdstuk. Het zal een antwoord geven op de vraag wat de verwachtingen zijn van de opdrachtgever. Daarnaast zal de opdracht worden gedefinieerd qua uitgangspositie van software, programmeertaal, type GPS-ontvanger en type platform. Ook zullen potentiële alternatieven worden besproken.

6.2 Alternatieven

Doel van de opdracht is onder andere om de tijden over een wedstrijdafstand te bepalen. Dit met behulp van het GPS-systeem dat door middel van een GPS-ontvanger in verbinding staat met een Personal Digital Assistant (PDA). Om te kijken of het bereiken van dit doel ook op een andere manier mogelijk is, is er gekeken naar mogelijke alternatieven. Dit aan de hand van de eisen zoals opgesteld in hoofdstuk 5.3. Hieronder worden de drie alternatieven vermeld die besproken zijn met de bedrijfsmentor.

Stopwatch

De oude vertrouwde manier van tijdmetingen. Echter als een coach meefietst langs de kant van het water kan hij niet bepalen hoeveel meter reeds is afgelegd sinds het begin van de wedstrijd en hoeveel meter het nog tot de finish is. Dit zal dus een kwestie van gokken zijn. Echter wellicht is dit in combinatie met een fietsenteller op te lossen. Daarnaast zouden er kilometerpaaltjes langs het roeiwater kunnen worden gezet. Het is echter bij tijdmeting door middel van een stopwatch onmogelijk om de opbouw van een roeitraining- of wedstrijd te meten en weer te geven. Aangezien dit het belangrijkste doel is van dit project is dit alternatief niet interessant om uit te voeren.

Mobiele telefoon

In de opdracht wordt beschreven dat er gebruik zal worden gemaakt van een PDA om het programma op te draaien. Gezien het feit dat een PDA redelijk prijzig is (vanaf € 210, zie bijlage 6) wordt er toch een drempel neergelegd voor het gebruik van dit programma. Zeker aangezien de gebruikers van het programma studenten zijn. Dit is enigszins in strijd in met de eis vanuit de opdrachtgever om gebruik van het programma zo goedkoop mogelijk te houden. Daardoor is gekeken naar goedkopere alternatieven waarbij werd uitgekomen op (smart)phone(s). Bij vervanging van de PDA door een mobiele telefoon welke beschikt over de juiste vereisten als Java en Bluetooth kan de kostendrempel aanzienlijk worden verlaagd. Dit omdat deze mobiele telefoons vanaf 0 euro te verkrijgen zijn in combinatie met een telefoonabonnement. Aangezien Java een platformonafhankelijke programmeertaal is zal er qua functionaliteit geen verschil optreden tussen de verschillende platformen (GSM dan wel PDA). Enige nadeel van het gebruik van dit alternatief is dat de schermjes kleiner zijn in vergelijking met de schermen op een PDA. Hierdoor is gekozen om voor dit potentiële alternatief verder te onderzoeken bij de keuze van het platform zoals gedocumenteerd in hoofdstuk 6.4.

Speedcoach

Als derde alternatief is de Speedcoach⁶ overwogen. Deze wordt gemaakt door Concept 2, een producent van roeibenodigheden. De Speedcoach heeft dezelfde tempofuncties als de StrokeCoach (om tempo te meten) maar heeft ook een extra sensor om de snelheid te

⁶ <http://www.concept2.nl/shop/product.php?open=12>

registreren. Hiervoor moet een 'low drag' impellertje (een soort propeller) op de huid geplakt worden, dat zonder sporen verwijderd kan worden.

De SpeedCoach Gold (€ 339) heeft uitgebreide programmeerfuncties en een geheugen dat de resultaten kan opslaan. Ideaal voor roeiers die alles van hun trainingen willen bijhouden. Computerinterface is binnenkort beschikbaar. Nadeel van de Speedcoach is echter dat de gemeten data pas achteraf kunnen worden weergegeven, na het aansluiten van de Speedcoach op een computer. Een eis van de opdrachtgever tijdens dit project was echter dat de gemeten data direct na de training weergegeven kunnen worden. Hierdoor kan namelijk de training direct met de roeiers worden geanalyseerd in plaats van dat dit pas ongeveer een week na dato kan geschieden. Hierdoor is het niet ondenkbaar dat de meeste roeiers vergeten zijn hoe een training is gegaan. De bedrijfsmentor kwam met een tweede nadeel namelijk de onnauwkeurigheid van de 'low drag' impeller op de huid van de boot. Hij kon uit ervaring vertellen dat de impeller (zeer) inconsistent is bij het berekenen van de snelheid met name bij woelig water. Deze conclusie werd van de bedrijfsmentor aangenomen en niet verder onderzocht aangezien dit alternatief reeds ongeschikt was gebleken als potentieel alternatief vanwege het eerste nadeel.

6.3 Keuze programmeertaal

Bij de keuze voor de programmeertaal is gekeken naar het aantal platforms waarop de verschillende programmeertalen te gebruiken zijn. Hiervoor is hieronder een overzicht opgenomen waarin weer wordt gegeven welke programmeertalen de verschillende platformen ondersteunen.

Ondersteunt ?	Pocket PC	Palm	Symbian	Smartphone	Phone
(micro) Java	JA (*)	JA	JA	JA	Sommige
C++	JA	JA	JA	??	NEE
(embedded) Visual Basic (***)	JA	JA	JA(**)	JA	??
Symbian	NEE	NEE	JA	JA	Sommige

(*) Java Virtual Machine (JVM) voor de Pocket PC is niet standaard geïnstalleerd, deze is echter gratis te verkrijgen/downloaden.⁷ De JVM's konden niet worden getest aangezien er geen Pocket PC voorhanden is.

(**) mits gebruik gemaakt wordt van Appforge⁸

(***) Visual Studio vereist

Eén van de randvoorwaarden die tijdens het opstellen van het plan van aanpak naar voren is gekomen, is dat bij de keuze van de architectuur (oa. programmeertaal) rekening dient te worden gehouden met het breed kunnen inzetten van het programma. Hierbij komen we via bovenstaande tabel uit bij Java als programmeertaal. Met deze programmeertaal is het mogelijk om het programma op alle platforms te draaien (incl. de mobiele telefoon). Aangezien de mobiele telefoon als potentieel alternatief wordt besproken (zie hoofdstuk 6.2) zou Symbian ook in aanmerking komen, ware het niet dat het niet gedraaid kan worden op PDA's

⁷ Voor meer informatie zie de volgende website :

<http://www.comp.lancs.ac.uk/computing/users/fitton/ppcJava.html> of zie bijlage 1

⁸ (US \$899 d.d. jan. '04 zie : http://www.theregister.co.uk/2003/09/24/palm_symbian_and_windows_apps/)

6.4 Keuze platform

Om te weten via welke communicatiemiddelen een GPS-ontvanger met een platform kan communiceren zal in deze paragraaf een overzicht worden gegeven van de verschillende platformen.

Daarna zal een keuze worden gemaakt voor welk platform het programma ontwikkeld zal worden. Eén van de randvoorwaarden die tijdens het opstellen van het plan van aanpak naar voren is gekomen, is dat bij de keuze van de architectuur (o.a. platform) rekening dient te worden gehouden met het breed kunnen inzetten van het programma.

In deze paragraaf zal duidelijk worden gemaakt wat de verschillende prijsvariaties zijn voor de platformen en over welke communicatiemiddelen ze beschikken om te kunnen communiceren met de verschillende soorten GPS-ontvangers (Compact Flash/ Secure Digital/ Bluetooth/ serieel / usb). Met platformen worden zowel de PDA's (Pocket PC/ Palm/ Symbian) als de (smart)phones bedoeld.

Als eerste zal gekeken worden naar de PDA's. Voor een overzicht van de PDA's is een test van de Consumentenbond d.d. september 2004 als uitgangssituatie genomen om een beeld te schetsen van de keuzemogelijkheden om het gebied van PDA's. Dit overzicht is opgenomen in bijlage 6. Uit het overzicht blijkt dat:

- 5 van 16 geteste PDA's beschikken over een compact flash uitbreidingsmogelijkheid, waarbij de goedkoopste de ASUS MyPal A620BT is met een prijs van € 270;
- 9 van de 16 PDA's beschikken over een secure digital uitbreidingsmogelijkheid, waarbij de goedkoopste de PALM Tungsten T2 is (destijds de verkrijgen voor € 210);
- 5 van de 16 PDA's via een kabel kunnen communiceren met een GPS-ontvanger waarbij de goedkoopste de Sony Clie PEG-TJ37 is met een prijs van € 320;
- 5 van de 16 PDA's beschikken over Bluetooth waarbij wederom de ASUS MyPal A620BT de goedkoopste is met een prijs van € 270.

Voor een overzicht van de GSM's als platform is er een overzicht gemaakt van de mobiele telefoons die beschikken over Bluetooth en MIDP 2.0 (Java) daar alleen telefoons die deze twee functies ondersteunen kunnen communiceren met een GPS-ontvanger.

Hieruit blijkt dat er vele telefoons beschikbaar zijn op de markt die vanaf € 0 te koop zijn indien ze in combinatie met een mobiele telefoonabonnement worden aangeschaft bij een telecomprovider. Het overzicht is opgenomen in bijlage 3.

Vanwege de lage prijs zijn met name de GSM's als platform zeer interessant. Deze telefoons zijn namelijk in combinatie met een telefoonabonnement te koop vanaf 0 euro. Dit is met name interessant aangezien studenten (de doelgroep) gebruik maken van deze aanbiedingen. De meeste studenten willen toch een mobiele telefoon aanschaffen. Hierdoor hoeft je niet speciaal hiervoor een veel duurdere (minstens € 210) pocket pc of palm te kopen.

Daarnaast zullen bij gebruik van een GSM als platform de functionaliteiten van het programma bij het gebruik van Java als programmeertaal exact hetzelfde zijn als bij gebruik van een PDA als platform. Dit omdat Java een platformonafhankelijke programmeertaal is. Enige verschil hierin zal het formaat van het beeldscherm zijn. Een beeldscherm van een GSM is doorgaans kleiner dan een scherm van een PDA. Hiermee dient dan rekening gehouden te worden bij het beeldschermontwerp zoals besproken in hoofdstuk 7.2.

Bij gebruik van een mobiele telefoon als platform dient deze echter wel te beschikken over de Bluetooth API van Java (zie hoofdstuk 6.8 voor uitleg). Daarnaast stellen sommige telefoons een maximum aan de geheugenruimte die een applicatie mag gebruiken (zie voor een overzicht van de telefoons bijlage 1).

De maximale geheugenruimte staat vermeld in de kolom “max. JAR size”. Van de telefoons waarvan de maximum geheugenruimte bekend is heeft de Nokia 6230 de minste geheugenruimte (128 kb) beschikbaar. Hier zal dan ook zo veel mogelijk rekening gehouden te worden bij de ontwikkeling van het programma.

Concluderend zal in plaats van een PDA als platform een GSM gebruikt worden. Dit vanwege de veel gunstigere prijs van een GSM terwijl er qua functionaliteiten geen verschil is vanwege het platformonafhankelijk Java. Er dient echter bij het beeldschermontwerp wel rekening gehouden te worden met het feit dat een GSM beschikt over een kleiner beeldscherm.

Wijzigingen werkzaamheden wegens keuze ander platform

In het begin van het project is bij het vaststellen van de opdrachtschrijving uitgegaan van een PDA als platform. Nu gekozen is voor een GSM als platform in plaats van een PDA zullen de werkzaamheden zoals in eerste instantie vastgelegd in de opdrachtschrijving veranderen.

Het zal dan met name gaan om de punten 7 en 8 zoals vastgesteld in hoofdstuk 4.6. Voor de duidelijk staan ze hieronder nogmaals.

1. Bepalen hoe gemeten data moet worden opgeslagen in het geheugen van de PDA. N.B. Er zit een automatisch log-functie op de meeste PDA's. Nadeel hiervan is echter dat de data naar de seriële poort wordt verstuurd.
2. Communicatie tussen PDA en een (programma op de) pc voor het uitlezen en archiveren van gemeten data

De werkzaamheden zijn in zoverre veranderd dat er gekeken zal moeten worden naar hoe de gemeten moet worden opgeslagen in het geheugen van een PDA. Daarnaast zal gekeken moeten worden hoe de communicatie tussen een GSM en een (programma op de) pc voor het uitlezen en archiveren van gemeten data.

6.5 Keuze GPS-ontvanger

In deze paragraaf zal de keuze voor de GPS-ontvanger worden toegelicht. Hierbij zal DGPS als mogelijke GPS-ontvanger niet worden meegenomen vanwege twee redenen welke zijn toegelicht in het kopje (on)nauwkeurigheid verder in deze paragraaf.

Eén van de randvoorwaarden die tijdens het opstellen van het plan van aanpak naar voren zijn gekomen is dat bij de keuze van de architectuur (o.a. platform) rekening dient worden gehouden met het breed kunnen inzetten van het programma. Hierdoor zullen per type verbinding de goedkoopste versies hieronder worden toegelicht aangezien de prijs zo laag mogelijk dient te worden gehouden. Dit was een wens van de opdrachtgever, zoals te lezen is in hoofdstuk 5.5.

De mogelijke verbindingen zijn: Bluetooth, Secure Card (SD), Compact Flash (CF), RS232 (serieel) of USB.

Naam	Prijs	Nauwkeurigheds systeem	type verbinding	ondersteund(e) protocol(len)
HOLUX GR-230	€ 129	Sirf II	Bluetooth	NMEA 0183 v2.20 en SiRF Binary
HOLUX GR-231	€ 169	SiRF XTrac v2.02	Bluetooth	NMEA 0183 v2.20 en SiRF Binary
Rikaline GPS Muis 6010	€ 99	Sirf II	Serieel	NMEA 0183 v2.20 en SiRF binary protocol
Haicom GPS Muis Hi204s	€ 109	Sirf II	Serieel	NMEA 0183
Rikaline 6030 Bluetooth GPS	€ 119	Sirf II/LP	Bluetooth	NMEA-0183
Garmin Foretrex 101	€ 139	Sirf II	RS232	RS232 with NMEA 0183 and proprietary Garmin
GlobalSat SD-501 (SDIO) GPS	€ 125	Sirf XTrac	Secure Digital (SD)	NMEA 0183 (Secondary: SiRF binary)
GlobalSat BC307 (CF) GPS	€ 99	Sirf II/LP	Compact Flash (CF)	NMEA0183 protocol

(On)nauwkeurigheid

Hoewel DGPS (iets) nauwkeuriger is dat de softwarematige nauwkeurigheidssystemen als EGNOS, WAAS en XTrac (zie hoofdstuk 2) ligt de prijs van een DPGS beacon ontvanger veel hoger.

Daarnaast is uitgaande van de nauwkeurighedsgegevens van producenten van de softwarematige nauwkeurigheidssystemen gekeken of de 99% eis aan nauwkeurigheid van de opdrachtgever kan worden gehaald (zie hoofdstuk 5.3). Bij een eerste verkenning van deze eis in de definitie-studie kan geconcludeerd worden dat de maximale onnauwkeurigheid van het programma 20 meter mag zijn (1% onnauwkeurigheid bij 2000 meter).

Uitgaande van de producentgegevens zal enkel het softwarematige XTrac boven de 20 meter nauwkeurigheid uitkomen. EGNOS en WAAS zullen volgens de productgegevens een nauwkeurigheid hebben van minder dan 7 meter.

Hierbij dient, zoals eerder verteld in hoofdstuk 2.1, bij te worden verteld dat XTrac deze lagere nauwkeurigheid van 20 meter alleen zal hebben bij slecht zicht in de open lucht.

Daar waar een GPS-ontvanger met EGNOS of WAAS helemaal geen positie kan bepalen.

Indien bij de eerste iteratie mocht blijken dat de eis van de opdrachtgever aangaande de 99% nauwkeurigheid niet gehaald kan worden met WAAS, EGNOS of XTrac (afhankelijk van welke in deze paragraaf zal worden gekozen) zal toch gekeken moeten worden naar DGPS indien de 99% eis niet veranderd zal worden. Het gebruik van DGPS zal dus niet verder onderzocht worden tenzij blijkt dat de softwarematige nauwkeurigheidssystemen niet voldoen aan de 99% eis van de opdrachtgever.

Hoewel de GPS-ontvanger met XTrac duurder is dan GPS-ontvangers met EGNOS (Sirf II) is het toch het overwegen waard om een GPS-ontvanger met XTrac te kopen aangezien de betere nauwkeurigheid van deze ontvangers.

Dit met name op gebieden met minder zicht op de open lucht. Hoewel het programma gebruikt zal worden op open roeiwater met zicht op de open lucht is met XTrac eerder te voorkomen dat er geen positie kan worden bepaald indien er onder een brug wordt doorgevaren of dat de GPS-ontvanger bij iemand in de jaszak zit.

De typische GPS positie nauwkeurigheid zonder SA is 15 meter. Typische differentie GPS (DGPS) positie nauwkeurigheid ligt tussen de 3 en 5 meter. De typische EGNOS positie nauwkeurigheid is ongeveer 7 meter. Hierbij zijn de producentgegevens gebruikt als bron. De nauwkeurigheid van XTrac ligt tussen de 5 tot 25 meter waarbij de nauwkeurigheid van 25 meter gehaald zal worden op plaatsen met weinig zicht op de open lucht (bijvoorbeeld in een broekzak). Daar waar de GPS-ontvangers met WAAS of EGNOS helemaal geen positie kunnen bepalen.

Daarnaast is het Europese EGNOS systeem op dit moment nog niet operationeel (zie hoofdstuk 2). Hierdoor zal een GPS-ontvanger met EGNOS waarschijnlijk de typisch GPS positie nauwkeurigheid zonder SA behalen (15 meter).

Gebruik Bluetooth

Telefoons waarop Java programma's draaien kunnen enkel communiceren met GPS-ontvangers via Bluetooth. Hierdoor zal de Pelargos Training Coach Bluetooth moeten gaan ondersteunen en zal gebruik moeten worden van een Bluetooth GPS-ontvanger.

Het programma kan dan echter ook nog gebruikt worden op verschillende Pocket PC's en Palm's (PDA's). Uit de test is namelijk gebleken dat 5 van de 16 geteste PDA's beschikken over Bluetooth.

6.6 Keuze GPS protocol

Gezien alle GPS-ontvanger die onderzocht zijn het NMEA 0.183 protocol ondersteunen wordt het programma alleen met dit protocol ontworpen te worden. Dit ook omdat sommige Bluetooth GPS-ontvanger alleen het NMEA 0.183 protocol ondersteunen (zie overzicht in de vorige paragraaf).

6.7 Keuze softwareprogramma

Na de keuze van de programmeertaal voor het ontwikkelen moet worden bepaald in welk ontwikkelprogramma het programma gemaakt zou worden. Gekozen is voor de programmeertaal Java, andere eis is dat het programma gratis te verkrijgen is aangezien er vanuit de vereniging geen geld beschikbaar is en er ook geen programma reeds in het bezit is van de afstudeerder dan wel van de vereniging. Derde eis is dat het pakket de beschikking moet hebben over de Bluetooth API package zoals vastgesteld in hoofdstuk 6.4 vanwege het gebruik van een GSM als platform (zie hoofdstuk 6.8). Indien het pakket namelijk niet zou beschikken over deze package dan zou het onmogelijk zijn om het programma te compileren.

Op internet ben ik op zoek gegaan naar gratis Java IDE's (Integrated Development Environment, ontwikkelomgeving) beginnende bij <http://java.pagina.nl>. Hierop was een overzicht van IDE's te vinden verdeeld in twee groepen; commercieel en niet-commercieel.

De volgende gratis IDE's waren er opgesomd:

1. Eclipse Eclipse + the ME plugin (downloadable from Sourceforge.net).
2. NetBeans + mobility pack.
3. jEdit

Alle drie de IDE's bleken te beschikken over het vereiste package, dan wel dat deze via een (gratis) plugin te installeren was. Aangezien er verder geen duidelijke verschillen waren is besloten om gebruik te maken van NetBeans. NetBeans is namelijk het eigendom van Sun Microsystems, en zal als zodanig beschikken over de meeste kennis van Java.














6.8 Gebruik Bluetooth API

Om met micro Java een connectie met een Bluetooth GPS-ontvanger te maken en er de data van te ontvangen is de Bluetooth API ontworpen door Sun. Aangezien het zelf implementeren van een soortgelijke code die deze API kan vervangen zeer tijdrovend is, is het handig en efficiënt om gebruik te maken van de Bluetooth API. Gebruik van de Bluetooth API stelt echter wel als eis dat hij ondersteund dient te worden op de telefoon waarop de applicatie gedraaid zal worden. Er is echter gebleken dat in ieder geval de grote telecommataatschappijen het gebruik van deze Bluetooth API (JSR #82) hebben goedgekeurd en ondersteunen.


Een overzicht van mobiele telefoons die zowel Java (MIDP 2.0) als (de) Bluetooth (API) ondersteunen is te vinden in bijlage 1. Gezien de tijdsbesparing qua programmeren die gebruik van de Bluetooth API oplevert, alsmede de brede ondersteuning vanuit telecom-producten is besloten om gebruik te maken van de Bluetooth API.


Deze brede ondersteuning van de Bluetooth API door de telecomproducten blijkt uit het stem-overzicht waarmee het gebruik van de Bluetooth API is ingestemd door de verschillende producten. Dit overzicht is hieronder weergegeven⁹.

Final results of the Community Draft Ballot for JSR #82

BEA Systems <input type="checkbox"/>	Cisco Systems 	Ericsson Inc. 	IBM 
Insignia Solutions <input type="checkbox"/>	Matsushita Electric Industrial Co., Ltd. <input type="checkbox"/>	Motorola 	Nokia Mobile Phones 
PalmSource, Inc. 	Philips Electronics UK Ltd 	Research In Motion, LTD (RIM) 	SONY International (Europe) GmbH 
Siemens AG 	Sun Microsystems, Inc. 	WindRiver Systems 	Zucotto Wireless 

Legenda

Yes  Er is ingestemd met gebruik van de Bluetooth API

No  Er is tegengestemd voor het gebruik van de Bluetooth API

Not voted ☐ Er is door de betreffende product niet gestemd

⁹ Bron : www.jcp.org/en/jsr/detail?id=82

Werking Bluetooth API

Om via Bluetooth met een ander apparaat te kunnen communiceren dient eerst te worden gescand naar beschikbare Bluetooth apparaten alvorens de manier van communiceren kan worden bepaald. Dit is afhankelijk van de services die beschikbaar zijn op het gevonden Bluetooth apparaat. Naar deze beschikbare services dient tevens te worden gevraagd: dit geschiedt nadat er een Bluetooth apparaat gevonden is.

Gebruik van de Bluetooth API biedt een raamwerk voor het vinden van devices die in het bereik zijn van de Java telefoon en tevens de services op die devices.

Bij gebruik van de Bluetooth API worden de volgende stappen doorlopen.

1. zoek devices door middel van de discoveryagent, indien een device gevonden is wordt de device discovered functie aan geroepen waarbij de informatie over het device als parameter wordt meegegeven. Indien het zoeken naar Bluetooth devices voltooid is, wordt de functie inquiryCompleted aangeroepen.
2. zoek services die op de devices draaien ook via de discoveryagent, indien een service gevonden is wordt de servicediscovered functie aan geroepen waarbij de informatie over de service als parameter wordt meegegeven. Indien het zoeken naar Bluetooth services voltooid is wordt de functie serviceSearchCompleted aangeroepen.
3. open connectie & inputstream. Open met de verkregen informatie een Stream-connection en een inputstream met het Bluetooth apparaat. Hierdoor is het mogelijk om de GPS data te ontvangen en uit te lezen.

6.9 BCC (Bluetooth Control Center) uitgelegd

In het testprogramma (zie hoofdstuk 8.3), werd in eerste instantie bij elke keer dat het programma werd opgestart gezocht naar Bluetooth apparaten en de services die erop draaiden. Aangezien het tijdrovend is om elke keer als het programma wordt opgestart opnieuw de Java telefoon te laten scannen naar alle aanwezige Bluetooth apparaten, is gekeken naar de mogelijkheden om gebruik te maken van het Bluetooth Control Center (BCC).

Dit verkort met name de zoektijd naar een Bluetooth apparaat omdat het programma niet elke keer bij het opstarten moet gaan scannen naar alle beschikbare Bluetooth apparaten, maar het gewoon een overzicht kan laten zien van alle bekende ("pre known") Bluetooth apparaten.

Een gebruiker kan dan makkelijk de door hem gewenste Bluetooth apparaat selecteren.

Een Bluetooth apparaat wordt pre known door hem eenmalig te autoriseren in het BCC van een mobiele telefoon door het opgeven van een passcode. Hierdoor worden de instellingen van de GPS-ontvanger opgeslagen in het BCC van de mobiele telefoon. Hierdoor kunnen deze instellingen van een Bluetooth apparaat gemakkelijk worden opgevraagd door onder andere een Java programma. Naar deze informatie hoeft dan niet iedere keer bij het opstarten van het programma gezocht te worden door middel van scannen (devicediscovery) naar Bluetooth apparaten. Het opvragen van deze preknown apparaten kan via de volgende broncode. Alle bekende (pre known) devices in het BCC worden hierdoor opgeslagen in de devices array.

```
// retrieve the discovery agent
agent = local.getDiscoveryAgent();
// return an array of pre-known devices
devices = agent.retrieveDevices(DiscoveryAgent.PREKNOWN);
```

HOOFDSTUK 7. ARCHITECTUURFASE

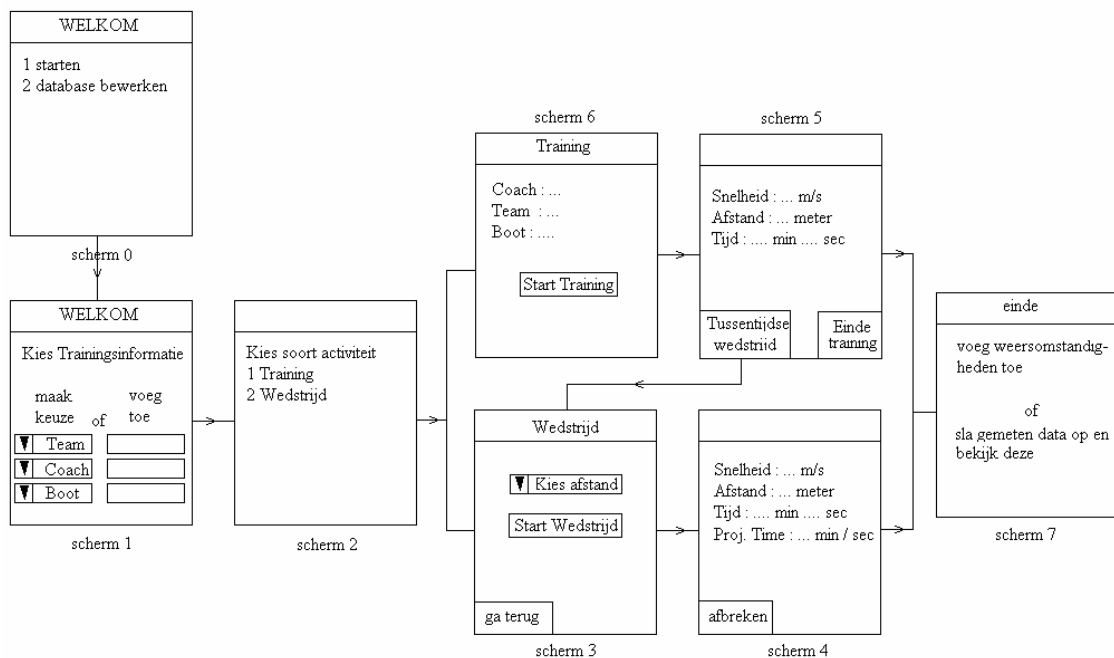
7.1 Inleiding

Na het definiëren van de opdracht in de definitiestudie zal, volgens de gebruikte IAD methode, begonnen worden met het ontwikkelen van het programma, beginnende bij het bepalen van de architectuur. Hierbij valt te denken aan het beeldschermontwerp, het klassediagram, het databaseontwerp, de opslag van GPS data en exporteren data naar personal computer. Dit zal in dit hoofdstuk worden uitgelegd.

7.2 Beeldschermontwerp

Aan de hand van de eisen zoals vermeld in de opdrachtoomschrijving en het plan van aanpak is er eerst een schets gemaakt van de GUI. Wat deze eisen en wensen waren en hoe deze zijn verwerkt in het beeldschermontwerp is hieronder opgesomd waarbij onderstaand plaatje is gebruikt ter illustratie van de tekst:

- Tijdens het meten dient er al een real-time weergave afgebeeld te worden van het verloop van de training of wedstrijd.
 - Verwerkt in scherm 4 en 5 (zie onderstaand plaatje) waarbij de huidige snelheid, afgelegde afstand, verstreken tijd en vermoedelijke eindtijd (enkel bij wedstrijd in scherm 4) wordt weergegeven.
- De verkregen input die tijdens de training is verzameld dient geëxporteerd te kunnen worden naar een PC.
 - Verwerkt in scherm 7 en wordt gedaan terwijl programma wordt afgesloten.
- De verkregen input dient aangevuld te kunnen worden met informatie over: de weersomstandigheden, instruerende coach, teamnaam en het type boot waarin is geroeid.
 - Verwerkt in scherm 1 waar van tevoren de keuze kan worden gemaakt voor de coach, het team en de boot. Op het einde kunnen de weersomstandigheden worden ingevoerd. Dit aangezien je pas op het einde van de training weet hoe de omstandigheden zijn geweest.
- Het moet mogelijk zijn om tijdens een training een wedstrijdstand af te leggen.
 - Verwerkt in scherm 5 waarbij een tussentijdse wedstrijd kan worden gekozen.



* Beeldschermontwerp

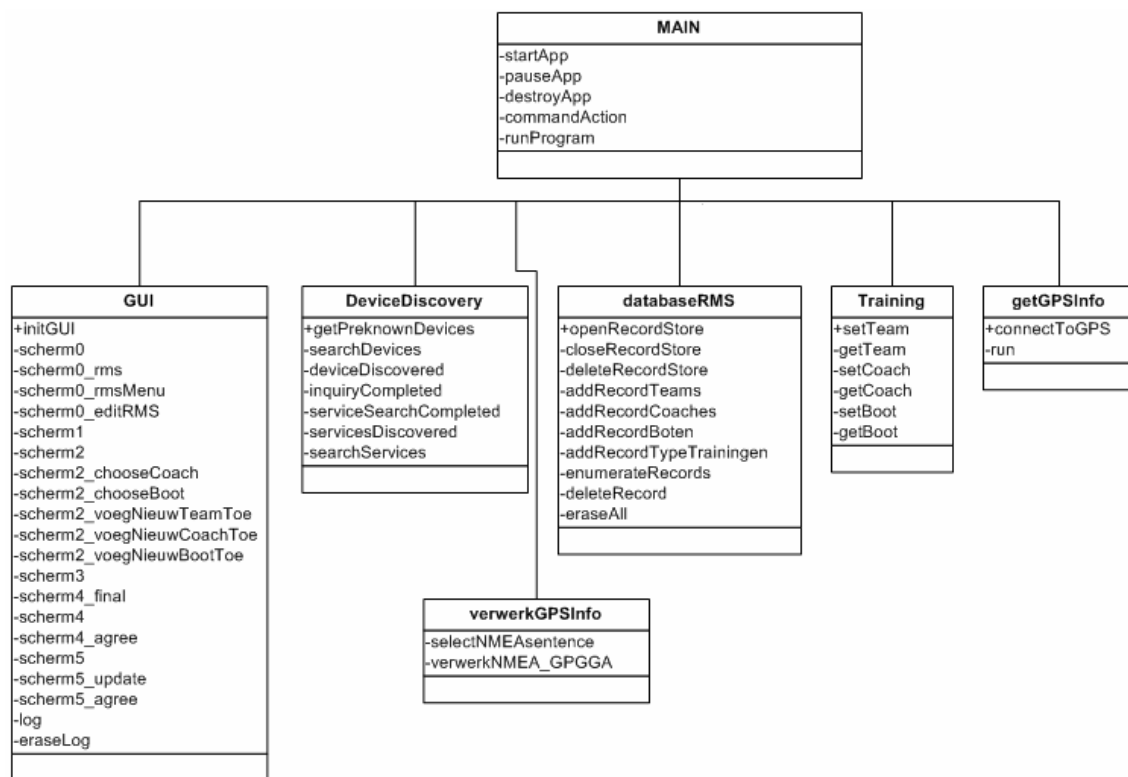
De laatste eis was dat het beeldschermontwerp zo eenvoudig mogelijk te bedienen moest zijn. Hiervoor is een soort “wizard” manier geïmplementeerd waarbij de keuzes die mogelijk zijn telkens aan de gebruiker worden weergegeven. Dit bijvoorbeeld bij scherm 5 waarin direct te mogelijkheid wordt gegeven om een tussentijdse wedstrijd in te plannen. Hierdoor hoeft een gebruiker niet te zoeken naar de mogelijkheid van een tussentijdse wedstrijd door terug te klikken naar scherm 2 en vervolgens een wedstrijd te beginnen.

Deze schets is tijdens een vergadering met de bedrijfsmentor besproken en goedgekeurd en is daarom verder uitgewerkt. Dit gebeurde door het implementeren van deze GUI in een testprogramma. Dit zal besproken worden in hoofdstuk 8.2.

7.3 Klassediagram

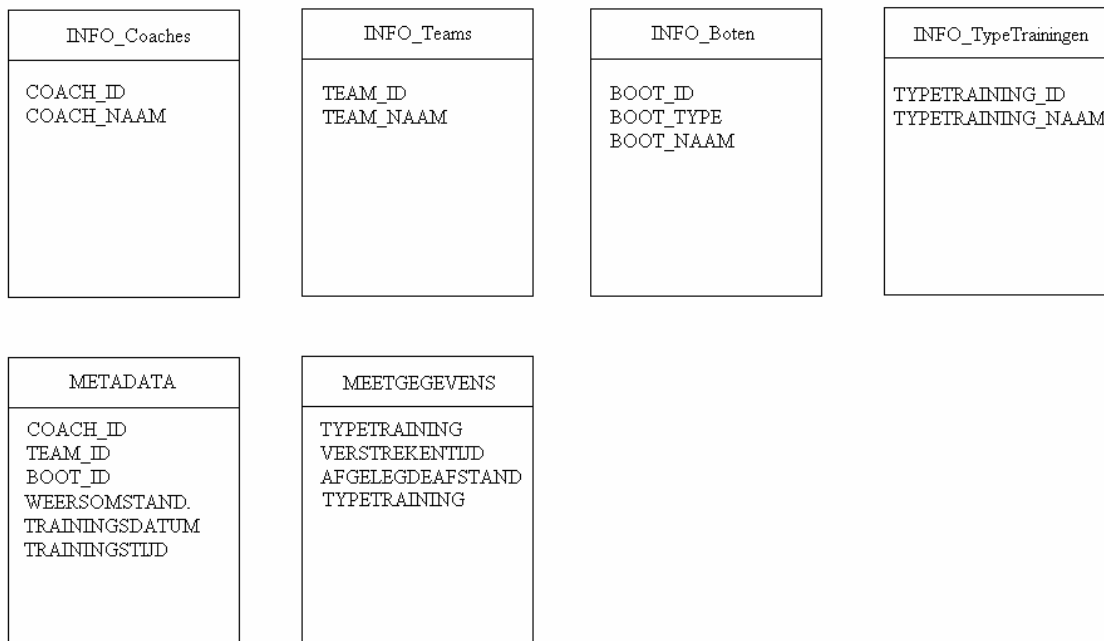
Vervolgens is aan de hand van de GUI een klassediagram bepaald. De klasse “MAIN” initialiseert alle andere klassen en start en stopt de thread in de klasse “getGPSInfo”. Deze thread haalt elke seconde de GPS data op uit de GPS-ontvanger en verwerkt deze in de klasse “verwerkGPSInfo”. Er is gekozen om de GPS data 1 keer per seconde op te halen aangezien de GPS-ontvanger deze data 1 keer per seconde verstuurd.

Alle schermen en alles wat daar mee te maken heeft zijn ondergebracht in de klasse “GUI”. Het contact leggen met de Bluetooth apparaten gebeurt in de klasse “DeviceDiscovery”, het bewerken en beheren van de database gebeurt via de klasse “databaseRMS” en in de klasse “Training” wordt de informatie van de training beheert. Gekozen voor deze scheiding van klassen is vanwege de drie testprogramma die zullen worden gemaakt om meer duidelijkheid te krijgen over de werking van verschillende componenten (Bluetooth, RMS en GUI) van de opdracht. Meer informatie hierover in hoofdstuk 8.1.



* klassediagram

7.4 Databaseontwerp



Er zijn geen onderlinge relaties tussen de entiteiten. Bij het ontwerpen van het databaseontwerp is onderscheid gemaakt tussen twee type data. Ten eerste is er de data die constant bewaard moet blijven zoals de namen van de coaches, teams en de boten. Hieruit dient elke keer bij het starten van het programma een keuze te worden door de gebruiker (zie beeldschermontwerp). Hiermee is voldaan aan de eis van de opdrachtgever om deze data aan een training toe te kunnen voegen (zie hoofdstuk 5.3). Deze gegevens worden opgeslagen in de bovenste vier databases. Gekozen is om voor iedere type informatie een aparte database aan te maken om de verschillende soorten informatie beter te kunnen beheren. Dit bijvoorbeeld voor het toevoegen van een coach of het wissen van alle teams bij het bewerken van de databases door de gebruiker (scherm 0 in het beeldschermontwerp; hoofdstuk 7.2).

Ten tweede is er specifieke data over een training anders is zoals afgelegde afstanden, tijd en datum. Deze data staan vermeld in de onderste twee databases (recordstores) en worden per training aangemaakt. Dit omdat er tijdelijke en specifieke informatie over de training in is opgeslagen. In de linker recordstore “METADATA” staat alle informatie over de training in opgeslagen zoals de coach, team en datum van de training. In de recordstore “MEETGEGEVENS” staan de GPS gegevens en informatie die nodig is voor het genereren van het overzicht voor de opbouw van de training.

Bij het ontwerpen van de databases is specifiek gekeken naar de mogelijkheid om bij een training alle type trainingen apart weer te kunnen geven. Aan deze eis vanuit de opdrachtgever diende namelijk voldaan te worden zoals vastgesteld in hoofdstuk 5.3.

Bijvoorbeeld als er tijdens de training 6 opzetjes waren gedaan dan moest de opbouw van ieder opzetje duidelijk van elkaar te onderscheiden zijn. Hiervoor is de volgende opzet van de recordstore “MEETGEGEVENS” bedacht;

GPS-coördinaten	Tijd	TypeTraining	Afgelegde afstand
X1, Y1	T1	1	2
X2, Y2	T2	1	4
X3, Y3	T3	1	6
X4, Y4	T4	2.1	8
X5, Y5	T5	2.2	12
X6, Y6	T6	2.3	15
X7, Y7	T7	2.4	17
X8, Y8	T8	3	19

De kolommen “GPS-coördinaten”, “Afgelegde afstand” en “Tijd” spreken voor zichzelf het gaat met name om de kolom “TypeTrainingen”. Hierin is per type training een aparte code in opgenomen om het onderscheiden van ervan mogelijk te maken. Bijvoorbeeld 2.1 is dan opzetje 1 en 2.2 opzetje 2 enz. Vervolgens is 3 dan het stuk training na de opzetjes. Vanwege een aparte code per type training is het mogelijk om de verschillende trainingen van elkaar te scheiden.

Aangezien aan het eind van de training alle in deze twee databases opgeslagen data worden geconverteerd naar een bestand kunnen deze databases telkens geleegd aan het eind van de training waardoor ze telkens weer opnieuw kunnen worden gebruikt. De data worden aan het einde van de training geconverteerd naar een bestand om de verkregen trainingsinformatie te kunnen exporteren naar een personal computer dan wel een PDA wat een eis was vanuit de opdrachtgever. Het databasebestand kan niet direct worden geëxporteerd naar een extern apparaat aangezien op een mobiele telefoon enkel micro Java wordt ondersteund welke niet gebruik maakt van gangbare databases als Access, maar gebruik maakt van recordstores vanwege de beperkte capaciteiten van een mobiele telefoon.

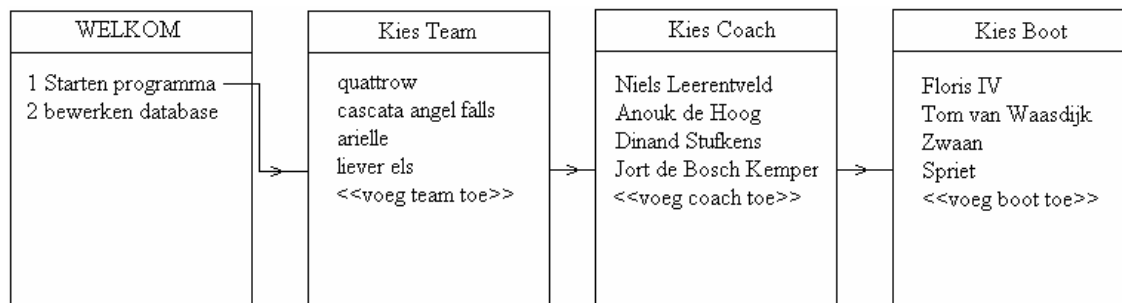
HOOFDSTUK 8. ONTWIKKELFASE

8.1 Inleiding

Na bepalen van de ontwerp en eisen aan het product begint het ontwikkelen van het programma. Het proces zoals verlopen is in de fase zal in dit hoofdstuk worden toegelicht. Aangezien de afstudeerder over weinig ervaring beschikte met het programmeren van een programma voor een mobiele telefoon zijn er voor de verschillende onderdelen van het programma separate testprogramma gemaakt. Deze hadden als doel om de werking van een bepaald onderdeel onder de knie te krijgen. Het ging dan om het programmeren van het beeldschermontwerp maar in meerdere mate voor het contact leggen met en uitlezen van een GPS-ontvanger via Bluetooth. Daarnaast is er een testprogramma gemaakt voor de opslag van de GPS-data via de recordstores. Het verloop van dit proces zal in dit hoofdstuk worden besproken.

8.2 Beeldschermontwerp

Tijdens het implementeren van de eerste schets, zoals besproken in hoofdstuk 7.2, van het testprogramma kwamen enkele praktische aandachtspunten naar voren welke echter eenvoudig konden worden aangepast. Het was namelijk onmogelijk om een dropdown menu in te voegen in het programma. Het ging dan met name op het welkom scherm (meest linkse). Er konden geen keuzemenu's worden weergegeven. Dit werd echter eenvoudig aangepast door het gebruik van drie verschillende schermen waarin dezelfde keuzes konden worden gemaakt door middel van een choice list met een impliciete keuze waardoor alleen één van de keuzemogelijkheden kan worden geselecteerd. Daarnaast ontstond hierdoor ook de mogelijkheid om gemakkelijk een keuze in te voegen waarmee een team, coach of boot toegevoegd kon worden.



Na deze aanpassing was de GUI goed in te voeren in het testprogramma en werkte zonder problemen. Ondanks deze aanpassing voldeed het ontwerp nog steeds aan de eisen zoals vastgesteld. Het testprogramma is op een mobiele telefoon ter controle aan de bedrijfsmentor voorgelegd. Hij had geen andere opmerking dan dat er bij het wissen van een rij uit een recordstore er om een bevestiging diende te worden gevraagd.

Als laatste is meegenomen hetgeen in hoofdstuk 6.4 staat vermeld. Hierin wordt besproken dat het gebruik van een GSM als platform in plaats van een PDA als enige nadeel met zich meebrengt dat een GSM vaak een kleiner scherm heeft dan een PDA. Hierdoor past er minder informatie op het scherm. Bij het testen van het beeldschermontwerp op de mobiele telefoon kwamen hierdoor echter geen problemen naar voren. Het enige nadeel was dat sommige namen te lang waren (bv "cascata angel falls") om in één keer in het beeldscherm te passen. Dit werd echter door Java zelf opgelost door gebruik van een lichtkrant. Hierbij wordt

continue de desbetreffende regel van voor naar achter telkens gedeeltelijk weergegeven. Hierdoor zie je niet te hele naam in één keer maar kun je deze geleidelijk aflezen.

8.3 Opslag GPS data

Voor de opslag van de GPS data kon geen gebruik gemaakt worden van een reguliere databases welke bij een programma op de computer gebruikt kunnen worden. Vanwege de beperkte processorkracht en geheugencapaciteit van een mobiele telefoon kunnen er geen databases op draaien als Access. Er dient echter op een mobiele telefoon gebruik gemaakt te worden van recordstore voor de gegevensopslag.

Aangezien ik onbekend was met de werking van deze recordstores is via Google gezocht naar meer informatie hierover. Via Google kwam ik uit op een website over Java ¹⁰ waarin het Record Management System (RMS) werd behandeld. In dit RMS worden de recordstores geregeld. Aan de hand van dit artikel is een testprogramma gemaakt waarin recordstore worden aangemaakt, worden uitgelezen en kunnen worden gewist. Daarnaast konden er records¹¹ worden toegevoegd en worden gewist.

Dit testprogramma is via de mobiele telefoon getest door het openen van een recordstore waarin vervolgens 10 records werden toegevoegd. Van deze records werd vervolgens een overzicht weergegeven. Deze records werden daarna getest of ze konden worden gewist en of de juiste record ook daadwerkelijk werd gewist.

8.4 Verbinding maken met Bluetooth apparaat

Vanwege gebruik Bluetooth API, zoals besproken in hoofdstuk 6.8, is er gezocht naar informatie hoe dit te implementeren. Hier zijn eerst de Javadocs gedownload, hierin staat per klasse in hoe de verschillende functies gebruikt kunnen worden.

Als eerste is gezocht op de website van Sun voor meer informatie over het implementeren van de Bluetooth API aangezien Sun de verstrekker is van de Bluetooth API. Op deze website is een artikel gevonden over dit onderwerp. Aan de hand van dit artikel is er een testprogramma gemaakt totaal losstaand van het Pelargos Training Coach programma. Dit testprogramma is stap voor stap opgebouwd en puur gericht op het contact maken met en het ontvangen van de data van de Bluetooth GPS-ontvanger. Eerst werd gezocht naar devices door middel van

```
// retrieve the local Bluetooth device object
local = LocalDevice.getLocalDevice();
local.setDiscoverable(DiscoveryAgent.GIAC);

// retrieve the discovery agent
agent = local.getDiscoveryAgent();
// place the device in inquiry mode
boolean complete = agent.startInquiry(DiscoveryAgent.GIAC, this);
// if device found, function deviceDiscovered is invoked
```

Zodra er een Bluetooth apparaat ontdekt is wordt door de DiscoveryListener de functie “devicediscovered” aangeroepen waarbij de informatie over het gevonden Bluetooth apparaat als parameter wordt meegegeven. Indien het scannen naar Bluetooth apparaten voltooid is, wordt door de DiscoveryListener de functie “discoverycompleted” aangeroepen; hierna kan dan naar de services gezocht worden op een Bluetooth apparaat.

¹⁰ <http://today.java.net/lpt/a/149>

¹¹ Records zijn de rijen in een recordstore

De gevonden informatie is telkens via een zelfgemaakte log-functie via een Textbox op het scherm van de Java telefoon getoond om te verifiëren of de ontdekking van het Bluetooth apparaat goed gelukt was. Telkens als bleek dat de code goed functioneerde is verder gegaan met de volgende stap.

Service discovery

Zoals aangegeven is er een testprogramma gemaakt om contact te maken met en het ontvangen van de data van de Bluetooth GPS-ontvanger. Voor het contact maken met de Bluetooth GPS-ontvanger diende eerst gezocht te worden naar devices alvorens per device gezocht kon worden naar de services. Hier kwamen echter de eerste problemen met het testprogramma om de hoek kijken, aangezien het artikel op de website van Sun alleen specifieke code voorbeelden geeft zoals getoond in onderstaande afbeelding.

Device Management

The Java Bluetooth APIs contain the classes `LocalDevice` and `RemoteDevice`, which provide the device-management in the Generic Access Profile. `LocalDevice` depends on the `javax.bluetooth.DeviceClass` class to retrieve the kinds of services it offers. The `RemoteDevice` class represents a remote device (a device within a range of reach) and retrieve information about the device, including its Bluetooth address and name. The following code snippet retrieves that device:

```
...
// retrieve the local Bluetooth device object
LocalDevice local = LocalDevice.getLocalDevice();
// retrieve the Bluetooth address of the local device
String address = local.getBluetoothAddress();
// retrieve the name of the local Bluetooth device
String name = local.getFriendlyName();
...
```

You can get the same information about a remote device:

```
...
// retrieve the device that is at the other end of
// the Bluetooth Serial Port Profile connection,
// L2CAP connection, or OBEX over RFCOMM connection
RemoteDevice remote =
    RemoteDevice.getRemoteDevice(
        javax.microedition.io.Connection c);
// retrieve the Bluetooth address of the remote device
String remoteAddress = remote.getBluetoothAddress();
```

Vanwege deze specifieke code voorbeelden is het onduidelijk hoe de verschillende functies met elkaar interacteren. Hierdoor werd ik met name door de volgende zin in het artikel “Once the local device has discovered at least one remote device, it can begin to search for available *services*” op het verkeerde spoor gezet. Deze zin werd namelijk door mij geïnterpreteerd als zodra er een remote device is gevonden er direct gezocht kon worden de services die op het desbetreffende apparaat beschikbaar zijn. Mijn code zag er daardoor als volgt uit. Om te beginnen werd er naar Bluetooth apparaten gezocht door middel van de

```
// place the device in inquiry mode
boolean complete = agent.startInquiry(DiscoveryAgent.GIAC, this);
```

Indien er een Bluetooth apparaat gevonden is, wordt door de discoverylistener de functie `devicediscovered()` aangeroepen waarin als parameter de informatie over het Bluetooth

apparaat wordt meegegeven. Door de eerder genoemde zin uit het artikel op de Sun website zag mijn `deviceDiscovered` functie er als volgt uit :

```
public void deviceDiscovered(RemoteDevice remoteDevice, DeviceClass
deviceClass)
{
    // search service of found remote device
    try {
        remote = remoteDevice;
        agent.searchServices(null,
            new UUID[]{new UUID(0x0003)},
            remote, this);
    }
}
```

Dit bleek echter niet te werken. Er werden door deze functie geen services ontdekt.

Oplossing voor vinden services

Via de code die was geschreven aan de hand van het vermelde artikel op de website van Sun konden geen services worden ontdekt. Aangezien ik door dit artikel er vanuit ging dat ik naar services moest zoeken in de `deviceDiscovered` functie zocht ik de fout bij de functie die naar services zoekt (deze code is hierboven ook vermeld) :

```
remote = remoteDevice;
agent.searchServices(null,
    new UUID[]{new UUID(0x0003)},
    remote, this);
```

Hier van uitgaande ging mijn aandacht specifiek naar de parameter “`new UUID[]{new UUID(0x0003)}`” in deze functie. In deze parameter wordt meegegeven “the set of UUIDs that are being searched for; all services returned will contain all the UUIDs specified here”. Een UUID is een universeel unieke identificatiecode voor de verschillende services.

In eerste instantie dacht ik dat ik de verkeerde UUID meegaf aan de functie. Om erachter te komen welk UUIDs er dienen te worden opgegeven is uitgezocht hoe het Bluetooth Serial Port Profile (SPP) in elkaar zit. SPP is het protocol wat gebruikt wordt door Bluetooth GPS-ontvangers om te communiceren met uitleesapparatuur. Via Google kwam ik terecht op een website¹² waarop stond dat het Bluetooth SPP gebruik maakt van RFCOMM dat “provides serial port emulation, enabling Bluetooth support for serial data connections”.

Het herleiden wat de UUID van RFCOMM was kon eenvoudig via de Javadocs van de Bluetooth API (klasse `UUID`). Hierin was vermeld dat de UUID van RFCOMM was : `0x0003`. Precies het UUID waarna in eerste instantie reeds was gezocht. Echter zoals al eerder vermeld kwam hier niets uit. Aangezien bleek dat ik naar de correcte services zocht, zat hier niet de essentie van het probleem.

Daar mij nog steeds onbekend was waar nu de fout zat ben ik via Google verder gaan zoeken naar informatie over hoe ik connectie kon maken met een Bluetooth GPS-ontvanger. Ik kwam uit op www.benhui.net . Op deze website was enige broncode te vinden waarmee duidelijk werd dat de service discovery pas kan worden aangeroepen indien device discovery voltooid is (door de discoverylistener wordt dan de functie `inquiryCompleted` aangeroepen). Dit was tegenstrijdig met wat ik begrepen had uit het Sun artikel.

¹² (<http://www.phptr.com/articles/article.asp?p=21336&seqNum=3>)

Na dit aangepast te hebben in mijn broncode had ik verwacht dat zowel het zoeken naar devices alsmede het zoeken naar services zou slagen. Niets was echter minder waar. Met de aangepast code werden nog steeds geen services ontdekt.

Op dit moment twijfelde ik eraan of de telefoon dan wel de Bluetooth GPS-ontvanger die ik gebruik wel goed werkte. Hierdoor ben ik eerst gaan zoeken op het forum van Siemens waar vermeld stond dat de Siemens S65 problemen heeft om met sommige Bluetooth apparaten te communiceren. Er werd geadviseerd om een nieuwe versie van de firmware te installeren om het probleem op te lossen. Na het installeren van de firmware was het probleem inderdaad opgelost en werden uiteindelijk de services ontdekt.

Gebruik Bluetooth Control center (BCC)

Vanwege de voordelen van het BCC zoals uitgelegd in hoofdstuk 6.9 is het BCC geïmplementeerd in het testprogramma.

```
// retrieve the discovery agent
agent = local.getDiscoveryAgent();
// return an array of pre-known devices
devices = agent.retrieveDevices(DiscoveryAgent.PREKNOWN);
```

Alle bekende (PREKNOWN) devices in het BCC worden hierdoor opgeslagen in de devices array.

Zoals vermeld is het testprogramma gemaakt om contact te maken met en het ontvangen van de data van de Bluetooth GPS-ontvanger. In deze paragraaf is besproken hoe het proces is verlopen met betrekking tot het contact leggen met de Bluetooth GPS-ontvanger, in het volgende paragraaf zal dit gedaan worden voor het ontvangen van de GPS data.

8.5 Ontvangen GPS informatie van Bluetooth apparaat

Nu het testprogramma met Bluetooth apparaten contact kon leggen was de laatste stap het ontvangen van de GPS-data van de Bluetooth GPS-ontvanger. Hiervoor dient er eerst een connection te worden gemaakt alvorens de informatie kan worden ontvangen. Zowel in het artikel van Java.sun.com, als in de voorbeeldcode werd gebruik gemaakt van een Streamconnection waarbij een connection url werd meegegeven aan de connector.open functie alvorens een inputStream geopend werd.

```
StreamConnection connection = (StreamConnection) Connector.open(url);
InputStream is = connection.openInputStream();
```

Deze connection url wordt samengesteld aan de hand van de informatie die verkregen is bij de servicediscovery zoals besproken in de vorige paragraaf. Aangezien zoals eerder reeds aangegeven het in eerste instantie niet lukte om services te discoveren is er destijds gekeken of deze connection url niet zelf samengesteld kon worden. Het bleek dat de connection url er standaard zo uit ziet

```
url = "btspp://" + <Bluetooth address> + ":" + <channel> +
";encrypt=false;authenticate=true;master=false"
```

Aangezien het Bluetooth address van het device waarmee verbinding gemaakt moet worden, verkregen is bij de devicediscovery was alleen het channel nog onbekend. Op het forum van benhui.net stond hierover een topic en er bleek dat het channel wat het meest gebruikt werd channel 1 is. Probleem zit hem echter in het woord meest. Sommige keren werd namelijk gebruik gemaakt van een ander channel. Om de stabiliteit van het programma te verhogen zouden dus één voor één alle channels langsgelopen moeten worden indien gebruik van

channel 1 niet van toepassing is. Hoewel dit echter technisch mogelijk is, lag het probleem in het feit dat het verbinding maken met een GPS-ontvanger dan soms wel (heel) lang kan gaan duren waardoor het programma (veel) trager zal worden.

De oplossing voor dit probleem kwam echter van dezelfde site (benhui.net) aangezien op deze site een voorbeeld van een Java code te vinden is over het zoeken naar Bluetooth devices en de services die daarop draaien. Zoals reeds eerder verteld werd aan de hand van deze code duidelijk dat er niet direct naar services gezocht moesten worden indien een device gevonden was maar dat dit pas moest gebeuren nadat de inquiry completed was.

Aan de hand van de informatie is het testprogramma aangepast en toen werkte de service discovery en werd de informatie (incl. gewenste channel) verkregen over de service die draaide op de Bluetooth GPS-ontvanger. Aan de hand van deze informatie kon telkens de juiste connection url bepaald worden met het juiste channel reeds ingevuld. Het maken van een Streamconnection werkte en het openen van een inputstream leverde tevens geen problemen op.

Uitlezen GPS data uit Bluetooth ontvanger

Nu er een connectie gemaakt kon worden met de Bluetooth GPS-ontvanger was de volgende stap het uitlezen van de GPS data. Hiervoor werden door beide eerder gebruikte bronnen (artikel op Java.sun.com & source code van benhui.net) gebruik gemaakt van:

```
is.read(buffer);
```

Hierbij wordt de ontvangen data in z'n geheel in de buffer geplaatst. Deze buffer is een byte array. Voor het weergeven van deze data in het testprogramma was het echter nodig om de byte array te converteren naar een string. Echter de output zoals na het converteren naar String werd weergegeven was incorrect.

Deze output was leek altijd op [B@f225d76d terwijl de normale NMEA sentences zouden moeten worden weergegeven. Een voorbeeld hiervan is:

```
$GPGGA,123519,4807.038,N,01131.000,E,1,08,0.9,545.4,M,46.9,M,,*47
```

Allereerst dacht in het feit dat de kern van het probleem lag bij de connection url zoals eerder hierboven besproken. Aangezien de parameter "encrypt" in deze voorbeeld url voorkomt dacht ik dat deze mogelijk op "true" stond en dat de ontvangen data dus was versleuteld.

Uit de documentatie bleek echter dat de encryptie van de data intern geregeld wordt en dat de data zoals opgeslagen in de buffer de NMEA sentences "gewoon" leesbaar zouden moeten zijn zonder enige vorm van decryptie.

De volgende dag dat ik het testprogramma draaide, werd er helemaal geen data meer ontvangen. Aangezien de code niet veranderd was sinds de vorige dag was in eerste instantie onduidelijk waar de bottleneck zat. Aangezien het programma de vorige dag was vastgelopen en sindsdien de telefoon niet herstart was, werd in eerste instantie gedacht dat hier de fout zat. Hoewel de normale functies van de mobiele telefoon gewoon doorwerkte kon het echter zo zijn dat er nog geheugenruimte voor het programma in gebruik waren of dat er dingen nog niet waren vrijgegeven.

Het simpel herstarten van de telefoon maakte echter niet veel uit op de werking van het programma en de GPS data werden wederom niet gevonden/weergegeven. Om er achter te komen waar de fout zat is de code opgedeeld in verschillende stukken met elk een aparte foutmelding indien er iets fout ging. Daarnaast werd informatie over de correct

functionerende code weergegeven op het scherm van de mobiele telefoon. Hieruit bleek dat het ophalen van de preknown devices uit het BCC goed verliep en dat het openen van een connectie en de inputstream geen problemen opleverde. Bij de code die de GPS data ontvangt ging er echter iets fout en er werd een foutbericht weergegeven welke door Java zelf wordt gegenereerd. Dit foutbericht was echter weinig zeggend, namelijk “null”. Hoewel nu bekend was waar het probleem zat was weinig bekend over de oorzaak hiervan.

Aangezien er drie partijen betrokken zijn bij het programma namelijk het Java programma, het Bluetooth control center en de GPS-ontvanger moest bij 1 van deze 3 de fout zitten. Omdat het programma ongewijzigd was, kon hier de fout niet zitten. Daarnaast was de telefoon herstart waardoor eerder gebruikte geheugenruimte weer was vrijgegeven en dus weer beschikbaar was. Daar aan de GPS-ontvanger ook niets was veranderd moest de oorzaak van het probleem bij het Bluetooth control center liggen. Omdat de enige functie die het BCC heeft het vastleggen van de bekende Bluetooth apparaten is, werd daarom de Bluetooth GPS-ontvanger opnieuw geautoriseerd in het BCC. Na dit gedaan te hebben werd de GPS data weer ontvangen en weergegeven door het programma. Hieruit bleek dat de Bluetooth apparaat iedere keer dat de GPS-ontvanger opnieuw is opgestart deze wederom moet worden geautoriseerd in het BCC door middel van het invoeren van de passcode. Dit in tegenstelling tot wat er in de handleiding van de mobiele telefoon is vermeld : “Vertrouwde apparaten hebben altijd toegang”.

Klaarblijkelijk is het eenmalig invoeren van de GPS-ontvanger in het BCC niet afdoende voor een permanente autorisatie. Hierdoor is gekeken of er in de code aanpassingen moesten worden gedaan om deze permanente autorisatie te kunnen bewerkstelligen. Dit omdat het wenselijker is dat de GPS receiver niet iedere keer moet worden geautoriseerd maar dat dit slechts eenmalig hoeft te geschieden.

Na via Google naar informatie te hebben gezocht over dit onderwerp bleek dat er twee mogelijkheden waren om autorisatie van de GPS-ontvanger in de code te bewerkstelligen. Voorwaarde hiervoor was wel dat het apparaat bekend moet zijn in het BCC. Aangezien de GPS-ontvanger bekend was in het BCC zijn beide mogelijkheden geprobeerd.

Allereerst zou het autoriseren mogelijk moeten zijn via de connection url zoals deze eerder ook is besproken. Indien om via de connection url de parameter “authorize=true” zou worden meegegeven dat de GPS-ontvanger zou worden geautoriseerd. Hierdoor is gechecked of deze parameter reeds verwerkt was in de connection url die automatisch werd gegenereerd uit de informatie die verkregen was bij de service discovery. Automatisch verkrijgen van de connection url gaat via de volgende code :

```
url = device.getConnectionURL(ServiceRecord.AUTHENTICATE_NOENCRYPT, false);
```

Hierbij kan de gewenste vorm van de connection url bepaald worden door de parameter “ServiceRecord.AUTHENTICATE_NOENCRYPT”. Hierdoor vraag je om een connection url die het Bluetooth apparaat authenticceert. Hieruit kwam de volgende url:

```
btssp://00081b0a5151:1;encrypt=false;authenticate=true;master=false
```

Dit was de url zoals deze momenteel gebruikt werd door het programma en waarmee eenmalige autorisatie in het BCC niet afdoende was. Aangezien in deze connection url de gewenste parameter “authorize=true” ontbreekt is deze toegevoegd om de volgende manier alvorens getracht werd om een connectie tot stand te brengen.

```
url = url + ";authorize=true";
```

Door deze toevoeging veranderde de werking van het programma echter niet en was het iedere keer opnieuw invoeren van de passcode nog steeds vereist.

Aangezien de eerste van de twee mogelijkheden om autorisatie van de GPS-ontvanger via de code te bewerkstelligen niet naar behoren functioneerde, is vervolgens de tweede optie geprobeerd. Dit hield in om de Bluetooth ontvanger na het maken van een connectie te autoriseren via de volgende code:

```
// authorize  
remoteGPS.authorize(conn);
```

Echter na gebruik van deze code was het euvel nog steeds niet verholpen. Omdat er verder geen informatie op het internet te vinden was over hoe dit probleem op te lossen is er eerst verder gegaan met de rest van het programma.

Het ontvangen van de GPS data waarbij het converteren van de byte array naar een String incorrect geschiedde. De output was namelijk telkens soortgelijk aan: [B@f225d76d. Gezien het feit dat het Bluetooth SPP een seriële communicatie simuleert alsmede het feit dat de data zoals werd ontvangen niet waren versleuteld werd via de inputstream “normale” binaire code ontvangen zoals opgeslagen in de byte array.

Hierdoor moet de gebruikte methode de data incorrect converteren. Het converteren gebeurde via de toString functie van de byte[] en het leek erop dat de data door de functie niet op de gewenste manier werd gedecodeerd. Aangezien deze output zeer waarschijnlijk wel verkregen was uit de Bluetooth GPS-ontvanger leek het alsof de data op een verkeerde manier werd gedecodeerd. Aangezien de data waarschijnlijk binair binnen komt is gekeken naar andere talstelsels omdat er in de documenten van de Bluetooth SPP geen informatie over de codering kon worden gevonden. Voordat er werd gekeken werd naar andere talstelsels is eerst opgezocht in de Javadocs naar welk talstelsel door de toString functie van de byte array werd geconverteerd. Dit was base 10. Eerst is geprobeerd om via andere manieren namelijk zoals naar hexadecimaal:

```
for (i=0; i < buffer.length; i++ ) {  
    temp = temp.concat(Integer.toHexString(buffer[i]));  
}
```

En op een soortgelijk manier werd octale code geprobeerd. Dit gaf echter ook niet het gewenste resultaat. Na geen van deze talstelsels werd de GPS data correct weergegeven. Hierdoor is gekeken naar alternatieven om een byte array te converteren naar een String. Via de Javadocs van de klasse String (Java.lang.String) kwam ik uit op de volgende twee mogelijkheden :

```
int l = is.read(buf);  
rs.close();  
String result = new String(buf, 0, l);
```

en

```
is.read(buffer);  
GPSStr = new String(buffer);
```

Bij geen van beide mogelijkheden verscheen er echter ook maar iets op het beeldscherm.

Alternatief voor uitlezen GPS data

Aangezien dit niet werkte heb ik via Google gezocht naar andere mogelijkheden om de GPS data van een Bluetooth ontvanger uit te lezen. Ik kwam uit op een topic op het forum van Java.sun.com¹³, bij een persoon die een soortgelijk programma maakte voor een Java telefoon. Hij schreef hierin dat het hem gelukt was om de GPS data vanaf een Bluetooth GPS-ontvanger correct te ontvangen van alle NMEA sentences (4 stuks), echter na twee keer liep het programma telkens vast. Hij wist niet waar het probleem zat maar dacht dat de oorzaak bij de buffer zat. Geïntrigeerd door het feit dat het in 1^{ste} en 2^{de} instantie wel werkte heb ik contact opgenomen met deze persoon en van hem ontving ik de volgende code:

```
ByteArrayOutputStream byteArrayOutputStream = new ByteArrayOutputStream();
int ch = 0;
//each NMEA Message ends with <CR><LF>
while ( (ch = in.read()) != '\n') {
    byteArrayOutputStream.write(ch);
}
byteArrayOutputStream.flush();
byte[] b = byteArrayOutputStream.toByteArray();
String GPSStr = new String(b);
```

De code vertoonde bij mij hetzelfde gedrag en na het twee keer ontvangen van de GPS data bleef het scherm hangen. Opmerkelijk hier was wel dat in deze het converteren van een byte array naar een String door middel van “GPSStr = new String(b)” wel op het scherm werd getoond. Dit terwijl deze methode eerder helemaal niets op het scherm toonde (zie hierboven).

Via Google heb ik vervolgens gezocht naar hoe gebruik te maken van een ByteArrayOutputStream. Dit leverde echter weinig resultaten op waardoor er op verschillende fora berichten zijn gezet met de vraag om hulp.

Tijdelijke work around

Hoewel voor het uitlezen van de GPS data verschillende soorten codes waren geprobeerd van verschillende bronnen kon er nog geen continue stroom van data worden ontvangen vanuit de Bluetooth GPS-ontvanger. Het implementeren van het ontvangen van de GPS data had reeds veel meer tijd gekost dan van tevoren was ingepland. Hierdoor werd besloten om eerst verder te gaan met de rest van het project alvorens het probleem op te lossen.

Het ontvangen van de GPS data zou worden gesimuleerd door middel van een array waarin NMEA sentences waren opgeslagen. Het uitlezen van de GPS data zou dan iedere seconde via het uitlezen van de volgende rijen in de array gebeuren. Hierdoor kon voorlopig worden gewerkt aan de rest van het project. Tegelijkertijd werden er op verschillende fora berichten gezet over mijn probleem. Onder andere op de websites van Sun en van Benhui.net.

Oplossing uitlezen GPS data

Hoewel werd verwacht dat de oplossing van het probleem zou komen via één van de fora kwam deze uit een heel andere hoek. Een kennis vertelde te beschikken over Bluetooth op zijn computer. Voor mijn project mocht ik tijdelijk gebruik maken van zijn computer op mijn programma te debuggen. Dit was met name interessant aangezien via een computer veel meer foutmeldingen en output worden weergegeven dan via de textbox van het programma op mijn telefoon. Op de computer kan namelijk gebruik worden gemaakt van de System.out.print() functie. Hierop is door mij ook een Bluetooth dongle aangeschaft.

¹³ <http://forum.java.sun.com/thread.jspa?forumID=76&threadID=570897>

Via deze functie kwam tijdens het debuggen een melding naar voren die op de mobiele telefoon niet werd weergegeven. Deze melding hield in een “IllegalArgumentException” bij de functie setChars() van de textbox. Uit de Javadocs bleek dat oorzaak hiervan was dat “the text would exceed the current maximum capacity”. Oftewel de data die in de textbox zou moeten komen te staan was te groot. Aangezien na iedere meting de inhoud van de textbox werd gewist kon het niet zo zijn dat de textbox hierdoor verstopt was geraakt. Er bleek inderdaad dat er voor de textbox een maximum van 128 characters was gereserveerd terwijl dit bij de buffer 500 was. Na het vergroten van maximale grootte in karakters van de textbox naar 512 was het probleem opgelost en werd er een continue stroom van GPS data ontvangen.

Deze continue stroom van GPS data leidde echter tot een tweede opstopping vanwege de flush functie van de bytearrayoutputstream. Bedoeling was dat deze functie alle binnen gekomen data zou wissen uit de bytearrayoutputstream om te veel gebruik van geheugenruimte te voorkomen. Dit werd echter niet gedaan. Hierdoor maakte het programma gebruik van te veel geheugenruimte en liep het programma vast op zowel de computer als de mobiele telefoon. Op de computer met Bluetooth dongle werd dit echter direct gemeld terwijl bij het draaien van het programma op de mobiele telefoon dit niet het geval was. De mobiele telefoon gaf wederom geen foutmelding anders dan “null”. Echter door de informatie van de computer kon de flush functie eenvoudig worden vervangen door de reset functie die het vereiste werk wel correct deed.

Hierna werd ook nog de eerder besproken code die vermeld stond in de Javadocs van de String klasse voor het uitlezen van de GPS data geprobeerd en deze werkte ook ! Echter toString() functie van de byte array werkte nog steeds niet naar behoren echter aangezien het probleem was opgelost is hier verder niet meer naar gekeken.

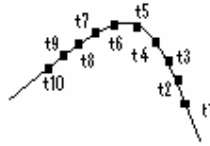
8.6 Bepalen afgelegde afstand

Bij het berekenen van de tijd waarin een wedstrijd (2000 meter, zoals vastgelegd in de randvoorwaarden) wordt afgelegd is het essentieel om de afgelegde afstand zo nauwkeurig mogelijk te bepalen. Dit is bij GPS met name lastig bij zeer bochtig parcours zoals bijvoorbeeld hardlopen in het bos. Indien een hardloper namelijk een bocht zou maken zou dit de nauwkeurigheid van het meten van de afgelegde afstand met behulp van GPS kunnen vertroebelen bij een situatie zoals weergeven in onderstaande plaatje.



Stel dat de hardloper neemt de bocht zoals weergegeven door de continue lijn. Echter de GPS heeft enkel metingen gedaan van ruim voor de bocht (t1) als wel een meting ruim na bocht (t2). De GPS zal dan bij het bepalen van de afgelegde afstand de meest korte afstand tussen de twee meetpunten bereken en dat is de gestippelde rechte lijn. Dit ondanks het feit dat de hardloper daadwerkelijk een veel grotere afstand heeft afgelegd.

De oplossing voor dit probleem zou zijn om het aantal metingen (drastisch) te verhogen, zoals weergegeven in het onderstaande plaatje.



Tijdens een vergadering is dit punt ook naar voren gekomen en samen met de bedrijfsmentor is overlegd of dit issue ook van toepassing zou zijn op dit project. Aangezien het programma bedoeld is om tijden te klokken over de afgelegde wedstrijdafstanden (2000 meter) is eerst gekeken naar wat voor soort water deze wedstrijden worden afgelegd.

Een groot deel van de wedstrijden waaraan teams van Pelargos aan deelnemen vindt plaats op de bosbaan (zie bijlage 2). Daarnaast is gekeken naar het roeiwater waar Pelargos haar trainingen uitvoert en dus het water waarop het programma gebruikt zal gaan worden. (zie bijlage 3).

Uit de analyse van de roeiwateren waarop het programma gebruikt zal gaan worden bleek dat deze over het algemeen zeer recht waren. Hierdoor is besloten om het programma dusdanig te implementeren met als Ausgangssituation dat het alleen wordt gebruikt bij wedstrijden op rechte stukken roeiwater.

Bereken afgelegde afstand

Hoewel er tijdens de afgelegde wedstrijdafstand weinig kromming in de aarde is zal gebruikt gemaakt worden van een ellipsoid model of WGS84 welke geïmplementeerd is in de location API uitgegeven door Nokia. Deze functie kan de afstand tussen twee opgegeven GPS locaties berekenen waarin de kromming van de aarde in mee wordt genomen. Over de functie waarin dit is geïmplementeerd staat in de Javadocs het volgende :

“Calculates the geodetic distance between the two points according to the ellipsoid model of WGS84. Altitude is neglected from calculations.

The implementation shall calculate this as exactly as it can. However, it is required that the result is within 0.35% of the correct result. “

Hieruit blijkt dat het berekenen van de afgelegde afstand door middel van WGS84 weinig (extra) onnauwkeurigheid inbrengt aan het GPS, zoals onderzocht in de eerste pilot (zie hoofdstuk 9).

HOOFDSTUK 9. PILOT 1 - BENADEREN NAUWKEURIGHEID VAN GPS

Tijdens de eerste iteratie van het project is de nauwkeurigheid van GPS benaderd en gekeken of GPS nauwkeurig genoeg zou zijn voor dit project. Aangezien dit systeem gebruikt gaat worden voor tijdmetingen tijdens trainingen van competitieroeiers hoeft het programma niet tot op de tiende seconde nauwkeurig te zijn. De bedrijfsmentor stelde als eis de wetenschappelijke nauwkeurigheid (99%).

Vanwege deze eis is deze iteratie ingevoerd. Indien mocht blijken dat deze eis niet gehaald kon worden zou dit betekend hebben dat het project zou stoppen. Om dit te voorkomen is een iteratie ingebouwd waardoor de mogelijk zou worden gecreëerd om terug te gaan naar de definitiestudie indien de 99% eis niet gehaald kon worden. In deze definitiestudie kon dan gekeken worden naar mogelijke alternatieven of de eis zou dan kunnen worden versoepeld.

Haalbaarheidsstudie studente Geodesie

Bij het vaststellen van de uitgangssituatie is er via internet een haalbaarheidsstudie gevonden van een studente geodesie. Deze had als onderwerp “GPS & Sport”. Op meerdere mails werd echter niet gereageerd door de betreffende student. Hierdoor kon geen gebruik gemaakt worden van deze bron.

1^{ste} verkenning 99% eis

In een eerste verkenning door de afstudeerder bij het maken van het plan van aanpak (waarin de 99% eis is vastgesteld) bleek dat deze eis waarschijnlijk haalbaar zou zijn. Deze eis gold voor een wedstrijdafstand van 2.000 meter wat betekent dat de onnauwkeurigheid maximaal 20 meter mag zijn (1% van 2.000 meter). Uit de documentatie van producenten van de GPS-ontvangers werd vermeld dat de nauwkeurigheid van EGNOS en WAAS rond de 7 meter lag en dat XTrac een nauwkeurigheid had van 5 tot 25 meter (zie ook hoofdstuk 2.1). Hierbij geldt dat de 25 meter nauwkeurigheid enkel wordt behaald bij slecht zicht in de open lucht (zie hoofdstuk 6.5 voor meer informatie). Aangezien op open roeiwater een goed zicht is in de open lucht zou deze 25 meter nauwkeurigheid waarschijnlijk niet gehaald worden maar in plaats daarvan lager liggen dan deze 25 meter.

Bij terugrekening van deze afwijkingen zou dit bij een wedstrijdafstand een onnauwkeurigheid betekenen van 99,75% tot 99,85% voor EGNOS en 99,85% nauwkeurigheid voor Xtrac. Incalculerend dat de producenten mogelijk de nauwkeurigheid hoog inschatten, zal er waarschijnlijk voldoende marge overblijven om boven de 99% nauwkeurigheid te blijven. Deze grens ligt namelijk bij een onnauwkeurigheid van 20 meter, wat respectievelijk een factor 4 en factor 6 2/3 is van hetgeen opgegeven is door de producenten.

Benaderen (on)nauwkeurigheid

Uit de eerste verkenning is gebleken dat de 99% eis waarschijnlijk gehaald wordt: met het oog daarop is deze nauwkeurigheid in de praktijk gemeten met een mobiele telefoon.

Bij het benaderen van de onnauwkeurigheid is uitgegaan van de reeds in het bezit zijnde GPS-ontvanger van de afstudeerder. Aangezien de GPS-ontvanger beschikt over het softwarematig nauwkeurigheidssysteem XTrac (zie hoofdstuk 2.1) moet wel gezegd worden dat deze GPS-ontvanger nauwkeuriger is dan minder geavanceerde GPS-ontvangers. Dit geldt zeker voor open plaatsen waar dus veel GPS satellieten in het zicht zijn zoals het geval is bij roeiwater. Uitleg over type GPS-ontvangers en over de overwegingen bij de keuze van een GPS-ontvanger is te vinden in hoofdstuk 6.5.

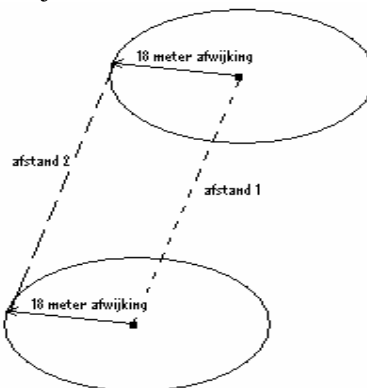
Bij het vaststellen van de mogelijke horizontale afwijking van de gemeten GPS-positie is uitgegaan van de data die door de GPS-ontvanger hierover zelf wordt verstrekt. Bij elke meting wordt namelijk erbij gemeld wat de mogelijke horizontale afwijking van de meting is. Om een betrouwbaardere benadering te krijgen zijn deze waarden van de mogelijke horizontale afwijking van de metingen over een afstand van 2000 meter gemeten. Hierbij werd iedere seconde de GPS data uitgelezen. Dit aangezien de GPS-ontvanger deze informatie één keer per seconde levert. De gemiddelde waarde die hieruit kwam bedroeg 12 meter. Gemiddeld aantal satellieten dat hierbij gebruikt werd, was 7. Er kunnen meer satellieten in het zicht geweest zijn: echter indien het signaal te zwak is, worden de data van deze zwakke signalen niet meegenomen in de berekening van de GPS locatie. Uit deze metingen komt een nauwkeurigheid voort van 99,4 %. Tijdens de meting had de GPS-ontvanger open zicht op de lucht en bevond zich bijvoorbeeld niet in een jaszak. Indien de GPS-ontvanger zich bevindt in een jaszak of iets anders wat het zicht op de (open) lucht verhindert, vermindert dit de kwaliteit van de metingen.

Aangezien de metingen over het algemeen weinig van elkaar verschilden, betekent dit dat de mogelijke horizontale afwijking van de GPS-ontvanger constant is.

Bepaling horizontale afwijking

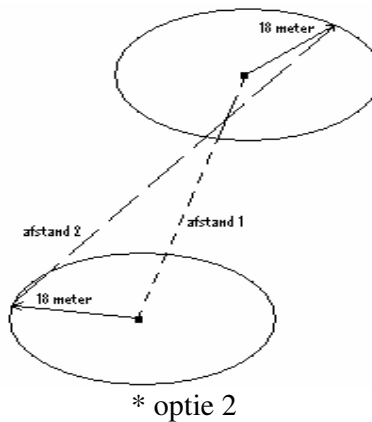
Omdat deze mogelijke horizontale afwijking constant is, werkt dit in het voordeel van dit project aangezien het niet primair wordt gebruikt voor positiebepaling maar voor het berekenen van (afgelegde) afstanden. Het was de afstudeerder echter onduidelijk hoe deze horizontale afwijking door de GPS-ontvanger werd berekend.

Er zijn namelijk twee mogelijkheden denkbaar. Bij berekenen van de afgelegde zou deze afwijking zichzelf namelijk verwaarlozen indien gesproken kan worden over de situatie zoals weergegeven in onderstaand plaatje.



* optie 1

Het zou echter een ander verhaal zijn indien er sprake zou zijn van een situatie zoals weergegeven in onderstaand plaatje.



Om te weten te komen of beide opties zoals hierboven weergegeven mogelijk zijn is hierover meer informatie ingewonnen. Dit is reeds gebeurd voor aanvang van het afstuderen toen bij het goedkeuren van de afstudeeropdracht ik, op advies van de vakdocent, een afspraak had met de heer De Vreught aangaande inwinnen informatie over verschillende methoden (zie hoofdstuk 5.4). Tijdens dit gesprek kwam de bepaling van de onnauwkeurigheid van GPS ter sprake. Hij kon mij vertellen dat optie 1 het meest waarschijnlijke is aangezien gedurende een wedstrijd gebruik zal worden blijven gemaakt van dezelfde satellieten voor de berekening van de GPS positie. Een wedstrijd duurt namelijk slechts 7 á 8 minuten waardoor de kans groot is dat er aan het eind van de wedstrijd dezelfde satellieten ‘in beeld’ zijn als bij het begin van de wedstrijd.

Door deze aanname zou dit betekenen dat de GPS positie altijd dezelfde horizontale afwijking heeft waardoor deze zichzelf verwaarloost bij het berekenen van de afgelegde afstand, zoals weergegeven in optie 1 hierboven.

Extra onnauwkeurigheid door bereken afstand

Zoals vermeld in hoofdstuk 8.6 wordt voor het berekenen van de afstand gebruik gemaakt van de location API van Nokia. Deze berekening gebeurt aan de hand van het ellipsoid model van WGS84¹⁴ waarbij rekening wordt gehouden met de kromming van de aarde. Ondanks de opname van deze extra variabele is de onnauwkeurigheid van de berekening van de afstand tussen twee GPS posities 0.35% (bron: Javadocs van Location API).

Hieruit blijkt dat het berekenen van de afgelegde afstand door middel van WGS84 weinig (extra) onnauwkeurigheid inbrengt in de berekening van de afstand.

Conclusie

Aangezien aan de eisen werd voldaan zoals gesteld voor de eerste iteratie, zoals uitgelegd in hoofdstuk 5.4, hoefde de definitiestudie niet nogmaals te worden uitgevoerd.

De mogelijkheid om via de iteratiestrategie evolutionair ontwikkelen het stoppen van het project te voorkomen als de 99% nauwkeurigheid niet zou worden gehaald, bleek dus achteraf een overbodige luxe.

¹⁴ <http://www.wgs84.com/>

HOOFDSTUK 10. PILOT 2 – VERIFIËREN BRUIKBAARHEID PROGRAMMA DOOR DE COACHES

Als tweede iteratie voor dit project zou geverifieerd worden of GPS nauwkeurig genoeg is voor de coaches die het programma gaan gebruiken en of het programma bruikbaar is voor de coaches. Hiervoor zouden zij gebruik maken van een eerste testversie van het programma.

Aangezien het implementeren van het programma meer tijd heeft gekost dan van tevoren was ingepland is hiervoor echter nog geen mogelijkheid geweest.

Hoewel het programma niet met coaches is overlegd, zoals voor de tweede iteratie diende te gebeuren, is het wel besproken met de bedrijfs-mentor. Deze is oud commissaris roeien van Pelargos en in het bezit van coachcursus A van de Koninklijke Nederlandse Roeibond (KNRB). Hij was tevreden over de nauwkeurigheid, zoals hierboven reeds is vermeld.

Daarnaast is de schets van het beeldschermontwerp in de architectuurfase aan de bedrijfsmentor overlegd (zie hoofdstuk 7.2).

Tevens is tijdens de ontwikkelfase het gemaakte testprogramma voorgelegd aan de bedrijfsmentor (zie hoofdstuk 8.2). Deze heeft het testprogramma op de mobiele telefoon meerdere keren doorgelopen en was tevreden over de werking.

Hoewel de bruikbaarheid van het programma dus niet in de praktijk is getest, is het prototype doorgenomen op de mobiele telefoon door de bedrijfsmentor. Daarnaast is een schets van het beeldschermontwerp tijdens één van de eerste meetings met de bedrijfsmentor overlegd en goedgekeurd zoals reeds uitgelegd in hoofdstuk 7.2 en hoofdstuk 8.2.

HOOFDSTUK 11. EVALUATIE

11.1 Proces evaluatie

In deze paragraaf zal het proces van dit project geëvalueerd worden. Persoonlijk heb ik een goed gevoel bij het verloop van het project. Over vele kanten van het project is eerst nagedacht onder andere in de meetings met de bedrijfsmentor. Daarnaast is er eerst nagedacht over de implementatie van het programma.

Dit blijkt met name uit het maken van de verschillende testprogramma's aangezien het gebruik van Bluetooth, RMS en de MIDP mij nog onduidelijk waren aan het begin van het project. Door het maken van deze testprogramma's was de werking van deze onderdelen mij veel duidelijker op het moment dat ik aan het echte programma begon. Hierdoor is de opbouw van het programma gestructureerder wat blijkt uit het klassediagram waarin de genoemde onderdelen een eigen klasse hebben.

IAD als methode

Aangezien vanuit mijn opleiding methoden als IAD niet behandeld zijn was dit nieuw voor mij. Bij het gebruik van IAD tijdens dit project werd ik hier echter in ondersteund door de bedrijfsmentor en de heer De Vreught. Het voordeel van het gebruik van IAD was met name het definiëren van de iteraties. Door van te voren de mogelijke knelpunten bepaald te hebben, is veel meer voorkomen dat we tijdens het project voor verrassingen kwamen te staan. Door het definiëren van de 99% eis als iteratie was dus van tevoren bekend dat er eventueel terug zou moeten worden gekeerd naar de definitiestudie. Dit geeft veel meer duidelijkheid en zekerheid voor zowel de opdrachtgever als opdrachtnemer.

Dubbele rol

Naast dat ik de afgelopen maanden bij de H.S.R.V. Pelargos afstudeerde was ik tevens bestuurslid. Deze combinatie heeft naar mijn mening echter geen problemen opgeleverd. Hoewel de gesprekken met de bedrijfsmentor natuurlijk ook deels over algemene zaken binnen de vereniging gingen waren de meetings toch zeker "to the point".

Ontbreken technische begeleiding

Één van de risicofactoren bij dit project was het ontbreken van technische begeleiding vanuit de H.S.R.V. Pelargos, zoals aangegeven in hoofdstuk 5.5. Tijdens het project is gebleken dat ik bij technische vraagstukken vooral gebruik heb gemaakt van digitale bronnen als websites en fora. Met name het contact maken met en het uitlezen van de Bluetooth GPS-ontvanger heeft mij hierdoor extra tijd gekost.

Nadat ik iemand met technische achtergrond had geraadpleegd was het probleem veel makkelijker op te lossen door het gebruik van een Bluetooth dongle. Door deze tip ging het ontwikkelen en debuggen van het programma veel sneller. Indien ik deze technische begeleiding eerder had gezocht zou het product waarschijnlijk wel af zijn geweest.

Ik ben echter tevreden met het gebruik van de work around nadat het programma vastliep na het enkele malen ontvangen van de GPS data terwijl de oorzaak van het probleem onbekend was. Door het simuleren van het ontvangen van de GPS data via het uitlezen van een array kon echter verder worden ontwikkeld aan het programma.

11.2 Product evaluatie

In deze paragraaf zal het product geëvalueerd worden.

Stand van zaken product

Met de huidige versie van het programma kan bij een 2000 meter wedstrijd worden gemeten in welke tijd deze is afgelegd en worden de gemeten waardes real-time weergegeven op het beeldscherm. Daarnaast is het beeldschermontwerp en het bewerken en beheren van de recordstores volledig geïmplementeerd.

Een overzicht van de metingen wordt echter nog niet weergegeven. Deze data kan daarnaast nog niet worden geëxporteerd naar een computer.

Hoewel deze twee punten wel vereist waren door de opdrachtgever is hier niet tijdens het project aan toegekomen. Dit omdat het implementeren van de code voor het verbinding maken met de GPS ontvanger via Bluetooth en het ontvangen van de GPS data meer tijd heeft gekost dan van tevoren verwacht. Daarnaast was bij het maken van de planning nog niet duidelijk dat het programma door middel van Bluetooth zou moeten communiceren aangezien het programma in eerste instantie voor een PDA zou worden gemaakt.

Opbouw product

Door het testen van de verschillende onderdelen (Bluetooth, RMS en MIDP) in een separaat testprogramma is er een duidelijker beeld ontstaan van de onderdelen voordat begonnen is met het echte programma. Hierdoor is de hoeveelheid overbodige broncode zeer beperkt met als gevolg dat de opbouw van het programma overzichtelijker en duidelijker is. Daarnaast is per functie uitleg gegeven over de werking ervan. Hiermee zijn Javadocs gegenereerd waarvan bij een eventuele vervolgstudie gebruikt kan worden gemaakt.

HOOFDSTUK 12. VERVOLGSTUDIES

Zoals in de afbakening van de opdrachtomschrijving voor dit project reeds is aangegeven zijn er mogelijkheden voor verdere uitbreiding van dit programma. Het gaat dan om het meten van het tempo waarmee de roeiers roeien door middel van het aanbrengen van sensor aan het bankje van de slag (de roeier die het tempo bepaald).

Naast het aanbrengen van een tempometer is het bijvoorbeeld ook mogelijk om een krachtmeter aan te brengen waarbij gemeten wordt hoeveel kracht de roeier individueel produceert. Hierdoor kun je van de verschillende roeiers zien wat hun bijdrage is aan de snelheid van de boot waarbij je te weten komt of de verschillende roeiers dezelfde inspanning doen.

Ook kan gedacht worden aan een hoekmeter welke op het scharnierpunt geplaatst kan worden waarover de roeiriem draait. Dit is handig om te weten voor de coach: hoe groter nl. de hoek die gemaakt wordt met de roeiriem, des te langer de roeihaal. En hoe langer de roeihaal is, des te langer de roeiboot wordt voortbewogen. Er dient tevens ook te worden voorkomen dat een roeier een te lange roeihaal maakt omdat dan de spieren in de rug te erg kunnen worden aangespannen. Dit kan een vervelende blessure veroorzaken indien dit niet tijdig opvalt.

Daarnaast kan er een hartslagmeter aan het programma op de GSM worden gekoppeld waarbij de hartslag van de roeier(s) wordt afgezet tegen de snelheid van de boot. Hierdoor krijg je een inspanningen/ snelheid tabel waaruit misschien af te leiden is bij welke hartslag je het best kunt roeien.

Om het verder ontwikkelen van het programma te vereenvoudigen is de werking hiervan gedocumenteerd, onder andere via commentaar in de code waarmee Javadocs zijn gegenereerd.

VERKLARENDE WOORDENLIJST

Bluetooth	Bluetooth is een technologie die het mogelijk maakt draadloos te communiceren tussen verschillende mobiele apparaten. Met Bluetooth kunnen gebruikers zonder kabels data verzenden tussen apparaten met een snelheid van maximaal 1 Mb per seconde over een afstand van maximaal 10 meter.
Bluetooth dongle	Uitbreidmodule voor Bluetooth op de computer.
Device	Term gebruikt door de Bluetooth API van Sun: bedoeld een Bluetooth apparaat.
GPS	Global Positioning System
Javadocs	Documenten over gebruik van de functies in Java
MIDP	Mobile Information Device Profile, set van Java API's die de functionaliteiten bieden voor apparatuur met beperkte processorkracht en geheugencapaciteit als bijvoorbeeld een mobiele telefoon
PDA	Zowel een Pocket PC als Palm.
Programma	Het softwareprogramma in micro Java dat ontwikkeld zal worden voor dit project.
Service	Een service die wordt verleend door een Bluetooth apparaat.
String	Hierin worden zinnen en karakters in een programmeertaal in opgeslagen.

GERAADPLEEGDE LITERATUUR

Bluetooth API implementeren (zoeken naar devices en services)

<http://developers.sun.com/techtopics/mobility/MIDP/articles/Bluetooth2/>
<http://today.java.net/pub/a/today/2004/07/27/bluetooth.html>

code voorbeelden benhui.net voor implementeren Bluetooth API

<http://www.benhui.net/modules.php?name=Bluetooth>

informatie over Location API (voor berekenen afgelegde afstand)

<http://www.jcp.org/en/jsr/detail?id=179>

informatie over EGNOS

http://www.esa.int/export/esaSA/GGG63950NDC_navigation_0.html

informatie over WAAS

<http://GPS.faa.gov/Programs/WAAS/waas.htm>

informatie over softwarematige nauwkeurigheidssystemen van GPS

<http://www.pocketGPS.co.uk/xtracornoxtrac.php>

informatie over nauwkeurigheid GPS tijdens roeien

http://www.vierzonder.nl/GPS_informatie.htm

informatie over RMS (Record Management System)

<http://today.java.net/lpt/a/149>

informatie over Bluetooth Serial Port Profile

<http://www.phptr.com/articles/article.asp?21336&seqNum=3>

informatie over schrijven van documentatie (Javadocs) voor eigen programma

<http://java.sun.com/j2se/javadocs/index.jsp>

informatie over programmeren op mobiele telefoon door middel van micro-Java

<http://java.sun.com>

Zocutto Wireless Inc., MIDP Primer

informatie over opbouw NMEA sentences

http://www.commlinx.com.au/NMEA_sentences.htm

R.J.H. Tolido, IAD het evolutionair ontwikkelen van informatiesystemen pilot 2, ISBN 90 39504016

AFSTUDEERVERSLAG

- BIJLAGEN -



Titel	: Pelargos Training Coach
Afstudeerblok	: 2004-2.2
Opdrachtgever	: Haagse Studenten Roeivereniging Pelargos
Bedrijfsmentor	: dhr. N. Leerentveld
Examinatoren	: dhr. H. Witkamp & dhr. J. van Peski
Afstudeerder	: dhr. G.J. de Boer
Versie	: 1.0
Plaats	: Den Haag
Datum	: 24 maart 2005

BIJLAGEN

1. Geschikte telefoons.
2. Roeiwater op de Bosbaan in Amsterdam.
3. Roeiwater op de Vliet in Voorburg / Rijswijk / Delft.
4. JAVA support on Pocket PC.
5. GPS en Galileo.
6. Test PDA's door Consumentenbond d.d. september 2004.

BIJLAGE 1. GESCHIKTE TELEFOONS

Hieronder een klein overzicht van de telefoons die geschikt zijn voor het programma wat wordt ontwikkeld. Dit overzicht is een ingekorte versie van de internet site :

<http://www.benhui.net/modules.php?name=Midp2Phones> . Op deze webpagina staan veel mobiele telefoons welke Bluetooth ondersteunen.

BenQ P31



Band	GSM 900/1800/1900	Heap size	Unknown
Screen	208 x 320	Max JAR size	Unknown
Color	65k	Memory size	Unknown
Data	GPRS Class 10 (48Kbps)	Platform	UIQ 2.1 Symbian 7.0
Supported API			
MMAPI 1.0	MIDP 2.0	Bluetooth API	CLDC 1.0 WMA 1.0

Link to [\[specification\]](#) [\[dev portal\]](#) [\[SDK\]](#) [\[review\]](#) [\[benchmark\]](#)

Samsung SGH-P400



Band	WCDMA 2000 / GSM 900/1800/1900	Heap size	16MB
Screen	208 x 320	Max JAR size	Unknown
Color	65k	Memory size	24Mb
Data	3G Data	Platform	UIQ 2.1 Symbian 7.0

Supported API				
CLDC 1.0	Bluetooth API	JTWI 1.0	MIDP 2.0	MMAPI 1.0
Mobile 3D API	Motorola Fun Lights API	WMA 1.0		

Network support HTTP, HTTPS, TCP, Server Socket, UDP.

Link to [\[specification\]](#) [\[dev portal\]](#) [\[SDK\]](#) [\[review\]](#) [\[benchmark\]](#)

Sony Ericsson P910



Band	GSM 900/1800/1900	Heap size	8Mb
Screen	208 x 320	Max JAR size	Unknown
Color	262k	Memory size	64MB
Data	GPRS	Platform	UIQ 2.1 Symbian 7.0

Supported API				
JTWI 1.0	Bluetooth API	CLDC HI	MIDP 2.0	MMAPI 1.0
Personal Profile	WMA 1.1			

Support Audio playback only.

Link to [\[specification\]](#) [\[dev portal\]](#) [\[SDK\]](#) [\[review\]](#) [\[benchmark\]](#)

Nokia 9500



Band	GSM 850 or 900/1800/1900	Heap size	Unknown
Screen	640 x 200	Max JAR size	Unknown
Color	65k	Memory size	90Mb
Data	GPRS Class 10 (48Kbps)	Platform	Nokia Series80 Symbian 7.0s

Supported API				
Bluetooth API	DRM Forward Lock	File Connection API	Foundation Profile	CDC 1.0
CLDC 1.1	JTWI 1.0	MIDP 2.0	MMAPI 1.0	Nokia S80 API
Personal Profile	PIM Access API	WMA 1.0		

Link to [\[specification\]](#) [\[dev portal\]](#) [\[SDK\]](#) [\[review\]](#) [\[benchmark\]](#)

Nokia 9300



Band	GSM 850 or 900/1800/1900	Heap size	20Mb
Screen	640 x 200	Max JAR size	Unknown
Color	65k	Memory size	80Mb
Data	EDGE	Platform	Nokia Series80 Symbian 7.0s

Supported API				
MMAPI 1.0	MIDP 2.0	JTWI 1.0	Bluetooth API	Foundation Profile
File Connection API	DRM Forward Lock	CLDC 1.1	CDC 1.0	Nokia S80 API
Personal Profile	PIM Access API	WMA 1.0		

Link to [\[specification\]](#) [\[dev portal\]](#) [\[SDK\]](#) [\[review\]](#) [\[benchmark\]](#)

Nokia 6600



Nokia 6620

Band	GSM 900/1800/1900	Heap size	3MB
Screen	176 x 208	Max JAR size	Dynamic
Color	65k	Memory size	6MB
Data	Unknown	Platform	Nokia Series60 Symbian 7.0s
Supported API			
DRM Forward Lock	CLDC 1.0	Bluetooth API	MIDP 2.0
Nokia S60 API	WMA 1.0		MMAPI 1.0

Link to [\[specification\]](#) [\[dev portal\]](#) [\[SDK\]](#) [\[review\]](#) [\[benchmark\]](#)



Nokia 7610

Band	GSM 850/1800/1900	Heap size	Unknown
Screen	176 x 208	Max JAR size	Unknown
Color	65k	Memory size	12MB
Data	EDGE	Platform	Nokia Series60 Symbian 7.0s
Supported API			
CLDC 1.0	DRM Forward Lock	Bluetooth API	MIDP 2.0
Nokia S60 API	WMA 1.0		MMAPI 1.0

Link to [\[specification\]](#) [\[dev portal\]](#) [\[SDK\]](#) [\[review\]](#) [\[benchmark\]](#)



Nokia 6630

Band	GSM 850 or 900/1800/1900	Heap size	Dynamic
Screen	176 x 208	Max JAR size	Dynamic
Color	65k	Memory size	8Mb
Data	GPRS	Platform	Nokia Series60 Symbian 7.0s
Supported API			
CLDC 1.0	DRM Forward Lock	Bluetooth API	MIDP 2.0
Nokia S60 API	WMA 1.0		MMAPI 1.0

Link to [\[specification\]](#) [\[dev portal\]](#) [\[SDK\]](#) [\[review\]](#) [\[benchmark\]](#)



Nokia 6260

Band	WCDMA 2000 / GSM 900/1800/1900	Heap size	Dynamic
Screen	176 x 208	Max JAR size	Dynamic
Color	65k	Memory size	10Mb
Data	3G Data	Platform	Nokia Series60 Symbian 8.0
Supported API			
Nokia S60 API	Mobile 3D API	MMAPI 1.0	MIDP 2.0
Bluetooth API	File Connection API	CLDC 1.1	DRM 1.0
WMA 1.0			JTWI 1.0
			PIM Access API

Link to [\[specification\]](#) [\[dev portal\]](#) [\[SDK\]](#) [\[review\]](#) [\[benchmark\]](#)



Nokia 6670

Band	GSM 850 or 900/1800/1900	Heap size	Dynamic
Screen	176 x 208	Max JAR size	Dynamic
Color	65k	Memory size	4.6MB
Data	GPRS	Platform	Nokia Series60 Symbian 7.0s
Supported API			
CLDC 1.0	DRM Forward Lock	Bluetooth API	MIDP 2.0
Nokia S60 API	WMA 1.0		MMAPI 1.0

Link to [\[specification\]](#) [\[dev portal\]](#) [\[SDK\]](#) [\[review\]](#) [\[benchmark\]](#)



Nokia 3230

Band	GSM 850 or 900/1800/1900	Heap size	Dynamic
Screen	176 x 208	Max JAR size	Dynamic
Color	65k	Memory size	8MB
Data	EDGE	Platform	Nokia Series60 Symbian 7.0s
Supported API			
CLDC 1.0	DRM Forward Lock	Bluetooth API	MIDP 2.0
WMA 1.0			MMAPI 1.0

Link to [\[specification\]](#) [\[dev portal\]](#) [\[SDK\]](#) [\[review\]](#) [\[benchmark\]](#)



Band	GSM 900/1800/1900	Heap size	Unknown
Screen	176 x 208	Max JAR size	Unknown
Color	262k	Memory size	Unknown
Data	EDGE	Platform	Nokia Series60 Symbian 7.0s

Supported API

CLDC 1.1	Bluetooth API	MIDP 2.0	MMAPI 1.0	Nokia S60 API
DRM 1.0	WMA 1.0			

Link to [\[specification\]](#) [\[dev portal\]](#) [\[SDK\]](#) [\[review\]](#) [\[benchmark\]](#)

Nokia 6230



Band	GSM 900/1800/1900	Heap size	512Kb
Screen	128 x 128	Max JAR size	128Kb
Color	65k	Memory size	3.5MB
Data	EDGE	Platform	Nokia Series40

Supported API

MIDP 2.0	MMAPI 1.0	CLDC 1.1	Bluetooth API	JTWI 1.0
Nokia S40 API	DRM 1.0	WMA 1.0		

Link to [\[specification\]](#) [\[dev portal\]](#) [\[SDK\]](#) [\[review\]](#) [\[benchmark\]](#)

Nokia 6255



Band	CDMA	Heap size	1Mb
Screen	128 x 160	Max JAR size	400Kb
Color	65k	Memory size	Unknown
Data	Unknown	Platform	Nokia Series40

Supported API

CLDC 1.1	File Connection API	Bluetooth API	MIDP 2.0	MMAPI 1.0
Mobile 3D API	Nokia S40 API	DRM 1.0	PIM Access API	WMA 1.0
CDMA phone with either MIDP 2.0 or BREW 2.0 (carrier dependent)				

Link to [\[specification\]](#) [\[dev portal\]](#) [\[SDK\]](#) [\[review\]](#) [\[benchmark\]](#)

Nokia 6260



Band	GSM 850/1800/1900	Heap size	Unknown
Screen	176 x 208	Max JAR size	Unknown
Color	65k	Memory size	10Mb
Data	GPRS	Platform	Nokia Series60 Symbian 7.0s

Supported API

CLDC 1.0	Bluetooth API	MIDP 2.0	MMAPI 1.0	Nokia S60 API
WMA 1.0				

Link to [\[specification\]](#) [\[dev portal\]](#) [\[SDK\]](#) [\[review\]](#) [\[benchmark\]](#)

Sony Ericsson P900



Band	GSM 900/1800/1900	Heap size	Unknown
Screen	208 x 320	Max JAR size	Unknown
Color	65k	Memory size	Unknown
Data	Unknown	Platform	UIQ 2.1 Symbian 7.0

Supported API

CLDC 1.0	Bluetooth API	MIDP 2.0	MMAPI 1.0	WMA 1.0
----------	---------------	----------	-----------	---------

Link to [\[specification\]](#) [\[dev portal\]](#) [\[SDK\]](#) [\[review\]](#) [\[benchmark\]](#)

BenQ P30



Band	GSM 900/1800/1900	Heap size	Unknown
Screen	208 x 320	Max JAR size	Unknown
Color	262k	Memory size	Unknown
Data	GPRS Class 10 (48Kbps)	Platform	UIQ 2.1 Symbian 7.0

Supported API

CLDC 1.0	Bluetooth API	MIDP 2.0	MMAPI 1.0	WMA 1.0
----------	---------------	----------	-----------	---------

Link to [\[specification\]](#) [\[dev portal\]](#) [\[SDK\]](#) [\[review\]](#) [\[benchmark\]](#)

bron : <http://www.amsterdam.sebos.amsterdam.nl/amsterdam/sebos/plattegrond/plattegrond2004.pdf>

NIEUWE MEER
(peil 2,02 m - N.A.P.)

Polder Meerzicht

The map shows the Nieuwe Meer area in Amsterdam, including the Polder Meerzicht and Oeverlanden Nieuwe Meer. It features a network of waterways, including the Oostelijke Canal, and various infrastructure elements like roads, bridges, and parking areas. Key locations marked include Oostelijke Canal, Oeverlanden Nieuwe Meer, Polder Meerzicht, and various waterways like the Oostelijke Canal and the Oostelijke Canal. The map also shows the location of the Nieuwe Meer and the Polder Meerzicht. The map is oriented with North at the top.

bron : <http://www.amsterdamsebos.amsterdam.nl/amsterdamsebos/plattegrond/plattegrond2004.pdf>

BIJLAGE 4. JAVA SUPPORT ON POCKET PC

(bron : <http://www.comp.lancs.ac.uk/computing/users/fittond/ppcjava.html>)



Java Support on Pocket PC



Name & Version	NSIcom CrEme 3.2.2	SavaJe OS 2.0	Sun Personal Java	Blackdown J2RE (ARM port)	Esemertec Jeode
Supported OS	CE.net, PocketPC/2002	-	WinCE 2.11	Familiar Linux	WinCE 2.11, PocketPC, Linux
JVM Compatibility	PersonalJava compliant	J2ME MIDP 2.0	PersonalJava compliant	Java 2 (1.3.1)	PersonalJava compliant
Speed	Fast	Very fast	Fast	Slow start, reasonable execution.	Fast
Additional Supported Packages	SWING COMM, CORBA, SQL, JSSE	SWING, JINI, COMM	-	All available under Linux	-
Supported Hardware	PocketPC/ARM, Palm, HPC	PocketPC/ARM	MIPS, SH3	iPAQ H36xx (or better)	Dell Axim X5
Cost	Contact NSIcom	Contact SavaJe	Free - Sun EOL	Free	\$49.95 from Handango

Name & Version	Ewe VM	SuperWaba VM	Kada	Kaffe
Supported OS	Linux, PocketPC/2002, Windows, Palm, HPC Pro	PocketPC, Palm	WinCE 2.11/3.0	Familiar Linux
JVM Compatibility	PersonalJava compliant, Java 1.1, Java 1.2	Check web site	J2ME MIDP 2.0	Kaffe
Speed	Medium-Fast	Medium	Fast (Ahead of time compile)	-
Additional Supported Packages	Waba, ActiveSync	Waba	J2ME, PersonalJava	Kaffe (AWT, J2SE)
Supported Hardware	PocketPC/ARM, Sharp Zaurus, Palm, HPC, Linux, Win32	PocketPC/ARM, Palm	PocketPC/ARM	PocketPC/ARM
Cost	Free	Free	Contact Kada	Free

Name & Version	Hewlett Packard ChaiVM	iCompiler	IBM WebSphere Studio
Supported OS	WinCE 2.11, PocketPC	PocketPC	WinCE 2.11, PocketPC
JVM Compatibility	PersonalJava compliant	PersonalJava compliant	J2ME MIDP 1.0
Speed	Slow	-	Fast (Ahead of time compile)
Additional Supported Packages	-	-	-
Supported Hardware	PocketPC/ARM	PocketPC/ARM	PocketPC/ARM, WinCE, PalmOS, Windows, Wind River..
Cost	Free	\$19.95	Web site

(Note: '-' in a cell indicates this information is unknown or unavailable, please contact us if you can help)

SavaJe XE

Description: Fully featured Native Java OS

Notes: PocketPC port was recently sold to recently sold to [MochaWorks](#) who have not released anything yet. Early versions demonstrated a Java OS with neat UI, MP3 player, organizer etc with support for PC card 802.11, CF & PC card Ethernet and MODEM.

URL: <http://www.mochaworks.com>

Sun PersonalJava

Description: Sun's own Java runtime environment for handhelds.

Notes: This product has been discontinued, though a CE 2.11 version is available from Sun's EOL(End Of Life) site. Fairly fast and well implemented, AWT components look same as those on desktop machines, a useful console window is provided for debugging etc, is also very well documented and supported. Only problem is that it does not manage memory or threads very well, command line options can be used to help slightly (by assigning more memory), though many simultaneous threads and large data structures always cause a crash.

URL: <http://java.sun.com/products/personaljava/index.html>

Blackdown J2RE

Description: Blackdown's Java 2 JDK for Linux ported to ARM

Notes: Requires Linux on iPAQ

URL: <http://www.blackdown.org/>

Jeode

Description: Well featured PocketPC VM

Notes: Comes free with some Compaq iPAQs (this version is limited to iPAQ devices only). Also available for Dell PDA's from Handango. The only other way I found to purchase Jeode is to join their educational institution program, which seems very expensive.

URL - <http://www.esmertec.com>

NSICOM CrEme

Description: Quick, good support, very polished - an excellent product

Notes: - Handles memory far better than Sun's Personal Java, also faster. Supports Truffle and Tiny AWT AWT components which all work perfectly, it is provided with a useful launcher application. Provides various useful command line switches (such as to remove WinCE task bar), comm support fully implemented & works perfectly (also tested with iButtons). Overall a very solid VM implementation.

URL: <http://www.nsicom.com/>

IBM WebSphere Studio

Description: VM is provided as part of Java development suite for embedded devices.

Notes: Commercial product, though Linux evaluation version are available for download

URL: <http://www.ibm.com/websphere>

Ewe VM

Description: An open source, free JVM with Active Sync toolkit. It is based on the popular Waba API www.wabasoft.com but extended to embrace the PocketPC.

Notes: Runs fast, stable, and the developer is very responsive to support questions.

URL: <http://www.ewesoft.com>

SuperWaba VM

Description: Another free JVM based on the Waba API, with support for both PalmOS and PocketPC with almost 100% source compatibility between the two platforms.

Notes: A lot of good GUI tools and libraries to extend the API to the platform.

URL: <http://www.superwaba.com.br>

Kaffe

Description: Clean room unlicensed (but legal) Java VM implementation.

Notes: Implementation is not complete and contains bugs.

URL: <http://www.kaffe.org>

jCompiler

Description: This is a Java IDE/compiler which runs on the PocketPC!

Notes: Doesn't seem to be an official web site, contact jCompiler@hotmail.com for more info.

URL: [\[Handango Link\]](#)

Kada

Description: A commercial JVM for Palm and Pocket PC. It supports the J2ME programming model and Kada is the first to offer a MIDP 2.0 release.

Notes: They offer an ahead of time (AOT) compiler to improve efficiency.

URL: <http://www.kadasystems.com>

Hewlett Packard ChaiVM

Description: HP's own clean room JVM implementation. Provided with its Jornada range of devices. No recent activity/releases.

Notes: Slow and slightly buggy implementation. Provided native compiler.

URL: <http://www.hp.com>

Another site you might find useful is Vik David's page [Java on PocketPC, the Unofficial FAQ](#)

This page was last uPDAted on 20/6/03 by Dan Fitton (df (at) comp.lancs.ac.uk) with much help from Rod Montrose (rcm (at) avidwireless.com)

BIJLAGE 5. GPS EN GALILEO

Bron : <http://www.tweakers.net/>

16 december 2004

Bush wil GPS en Galileo kunnen uitschakelen

Dat het Amerikaanse leger de nauwkeurigheid van GPS kan beïnvloeden was reeds bekend. President Bush heeft nu echter ook een wetsvoorstel gedaan voor bevoegdheid om GPS en het Europese Galileo-systeem in noodsituaties geheel uit te kunnen schakelen. Eerder was er sprake van onderhandelingen over het beperken van de toegang tot Galileo, maar die kunnen met dit voorstel onder druk worden gezet. Hoe het Amerikaanse leger het Europese systeem zou willen uitschakelen en of dat in overleg met de EU gebeurt is onbekend, omdat een gedeelte van het document waarin het voorstel is uitgewerkt nog geheim is.

De maatregel is volgens een woordvoerder van het Witte Huis vooral bedoeld voor de veiligheid van de burgers. Omdat GPS ook wordt gebruikt voor het richten van raketten en bombardementen, zouden terroristen er mogelijk controle over willen krijgen. In dat geval zou het systeem bijvoorbeeld uitgeschakeld kunnen worden. Experts zijn het er over eens dat een dergelijke stap niet te gemakkelijk moet worden genomen, mede door de afhankelijkheid van lucht- en scheepvaart van het GPS-systeem:

"This is not something you would do lightly," said James A. Lewis, director of technology policy for the Washington-based Center for Strategic & International Studies. "It's clearly a big deal. You have to give them credit for being so open about what they're going to do."

30 november 2003

Europees navigatiesysteem Galileo onder controle VS

Heise bericht dat het Europese satellietnavigatiesysteem Galileo, bedoeld om Europa onafhankelijk te maken van het Amerikaanse GPS-netwerk, waarschijnlijk toch beïnvloedbaar wordt door de Verenigde Staten. Het Amerikaanse leger zou ongevraagd het signaal van Galileo kunnen verminderen of zelfs door storingen onbruikbaar maken, zoals dat nu ook kan met GPS-signalen ten tijde van oorlog of politieke onrust. Daarnaast wil de Amerikaanse overheid ook de mogelijkheid hebben om het normale ongeëncrypteerde Galileo-signaal, dat tijdens normaal gebruik wordt uitgezonden, te kunnen verzwakken. Als de VS haar zin zou krijgen zou dit betekenen dat een ander groot voordeel van Galileo, de grotere nauwkeurigheid ten opzichte van bestaande systemen, zou wegvallen. De definitieve onderhandelingen hierover zullen in januari plaatsvinden in Washington:

Die entscheidende Verhandlungsrunde darüber steht im Januar in der US-Bundeshauptstadt Washington an. Rund vier Milliarden Euro soll das Galileo-Projekt kosten und im Jahr 2008 an den Start gehen. Rund 30 Satelliten werden für das System benötigt. Zuletzt hatten sich China und Indien bereit erklärt, sich an der Finanzierung des Projekts mit zusammen 500 Millionen Euro zu beteiligen.

In deze bijlage is een kopie opgenomen van de test die de consumentenbond heeft gedaan voor PDA's. De test dateert uit september 2004 en geeft de prijzen alsmede de verschillende communicatiemiddelen de diverse PDA's beschikken.

PDA's

Markt & type	Prisje *	Testscore in %	Modelle- bedien	Synchro- nizatie	Energie- gebruik	Scher- m	Gebruiks- gemak	Formaat	Gewicht	Opslagcapaciteit	Batterij- levensduur	Aan- sluiting	Software	Commu- nicatie
Testscore in %														
ASUS MyPal A716	385	69	++	+	+	+	+	13,5 x 7,8 x 1,8	224	55,8	CM	P/S	HMD	BOSW
Toshiba e505	510	67	++	+	+	+	+	13,5 x 8,8 x 1,7	198	126,8	CM	P/S	HMD	BOSW
Sony Clie PEG-T655	510	65	+	+	++	+	+	12,2 x 7,3 x 1,3	184	29,4	S	O/S	HMD	ISW
HP iPAQ h4355	540	65	+	+	+	+	+	11,9 x 7,5 x 1,4	167	56,8	M	P/S	HMD	ISW
ASUS MyPal A620BT	270	63	+	+	+	+	+	12,5 x 7,2 x 1,4	147	55,3	C	P/S	HMD	BOSW
Toshiba e405	375	63	+	+	+	+	+	12,5 x 7,6 x 1	130	62,3	M	P/S	HMD	BOSW
Sony Clie PEG-N673WU	425	63	□	□	□	+	+	13,2 x 7,2 x 2,2	232	11,0	CS	O/S	HMD	IS
Sony Clie PEG-N680W	425	63	□	□	□	+	+	13,2 x 7,2 x 2,2	238	16,0	CS	O/S	HMD	IS
HP iPAQ h4155	420	61	+	+	+	+	+	11,4 x 7,1 x 1,4	133	56,8	M	P/S	HMD	BOSW
ViewSonic PocketPC V36	58		+	+	+	+	+	12,7 x 7,7 x 1,6	147	51,1	M	P/S	HMD	IS
Palm Tungsten T2	380	57	□	+	□	+	+	10,2 x 7,5 x 1,5	159	29,2	M	O/S	HMD	BOSW
Sony Clie PEG-T137	320	56	□	+	□	+	+	12,7 x 7,7 x 1,6	147	51,1	M	P/S	HMD	IS
Palm Tungsten E	210	55	+	+	□	+	+	11,4 x 7,5 x 1,4	165	23,0	S	O/S	HMD	BOSW
Garmin iQue	810	54	□	□	-	+	+	11,5 x 7,6 x 1,2	130	28,8	M	O/S	HMD	IS

* 0-20 = slecht, 20-40 = matig, 40-60 = redelijk, 60-80 = goed, 80-100 = zeer goed

++ = zeer goed
+ = goed
□ = redelijk
- = matig
- = slecht
▲ = beste koop

3. Gangbare pc's op internet mei 2004, list ingevuld: nog niet leverbaar in Nederland

2. C = Compact Flash
M = MultiMedia Card/Secure Digital Card
S = Memory Stick
3. D = Palm OS
P = Pocket PC
S = penning (stylus)
T = toetsenbord
H = hoofdtelefoon
M = microfoon

5. Alle: Microsoft, MP3-speler, MS Office
B = beeldscherm of filmspeler
C = backup mogelijkheid
D = dicteermogelijkheid
E = eBook lezer
S = spreadsheet
T = tekstverwerker
U = uitgaansbecher
W = webbrowser

6. B = Blue Tooth
I = infrarood
K = kabel
W = Wi-Fi
7. C = camera
D = beeldscherm
G = GPS
Global positioning system

IMPORTeurs
Auss: (0591) 666 835, <http://nl.auss.com>
Compag: (0182) 50 58 07, www.compaq.nl
Dell: (020) 674 45 00, www.dell.nl
Garcin: (0302) 540 830, www.garcin.nl
HP: (020) 347 71 35, www.hp.nl
Palm: (076) 524 45 33, www.palm.com/nl
Sharp: (020) 635 95 00, www.sharp.nl
Sony: (020) 658 18 88, www.sony.nl
Tooth: (010) 20 23 00, www.tooth.nl