“Parels der Informatica”

Ontwerp Module 1.1 van de studie INF

— Werkdocument —

Pieter-Tjerk de Boer
Pascal van Eck
Jan Kamphuis
Maurice van Keulen
Arend Rensink

23 april 2013

Inhoudsopgave

1 Inleiding **5**
1.1 Ontwerpcriteria . . . 5
1.2 Wiskundelijn . . . 5

2 Globale opzet **6**
2.1 Het parel-concept . . . 6
2.2 Project . . . 7
2.3 Overige leerlijnen . . . 7
2.4 Toetsing . . . 7
2.4.1 Herkansingsmogelijkheden . . . 8
2.5 Organisatie uitwerking ontwerp . . . 8
2.6 Organisatie tijdens module . . . 8

3 Detailopzet en keuzes **9**
3.1 Werkvormen . . . 9
3.2 Project . . . 9
3.2.1 Leerdoel . . . 9
3.2.2 Uitgangspunten voor het ontwerp . . . 9
3.2.3 Opdrachtsomschrijving . . . 10
3.2.4 Verroostering en embedding in de parels . . . 10
3.2.5 Begeleiding . . . 10
3.2.6 Achtergrond: Wat is twitter data mining? . . . 11
3.3 Materiaal . . . 11
3.3.1 Programmeertalen . . . 11
3.4 Academische vaardigheden . . . 12
3.4.1 Leerdoelen . . . 12
3.5 Resources . . . 13
3.5.1 Groepsgroottes en mankracht . . . 13
3.5.2 Zalen . . . 13
3.6 Vergelijking met ACM curriculum . . . 13

4 Weekindeling **15**
4.1 Week 1: Pink Pearl — The Intestines of your Computer . . . 15
4.1.1 Inleiding . . . 15
4.1.2 Leerdoelen . . . 15
4.1.3 Indeling . . . 15
4.1.4 Opdracht . . . 16
4.1.5 Docenten . . . 16
4.1.6 Studiemateriaal . . . 16
4.1.7 Academische Vaardigheden . . . 16
4.2 Week 2: Black Pearl — Algorithms for Sorting and Searching . . . 17
4.2.1 Wiskunde . . . 17
4.2.2 Academische Vaardigheden . . . 17
4.2.3 De parel — Inleiding . . . 17
4.2.4 Leerdoelen . . . 18
4.2.5 Indeling . . . 18
4.2.6 Opdracht . . . 19
4.2.7 Docenten . . . 19
4.2.8 Studiemateriaal . . . 19
4.3 Week 3: Green Pearl — Cryptography . . . 20
4.3.1 Inleiding . . . 20
4.3.2 Leerdoelen . . . 20
4.3.3 Indeling . . . 20
4.3.4 Opdracht . . . 21

4.3.5 Docent . . . 21
4.3.6 Studiemateriaal . . . 21
4.3.7 Academische Vaardigheden . . . 21
4.4 Week 4: White Pearl — Software Engineering . . . 22
4.4.1 Inleiding . . . 22
4.4.2 Leerdoelen . . . 22
4.4.3 Indeling . . . 22
4.4.4 Opdracht . . . 23
4.4.5 Docenten . . . 24
4.4.6 Materiaal . . . 24
4.4.7 Academische Vaardigheden . . . 24
4.5 Week 5: Red Pearl — Operating Systems and the Internet . . . 25
4.5.1 Inleiding . . . 25
4.5.2 Leerdoelen . . . 25
4.5.3 Indeling . . . 25
4.5.4 Opdracht . . . 25
4.5.5 Docenten . . . 26
4.5.6 Studiemateriaal . . . 26
4.5.7 Academische Vaardigheden . . . 26
4.6 Week 6: Blue Pearl — Functional Programming . . . 27
4.6.1 Wiskunde . . . 27
4.6.2 Academische Vaardigheden . . . 27
4.6.3 Programmeertaal . . . 27
4.6.4 Leerdoelen . . . 27
4.6.5 Docent . . . 28
4.6.6 Indeling . . . 28
4.6.7 Opdracht . . . 28
4.7 Week 7: Yellow Pearl — Visual and Intelligent . . . 29
4.7.1 Inleiding . . . 29
4.7.2 Leerdoelen . . . 29
4.7.3 Indeling . . . 29
4.7.4 Opdracht . . . 30
4.7.5 Docenten . . . 30
4.7.6 Studiemateriaal . . . 30
4.7.7 Academische Vaardigheden . . . 30
4.8 Week 8 . . . 31
4.8.1 Opzet van deze week . . . 31
4.8.2 Weekindeling . . . 31
4.8.3 Toetsing en beoordeling . . . 31
4.8.4 Academische Vaardigheden . . . 31
4.9 Week 9 . . . 32
4.9.1 Opzet van deze week . . . 32
4.9.2 Weekindeling . . . 32
4.9.3 Toetsing en beoordeling . . . 33
4.9.4 Academische Vaardigheden . . . 33
4.10 Week 10 . . . 34
4.10.1 Opzet van deze week . . . 34
4.10.2 Weekindeling . . . 34
4.10.3 Toetsing en beoordeling . . . 34
4.10.4 Academische Vaardigheden . . . 34
4.11 Week 11 . . . 35
4.11.1 Inleiding . . . 35
4.11.2 Leerdoelen . . . 35
4.11.3 Indeling . . . 35
4.11.4 Opdracht . . . 35
4.11.5 Docenten . . . 36

4.11.6 Studiemateriaal 36
4.11.7 Academische Vaardigheden 36

A Open Issues **37**

B Actiepunten **37**

1 Inleiding

Dit is het werkdocument voor het ontwerp van Module 1.1 in het beoogde nieuwe BSc-curriculum van INF. Het is een neerslag van de meest recente stand van de ideeën rond inhoud, werkvormen en toetsing van de module. Hieronder vatten we samen aan welke randvoorwaarden de module dient te voldoen, met andere woorden, wat de criteria voor het ontwerp zijn.

Noot voor de lezer. Niets in dit document is definitief! Het is niet meer of minder dan een werkdocument en is primair bedoeld voor het module-ontwerpteam zelf. Elke vorm van constructief commentaar door derden is echter welkom en zal indien mogelijk mee worden genomen in het ontwerpproces.

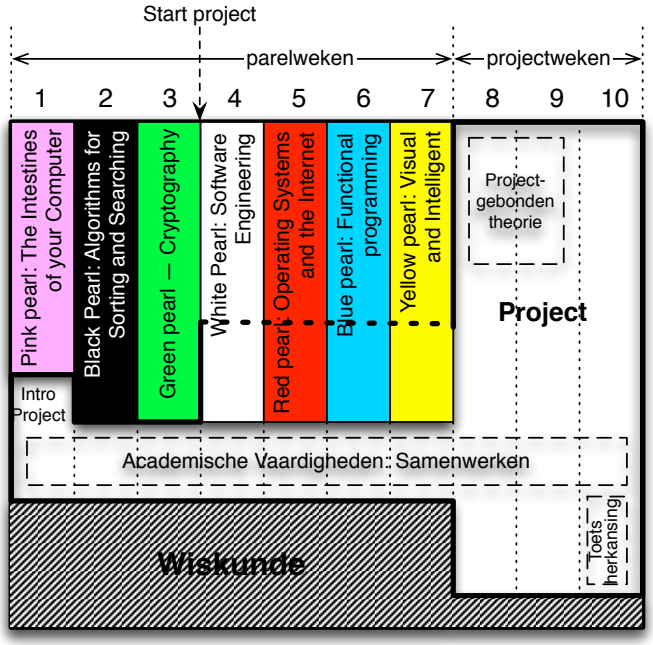
1.1 Ontwerpcriteria

Hoofddoelen van de module zijn studenten een indruk geven van de volledige breedte van de Informatica, zich trots te laten voelen om een informaticus te zijn/worden, selecteren op interesse en inspanning en een werkcultuur van nominaal studeren te creëren. Belangrijke randvoorwaarden en uitgangspunten:

1. De module wordt voor een deel gedeeld met BIT (≈3 weken, mogelijk ook project; zie §2.1), o.a. om BIT-studenten zoveel mogelijk dezelfde blootstelling aan programmeren te geven als INF-ers t.b.v. module 1.2.
2. De "cross-cutting concerns" (Academische vaardigheden, Software development, Security, Concurrency) komen zichtbaar aan bod.
3. Studeerbaar voor studenten met VWO Wiskunde-B; geen voorkennis programmeren.
4. De module levert voorkennis voor potentieel alle daaropvolgende INF-modules van het 1^e en 2^e jaar.
5. Activerende onderwijsvormen en aansluiten bij belevingswereld van de studenten.
6. Interessant voor studenten van andere studies als 'informatica-module'.
7. Itt andere modules zijn de leerdoelen niet vooraf gegeven en kunnen dus niet als ontwerpcriterium gezien worden.
8. Project heeft thema "Sport".

1.2 Wiskundelijn

Op UT-niveau is vastgelegd dat elke eerstejaarsmodule naast het opleidingsspecifieke gedeelte ook een component wiskunde inhoudt, in het vervolg de *wiskundelijn* genoemd. Voor Module 1.1 is de omvang van de wiskundecomponent 4 EC, oftewel ruim een kwart van de totaal beschikbare tijd.



Figuur 1: Globaal overzicht over de module

2 Globale opzet

Het basisidee is een module met 7 "*parelweken*" en een project.

2.1 Het parel-concept

Een parel concentreert zich op een verworvenheid van de informatica. Een opdracht staat centraal in die week en de daarvoor benodigde basiskennis en vaardigheden worden in diezelfde week onderwezen. Elke parel wordt getoetst dmv deze opdracht en een schriftelijke (deel)toets. Voor de duidelijkheid wordt de term *opdracht* consequent gebruikt voor de parelopdracht en de term *project* voor het eindproject.

De volgende parels zijn voorgesteld en de afgestemd met BIT:

1. *Pink pearl — The Intestines of your Computer*
   Onderwerp: Computerarchitectuur. Opdracht rondom programmeren microcontroller.

2. *Black pearl — Algorithms for Sorting and Searching* [met BIT]
   Onderwerp: Programmeren. Opdracht rondom zoekmachines.

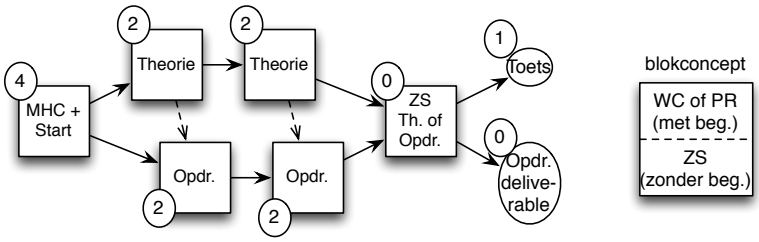
3. *Green pearl — Cryptography*
   Onderwerp: Cross-cutting concern 'Security': wiskunde achter cryptografie. Opdracht rondom en/decryptiesysteem programmeren

4. *White pearl — Software engineering* [met BIT]
   Onderwerp: Requirementsanalyse. Opdracht rondom requirementsanalyse voor het project.

5. *Red pearl — Operating Systems and the Internet*
   Onderwerp: Operating systems, computernetwerken. Opdracht nog niet bepaald.

6. *Blue pearl — Functional programming* [met BIT]
   Onderwerp: Een ander programmeerparadigma, namelijk functioneel programmeren. Opdracht nog niet bepaald.

7. *Yellow pearl — Visual and Intelligent*
   Onderwerp: Data-visualisatie, User-interfaces, kunstmatige intelligentie. Opdracht rondom user-interfaceontwerp voor het project.



Figuur 2: Concept voor onderwijs per parel in blokken elk doorgaans voor de helft onder begeleiding.

Hoewel het onderwijs enigszins zal variëren per parel, is het concept van Figuur 2 vastgesteld ivm studeerritme en verroostering. De kop, een motiverend hoorcollege, en de staart, een toets en een deadline voor de opdracht, zullen elke parelweek terugkomen. De overige 5 blokken worden parel-specifiek ingevuld met als uitgangspunt dat de helft van de uren onder begeleiding zijn.

2.2 Project

Het project behelst het realiseren van een Twitter-analysetool voor gebruik bij de Batavierenrace. Merk op dat wat er op Twitter gebeurt, gezien kan worden als puur "praten over hardlopen", maar dat het analyseren daarvan verschillende doelen en belanghebbenden kan dienen, bijvoorbeeld organisatorische knelpunten, publiekstevredenheid of calamiteitenondersteuning. Er wordt complexiteit en variëteit in het project gebracht door de projectgroepen één doel en belanghebbende te laten kiezen. De groepen krijgen een database met verrijkte twitterberichten en een configureerbare user-interface ter beschikking gesteld. Een aantal maar niet alle parels zijn in meer of mindere mate verbonden met het project. Tijdens het project is er nog projectgebonden onderwijs, bijvoorbeeld over databases.

2.3 Overige leerlijnen

Het wiskundeonderwijs concentreert zich synchroon met de parels in de eerste 7 weken. "Samenwerken" is het aandachtspunt vanuit Academische Vaardigheden voor deze module. Elke week heeft hiervoor een vast moment voor colleges, rollenspelen en projectgerichte activiteiten. Alle cross-cutting concerns komen zichtbaar aan bod:

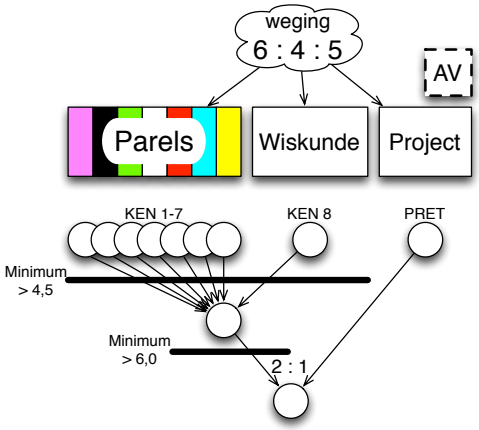
- Software Development als parel
- Security eveneens als parel
- Concurrency als bonusonderdelen in de opdrachten van enkele parels ter bewustmaking van het complexiteitsverhogende effect van deze kwestie.

AP 1: *Bij de opdrachten van specifieke parels (1,2,5?) concurrency-georiënteerde bonusonderdelen opnemen.* **MvK**

2.4 Toetsing

De beoordeling van de module bestaat uit drie componenten: de parels, de wiskunde en het project (zie Figuur 3). Academische vaardigheden weegt niet mee in het cijfer; meedoen is voldoende. De globale weging tussen de drie componenten is respectievelijk 6:4:5. Dit wordt als volgt in het KEN/PRET-systeem gerealiseerd.

Elke parel wordt beoordeeld met één cijfer: KEN1 t/m KEN7. Het parelcijfer wordt vastgesteld op basis van een 45min toets en een opdracht. De opdracht is voorwaardelijk, dwz moet voldoende worden gemaakt, maar kan ook bonuspunten opleveren. Het cijfer van de parel is dus het toetscijfer plus eventuele bonuspunten. Het wiskundecijfer (KEN8) wordt vastgesteld door de wiskundedocenten op basis van hun toetsen. Het projectcijfer (PRET) wordt



Figuur 3: Overzicht bepaling eindcijfer

vastgesteld door de projectbegeleider en is minimaal een 6 mits er geen sprake is van een wanprestatie qua inzet.

Voor alle KEN-cijfers geldt het standaardminimum van 4,5. Met een weging van 3 voor elk parelcijfer KEN1-7 en 14 voor KEN8 realiseren we de gewenste weging van 6:4 voor de parels tov. de wiskunde. Dit is het KEN-cijfer waarvoor een minimum van 6,0 geldt. Met een weging 2:1 voor KEN:PRET realiseren we uiteindelijk de gewenste weging 6:4:5 voor parels:wiskunde:project.

Deadline voor de parelopdracht is vrijdagnacht 23:59. De student kan ongevraagd uitstel nemen tot zondagnacht 23:59, maar verliest hiermee de mogelijkheid tot bonuspunten. Niet inleveren is onvoldoende (< 4.5) voor de parel.

2.4.1 Herkansingsmogelijkheden

Herkansingsmogelijkheden binnen de module:

- Vrijdagochtend wk9: pareltoetsen 1–4 (mogelijk 2 achter elkaar)
- Vrijdagochtend wk10: pareltoetsen 5–7 (mogelijk 2 achter elkaar)
- Max 2 pareltoetsen herkansbaar
- Bij calamiteiten kan de pareldocent ter herkansing van een opdracht een aanvulling aanbieden. Deadline is vrijdagnacht in wk9 (1–4) respectievelijk wk10 (5–7).
- Max 2 parelopdrachten te herkansen op deze manier.

2.5 Organisatie uitwerking ontwerp

Elke parel, het project en Academische Vaardigheden worden uitgewerkt door een ontwerpteam (detailontwerp van de sessies, leerdoelen, studiemateriaal, opdracht, organiseren benodigde docenten):

- Pink: Andre Kokkeler en Pieter-Tjerk de Boer
- Black: Arend Rensink
- Green: Andreas Peter (PostDoc DIES; begint per 1/3/13) en Maurice van Keulen
- White: Christiaan Katsma en Pascal van Eck
- Red: Pieter-Tjerk de Boer en nog iemand (wordt naar gezocht)
- Blue: Jan Kuper en Arend Rensink
- Yellow: Betsy van Dijke en Maurice van Keulen
- Project: Pascal van Eck, Maurice van Keulen en Mena Badieh Habib[1]
- Academische Vaardigheden: Anne Remke en Karen Slotman

2.6 Organisatie tijdens module

Tijdens het 'draaien' van de module zal er behoefte zijn aan overleg en afstemming vanwege het grote aantal betrokken docenten en studentassistenten. Daarnaast zal er behoefte zijn aan monitoring: hoe de voortgang van de studenten te kunnen volgen als ook het draaien zelf (knelpunten tijdig identificeren en snel kunnen oplossen). Ook het concept van een 'rapportvergadering' lijkt nodig om te beslissen over randgevallen (zowel slagen/zakken als ook eindcijfers), ziekte en andersoortige tijdelijke uitval, etc.

AP 2: *Ontwerp voor "Organisatie tijdens module" maken* **MvK**

[1] Mena is een AiO van de DB-groep die coördineert: het vergaren van alle Tweets van de Batavierenrace 2013 en het realiseren van de rudimentaire dashboard.

3 Detailopzet en keuzes

In het onderstaande gebruiken we de volgende afkortingen voor de drie onderdelen van de module:

A Academische vaardigheden;
W De wiskundelijn;
P Parels van de informatica;
E Het eindproject.

3.1 Werkvormen

Colstructie (col) Een mengvorm van hoorcollege en werkcollege, voor de complete groep.
Hoorcollege (hc) Een of twee uur durend college voor de complete groep, waarin de stof geheel of gedeeltelijk de revue passeert (traditioneel hoorcollege), ofwel motivatie en structurering van de stof aan de orde komt (motiverend hoorcollege).
Instructiepracticum (ipr) Een mengvorm van werkcollege en (strak geregisseerd) docentgestuurd practicum: studenten doen kleine opdrachten, theoretisch en praktisch, die afgetekend moeten worden. Dit wordt gedaan in groepen van ongeveer 25 personen.
Opdracht (opd) Praktische opdracht binnen een parel. Gedaan in groepen van variërende grootte per parel, in principe zelfgestuurd.
Peer feedback (pfb) In groepen van plm. 8, in principe zelfgestuurd.
Project (prj) Wordt alleen gebruikt voor activiteiten voor het eindproject waarbij studenten in groepen van 6 werken in een niet centraal gereserveerde ruimte (bijv. in projectkamers of het Educafé), eventueel onder begeleiding van een vaste coach.
Vragenuur (qa) Gelegenheid voor studenten om vragen te stellen, voor docenten om nader toe te lichten. qais voor de complete groep gezamenlijk.
Werkcollege (wc) Sessie in groepen van plm. 25 personen, waarin opgaven door de studenten individueel worden doorgewerkt met actieve aanwijzingen van de docent; af en toe wordt iets klassikaal uitgelegd.
Toets (tts) Diagnostische toets of eindtoets; voor de complete groep.
Zelfstudie (zs) Vindt thuis plaats, of (in sommige gevallen) onder begeleiding in een collegezaal.

AP 3: *Streven is om het "interview een wetenschapper" uit AVI1 te behouden, mogelijk ook "interview met een alumnus", maar waar geven we dat een geschikte plek in de module?* **PTdB**

3.2 Project

3.2.1 Leerdoel

Naast het verdiepen van enkele parelleerdoelen, heeft het eindproject ook haar eigen leerdoelen:

- een informatievraag van een eindgebruiker operationaliseren met een ontwerp voor data-analyse en -visualisatie en deze realiseren als een sociaal intelligent systeem.
- kennis en vaardigheden in samenhang en in een team toepassen in een integraal project dat aspecten uit de praktijk bevat.

3.2.2 Uitgangspunten voor het ontwerp

- Project moet motiverend zijn: studenten moeten aan het einde het idee hebben dat ze wat bereikt hebben, dat ze kennelijk iets nuttigs geleerd hebben in deze module;
- Er moet iets opgeleverd worden dat waarde heeft voor de denkbeeldige klant, het moet werken en af zijn. Het alternatief, iets superambitieus opstarten, niet afmaken en dan een voldoende krijgen omdat het ontwerpverslag zo'n goede structuur heeft, is niet acceptabel.
- Project moet redelijk voorgestructureerd zijn: het is niet goed als we de studenten in het diepe gooien en op de maandag van week 1 zeggen: vrijdag over drie weken is de deadline, hoe je de tussenliggende tijd indeelt moet je zelf weten, succes! Inzet van tutoren is niet voldoende; wij moeten van tevoren structuur aanbrengen.

- Tijdens het project moet databasetheorie aan bod komen (want: zit niet in de parels, maar verdient het wel om als verworvenheid van de informatica).
- Project moet beoordeeld kunnen worden, zowel groepsprestatie als individueel.
- We maken er geen rollenspel van: studenten zijn niet zogenaamd een softwarehuis dat in opdracht van de Bata een twitter mining tool aan het maken is.
- Er is geen competitie-element: we kiezen niet welke uitwerking het beste is en wijzen geen winnaar aan.
- We gaan er niet vanuit dat we tijdens het project toegang hebben tot een daadwerkelijke klant die de studenten zouden kunnen benaderen om erachter te komen wat die klant precies wil. De studenten zullen zich zelf moeten inleven in de denkbeeldige klant of zelf een representant vinden.

3.2.3 Opdrachtsomschrijving

De taak voor de studenten is om voor de Batavierenrace-organisatie een tool te maken die een bepaalde groep belanghebbenden bij de Bata inzicht verschaft in wat er over de Bata gezegd wordt op Twitter. Belanghebbenden kunnen zijn: de organisatie van de Bata, politie, de pers, de toeschouwers, en natuurlijk: hardlopers zelf. Studenten kiezen zelf op welke groep belanghebbenden en welke informatievraag zij zich richten; de uitwerking van het project is dus alleen al om deze reden niet voor alle groepen hetzelfde. Voorbeelden van informatievragen zijn: "Waar is het nu gezellig?" (toeschouwers), "Is er iemand gevallen, en zo ja, waar?" (EHBOers), "Vond men 't een leuke en spannende race?" (pers).

De studenten krijgen een dashboard met een configureerbare user-interface ter beschikking (met o.a. componenten zoals een word cloud, grafiek, zoek/filter form, lijst met voorbeeld-tweets), een rule-engine met geïntegreerde classifier (waarmee de studenten in week 7 leren werken) en een database met de tweets van de Batavierenrace 2013 (verrijkt met resultaten van sentiment analysis e.d.). De studenten worden geacht hiermee een regelgebaseerd sociaal intelligent systeem te realiseren dat de informatievraag van de gekozen belanghebbende kan beantwoorden. Dit komt neer op het programmeren van een verzameling event-condition-action regels die de twitter data analyseren en de dashboard aansturen. Voor bovengemiddelde studenten en studenten met meer programmeerervaring is het mogelijk om deze opzet (mogelijk met hulp van de studentassistenten) uit te breiden met nieuwe event- en action-types en user-interfacecomponenten.

Merk op dat het niet mogelijk is om buiten de scope de gegeven rule engine en dashboard te treden en dat iets superambitieus opstarten en dat niet afmaken, niet acceptabel is; men moet iets opleveren dat af is, werkt, en waarde heeft voor de denkbeeldige klant. Bovendien, het gaat in het eindproject om de technische realisatie van zo'n systeem en niet om de implementatie (in de zin van inbedden in de organisatie en hun werkwijze). Pure technology push dus.

3.2.4 Verroostering en embedding in de parels

Het project wordt gelijk in week 1 geïntroduceerd, maar start pas officieel in week 4[2] waarin de requirementsspecificatie voor het te realiseren systeem worden opgesteld en waarin requirements engineering, de ontwerpcyclus en 'formeel' projectmanagement wordt onderwezen. De parels in weken 5 en 6 zijn niet hecht met het project verweven, maar studenten kunnen er even goed aan doorwerken als ze daar ruimte voor vinden.

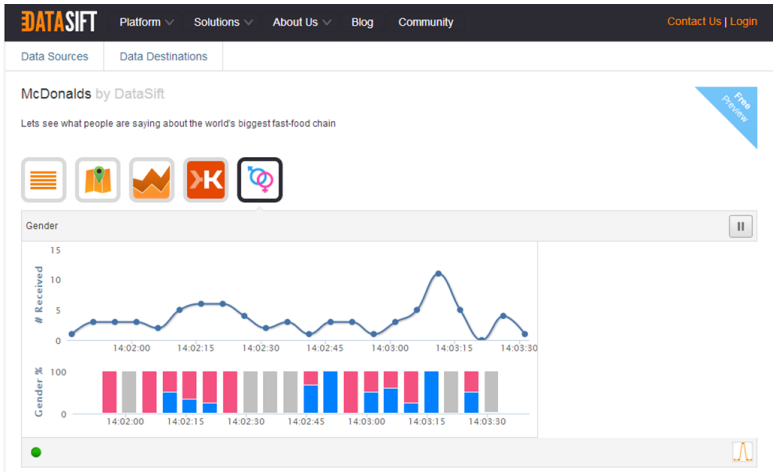
In week 7 wordt een stakeholder- en contextanalyse gedaan voor het project en worden de beginselen van kunstmatige intelligentie worden onderwezen. Tevens starten de studenten met het programmeren van het regelgebaseerde sociaal intelligente systeem.

De weken 8, 9 en 10 staan primair in het teken van het eindproject. Deze weken zijn voorgestructureerd met (niet getentamineerd) projectgebonden onderwijs, oplevermomenten voor (deel)resultaten en presentaties. In week 8 staat ontwerpen centraal en in week 9 het uitprogrammeren. Week 9 heeft op donderdag een verplicht inlevermoment voor het gehele systeem waarvoor men op vrijdag een officieuze voorlopige beoordeling ontvangt. Week 10 is minder intensief en geeft ruimte voor een extra iteratie. Het eindproject en daarmee de hele module wordt afgesloten met een laatste ronde met presentaties op vrijdag.

3.2.5 Begeleiding

Er is een vast team van studentassistenten verbonden met deze module. Deze studentassistenten zijn het eerste aanspreekpunt voor de studenten voor technisch-inhoudelijke vragen gedurende de parelweken alsmede de projectweken. Daarnaast krijgt elke groep een docentbegeleider toegewezen die verantwoordelijk is voor zowel

[2] Dit hoeft de studenten er niet van te weerhouden al eerder een groep te vormen en al voorbereidende activiteiten te ontplooien.



Figuur 4: Gender of people twittering about McDonald's over time.

procesbegeleiding als expertadvies. In de academische vaardigheden-lijn wordt specifiek aandacht besteed aan het functioneren van het team.

3.2.6 Achtergrond: Wat is twitter data mining?

Twitter data mining, of meer algemeen: social media data mining gaat over inzicht krijgen in wat (potentiële) klanten over jou zeggen en het daaruit filteren van voorkeuren, wensen en percepties. Traditioneel doen organisaties dat m.b.v. marketingonderzoek in de vorm van enquetes en/of klantenpanels. Social media data mining is een alternatieve manier om hetzelfde doel te bereiken (zie ook deze blog. In de context van de Batavierenrace zou je het ook als volgt kunnen formuleren: Vóór en tijdens de Batavierenrace zijn deelnemers bezig met twee dingen: hardlopen en praten over hardlopen. De Batavierenrace heeft al de mogelijkheid om inzicht te krijgen in hoe dat hardlopen gaat (wat wil je: je bent een race, dan moet je kunnen meten wie het snelst is). Twitter data mining gaat over inzicht krijgen in die tweede activiteit: praten over hardlopen, specifiek over de bata.

De batavierenrace is niet de enige die geïnteresseerd is in social media data rondom sport. Zie het filmpje (Bloomberg, 3:11m) waarin social media mining door de New Jersey Devils (een professioneel ijshockey team) centraal staat. Ook de CEO van een belangrijke tool vendor (Crimson Hexagon) komt aan het woord.

Social media data mining vormt een levendige markt waarop al een aantal aanbieders actief zijn: Klout.com, trackur.com, Crimson Hexagon. De eerste twee bieden een kant en klare, niet al te dure online service. Crimson Hexagon lijkt zich meer te richten op totaaloplossingen (incl. consultancy, lokaal installeren van software, etc.). Datasift levert geen kant en klaar product, maar een platform, waarmee je dan 'dashboards' kunt maken zoals dit voorbeeld gebaseerd op McDonald's (of het werkt hangt af van het tijdstip van de dag: er moeten natuurlijk wel tweets zijn om iets te laten zien), zie ook Figuur 4. Nog een voorbeeld: Twitcident: Twitter mining om snel plaats, omstandigheden en omvang van een ramp of ander grootschalig incident te bepalen.

3.3 Materiaal

De module wordt in het Nederlands aangeboden, met de studiehandleiding, hoorcolleges en presentaties (incl. presentatiesheets) in het Nederlands tenzij beschikbaarheid van personeel het noodzakelijk maakt om één of enkele hoorcolleges in het Engels te geven. Leesmateriaal buiten de studiehandleiding is zoveel mogelijk in het Engels. De module probeert zoveel mogelijk aan te sluiten bij standaarden in het curriculum (bijv. het boek "Skill Sheets" voor Academische Vaardigheden), en de kosten voor studenten zo laag mogelijk te houden.

3.3.1 Programmeertalen

Er zijn drie plekken in de module waar programmeren een grote rol speelt: (1) de zwarte parel: algoritmiek, (2) de blauwe parel: functioneel programmeren en (3) het project. Elk stelt significant andere eisen aan de te gebruiken programmeertaal, terwijl het, zeker voor studenten zonder programmeervoorkennis, ongewenst is om

Variant	Aantal talen	Zwarte parel	Blauwe parel	Project	Bijzonderheden
1	3	C	Puur FP	Python	Wel 3 programmeertalen
2	2	C	Puur FP	Puur FP	Pure functionele taal voor programmeren dashboard in project
× 3	2	C	Python	Python	Functioneel programmeren onderwijzen met ondersteuning daarvoor in Python
4	2	Java	Puur FP	Java	Dashboard in Java programmeren en in algoritmiekparel voor Java kiezen ipv C
→ 5	2	Python	Puur FP	Python	Dashboard in Python programmeren en in algoritmiekparel voor Python kiezen ipv C
× 6	2	C	Puur FP	C	Dashboard in C programmeren

Tabel 1: Varianten voor programmeertaalkeuze

3 programmeertalen te gebruiken. Voor een gedegen keuze, onderscheiden we de varianten in Tabel 1. De volgende overwegingen spelen een rol bij de varianten.

1. Sluiten we nog niet uit, maar 3 programmeertalen geeft wel een verkeerd signaal.
2. De vraag is of een pure functionele taal realistisch is als implementatietaal van een dashboard. Het gaat met name om (a) client/server programmeren ivm connectie met databaseserver, en (b) een GUI programmeren.
3. Sluiten we uit. Python is geen goede keuze voor de blauwe parel. Het gaat in die parel om de paradigmashift en dus is uit didactisch oogpunt een pure functionele taal gewenst en niet een taal die paradigma's combineert (zowel Python als Javascript zijn gemixte talen).
4. In de zwarte parel was oorspronkelijk voor C gekozen om wat voorkennis van C te creëren voor latere modules die C gebruiken (1.3 netwerken en 2.1 computerarchitectuur). Aan die wens komen we in deze variant niet tegemoet. Zowel algoritmiek onderwijzen als een dashboard programmeren kan prima in Java.
5. Idem, maar in deze variant stappen we over op Python ipv Java.
6. Sluiten we uit: C valt niet te beargumenteren als programmeertaal voor een dashboard.

Leidende variant is variant 5. In de race blijven 1, 2 en 4.

3.4 Academische vaardigheden

Hoofddocenten: Karen Slotman en Anne Remke.

Het onderwerp uit de Academische Vaardigheden waarop in deze module wordt geconcentreerd is *Samenwerken*. Achterliggende studiemateriaal zijn de betreffende hoofdstukken uit "Skill Sheets". Er worden 8 bijeenkomsten georganiseerd op een vast tijdstip elke week (dinsdag 6+7 uur). De bijeenkomsten zijn om-en-om plenair en in groepjes spelvormen. Zie §4 voor details over de individuele bijeenkomsten.

De beoordeling van de academische vaardigheden is voorwaardelijk: meedoen is voldoende; er wordt geen cijfer gegeven dat meeweegt in het eindcijfer. Wel zal er overleg zijn met de docentbegeleiders van de groepen die de *P*-cijfers bepalen.

3.4.1 Leerdoelen

- erkennen van het belang van samenwerken
- ability to apply the principles of project management
- giving and receiving feedback
- formations of groups (internally / externally induced)
- Understand the basics of effective brainstorming and meetings
- Understand decision-making techniques and how to effectively organise meetings
- Identify free-riding behaviour in others and yourself
- Understand that free-rider behaviour takes different forms throughout the various project stages; take effective action
- reflecteren op de afgelopen project weken, zijn de geleerde technieken toegepast?
- defineeren van persoonelijke leerdoelen voor de komende projecten

3.5 Resources

3.5.1 Groepsgroottes en mankracht

In de parelweken zullen onder begeleiding van een docent, AiO of student assistent werkcolleges en practica in kleinere groepen plaatsvinden (richtlijn: 20-25 studenten, dwz 3 à 4 groepen waaronder één groep voor alle BIT-ers). Datzelfde geldt voor de sessies voor Academische Vaardigheden. We gaan werken met een vaste groep van 4 studentassistenten voor alle parelweken mits we studentassistenten voor een dergelijk schema kunnen vinden. Voor opdracht- en projectbegeleidingsmomenten zijn steeds 2 studentassistenten beschikbaar (men kan onderling uitmaken wie welke uren pakt).

Het project zal worden uitgevoerd in groepen van 6 studenten. Studenten mogen zelf de groepen indelen vóór week 4 (het 'echt' begin van het project). In de parelweken wordt er veelvuldig gekozen voor het werken in paren. Hiervoor worden studenten door ons ingedeeld, per activiteit opnieuw.

Voor het project zal elke groep een procesbegeleider aangewezen krijgen. Dit zal een ervaren docent zijn, omdat uit ervaring is gebleken dat een project alleen echt goed activerend werkt met goede coaching.

3.5.2 Zalen

We onderkennen drie soorten zalen:

GI "Groot Inflexibel": traditionele hoorcollegezaal voor plenaire sessies. Voor weken 2, 4, 6 moeten er ≈90 studenten in kunnen, voor de overige weken ≈60.

GF "Groot Flexibel": een zaal voor de gehele populatie studenten waarbij je de tafels kunt verzetten en waar ook voldoende stopcontacten aanwezig zijn (ivm laptopgebruik). Voor weken 2, 4, 6 moeten er ≈90 studenten in kunnen, voor de overige weken ≈60.

K "Klein": een traditionele werkcollegezaal voor ≈25 studenten. Voor weken 2, 4, 6 zijn er per week 4 dergelijke zalen nodig, voor de overige weken 3.

T "Toetsruimte": capaciteit ≈90 studenten voor weken 2, 4, 6, 9 en 10 en ≈60 voor de weken 1, 3, 5, 7. Één zaal is ideaal, maar verdelen over twee is ook een optie; dan zijn er simpelweg twee surveillanten benodigd.

De parel- en projectweken zijn zo ingericht dat ze zoveel mogelijk gebruik kunnen maken van dezelfde zalen. De verdeling hieronder geldt dus voor alle weken 1 t/m 10, tenzij anders vermeld.

Uren	Zaal	Bijzonderheden
ma 1+2	GI	weken 1 en 9
ma 6+7	GI	
ma 8+9	K	
di 1+2	GI	Alleen weken 1 t/m 8
di 3+4	K	
di 6+7	K	voor weken 1,3,5,7,9
	GF	voor weken 2,4,6,8,10
di 8+9	K	Alleen weken 1 t/m 8
wo 6+7	GF[2]	
do 1+2+3+4	GF[2]	
vr 3+4	T	
vr 6+7+8+9	GF[2]	Muv. weken 3, 5 en 8 ivm wiskundetoetsen

3.6 Vergelijking met ACM curriculum

In termen van het ACM/IEEE Computer Science Curriculum overlapt deze module met de inhoud van de volgende kernonderwerpen (zie: http://ai.stanford.edu/users/sahami/CS2013).

- AL/ Algorithms and Complexity (6 uur)
- AR/ Architecture and Organization (4 uur)
- HC/ Human Computer Interaction (2 uur)

[2]Deze zalen zijn bedoeld om de studenten van projectruimte te voorzien alsmede om een vaste plek te creëren waar de studentassistenten te vinden zijn voor vragen en hulp. Alternatieven zijn (a) twee zalen met een capaciteit van ≈45, of (b) één zaal met een capaciteit van ≈25 waar de studentassistenten zitten, waarbij we de studenten zelf geen projectruimte bieden.

- IAS/ Information Assurance and Security (2 uur)
- IM/ Information Management (1 uur)
- NC/ Networking and Communication (2 uur)
- OS/ Operating Systems (4 uur)
- PD/ Parallel and Distributed Computing (1 uur)
- PL/ Programming Languages (3 uur)
- SDF/ Software Development Fundamentals (19 uur)
- SE/ Software Engineering (8 uur)
- SP/ Social and Professional Practice (3 uur)

De genoemde uren zijn de geschatte contacturen volgens het ACM curriculum. Totaal beslaat deze module ongeveer 54 contacturen. Het hele Computer Science kerncurriculum beslaat 306 uur.

4 Weekindeling

4.1 Week 1: Pink Pearl — The Intestines of your Computer

Uur	Ma			Di			Wo			Do			Vr		
1	P	hc	D∞	P	hc	D∞	W	zs	N	P	col	D∞	P	zs	N
2	E	hc	D∞	P	col	D∞	W	zs	N	P	col	D∞	P	zs	N
3	W	hc	D∞	P	ipr2	S24	W	wc	D24	P	opd2	S	P	tts	D∞
4	W	hc	D∞	P	ipr2	S24	W	wc	D24	P	opd2	S	P	tts	D∞
6	P	hc	D∞	A	wc	D24	P	opd2	S	W	zs	N	P	opd2	S
7	P	col	D∞	A	wc	D24	P	opd2	S	W	zs	N	P	opd2	S
8	P	wc	D24	P	opd2	S	P	zs	N	W	wc	D24	P	opd2	S
9	P	wc	D24	P	opd2	S	P	zs	N	W	wc	D24	P	opd2	S

Afk	Werkvorm
col	Colstructie
hc	Hoorcollege
ipr	Instructie-practicum
opd	Opdracht
pfb	Peer feedback
prj	Project
qa	Vragenuur
wc	Werkcollege
tts	Toets
zs	Zelfstudie

Afk	Onderdeel
P	Parel
E	Eindproject
A	Academ. vaardigh.
W	Wiskunde

Afk	Begeleiding
D	Docent
S	Student-assistent
N	Geen

[Wiskunde] A Euclides: Sets, Numbers, Functions and Relations (Notations of sets; Operations on sets; Special subsets of real numbers; Inclusion-Exclusion principle; Logical statements, connectives; Quantifiers (logical operators); Truth Tables; Logical Reasoning)

4.1.1 Inleiding

In deze parel maakt de student kennis met de werking van de programmeerbare machine die we computer noemen. Essentieel daarvoor zijn binaire logica en rekenen, en de basis-architectuur van computers. De kennis wordt in de praktijk gebracht door een eenvoudige computer (microcontroller) in machinetaal te programmeren.

4.1.2 Leerdoelen

Na het absorberen van deze parel

- kan de student overweg met binaire logica (0, 1, AND, OR, EXOR) en met het binaire talstelsel;
- kan de student de essentie van de architectuur van een computer (Von Neumann / Harvard) uitleggen, alsmede de rol van de componenten van deze architectuur (register, ALU, geheugen, I/O, bus);
- kan de student eenvoudige programma's voor een microcomputer in machinetaal maken, gebruikmakend van arithmetische, I/O- en (conditionele) spronginstructies.

AP 4: *Leerdoelen toetsbaarder maken, waarmee ook inhoud van toets en werkcollege duidelijker wordt* **PTdB**

4.1.3 Indeling

Voorstel, nog verre van definitief:

Ma 1–2 Inleiding op de studie, de eerste module en de eerste opdracht.

Ma 6–9 Binaire logica en rekenen, middels hoorcollege en werkcollege.

Di 1–4 College computerarchitectuur en basis van machinetaal. Vraagstukjes computerarchitectuur en kennismaking Arduino.

Di 6–7 Academische Vaardigheden: Rollenspel: Oefening Intentie tot samenwerken.

Di 8–9 Opdracht realiseren.

Wo 6–7 Opdracht realiseren.

Wo 8–9 Zelfstudie Binaire logica en rekenen.

Do 1–2 Vraagstukken binaire logica en rekenen en computerarchitectuur.

Do 3–4 Opdracht realiseren.

Vr 1–4 Zelfstudie en toets

Vr 6–9 Opdracht realiseren.

4.1.4 Opdracht

In de opdracht maken de studenten een sirene + looplicht op de Arduino (AVR-processor), in machinetaal, met elementairst mogelijk gereedschap: (hexa?)decimale getallen intypen in een teksteditor, en deze met een door ons aan te leveren tooltje naar de Arduino sturen. Omdat in de AVR de instructietiming deterministisch is, is timing d.m.v. wachtlussen mogelijk. De opdracht laat flexibiliteit in hoe "fancy" het eindresultaat wordt.

Getalenteerde studenten zouden kunnen worden uitgedaagd zich in de on-chip-peripherals te verdiepen (bijv. timerinterrupt), of geavanceerder geluid te maken (bijv. met pulsbreedtemodulatie).

Ander idee voor getalenteerden is een wedstrijdelement in de vorm van het realiseren van een bepaalde berekening met zo weinig mogelijk instructies.

4.1.5 Docenten

PTdB + leden moduleteam 2.1.
Hoofddocent nog te bepalen.

4.1.6 Studiemateriaal

"Korte gedeelten" overgenomen uit boek(en) in reader; nog te selecteren.
Referentiemateriaal voor de praktische opdracht, vermoedelijk online: instructieset AVR, hardwaredocumentatie van de gebruikte Arduino-bordjes, documentatie over de nodige tools.

4.1.7 Academische Vaardigheden

Rollenspel: Oefening Intentie tot samenwerken (opsplitsen in groepen met ≈20 studenten). Leerdoelen van de bijeenkomst:

- erkennen van het belang van samenwerken
- erkennen van de valkuilen voor een goede samenwerking

4.2 Week 2: Black Pearl — Algorithms for Sorting and Searching

Uur	Ma			Di			Wo			Do			Vr		
1	W	tts	D∞	P	hc	D∞	W	zs	N	P	opd2	N	P	zs	N
2	W	tts	D∞	P	ipr	N	W	zs	N	P	opd2	N	P	zs	N
3	W	hc	D∞	P	ipr	S24	W	wc	D24	P	opd2	S24	P	tts	D∞
4	W	hc	D∞	P	ipr	S24	W	wc	D24	P	opd2	S24	P	tts	D∞
6	P	hc	D∞	A	col	D∞	P	opd2	S24	W	zs	N	P	opd2	S24
7	P	ipr	S24	A	col	D∞	P	opd2	S24	W	zs	N	P	opd2	S24
8	P	ipr	S24	P	ipr	S24	P	zs	N	W	wc	D24	P	opd2	N
9	P	pfb8	D24	P	pfb8	D24	P	zs	N	W	wc	D24	P	opd2	N

Afk	Werkvorm
col	Colstructie
hc	Hoorcollege
ipr	Instructie-practicum
opd	Opdracht
pfb	Peer feedback
prj	Project
qa	Vragenuur
wc	Werkcollege
tts	Toets
zs	Zelfstudie

Afk	Onderdeel
P	Parel
E	Eindproject
A	Academ. vaardigh.
W	Wiskunde

Afk	Begeleiding
D	Docent
S	Student-assistent
N	Geen

4.2.1 Wiskunde

A Euclides Logic (Elementary properties of functions; Onto, one-to-one; Direct proof; Contraposition / Contradiction - Counterexample; Mathematical induction)

B1 Leibniz Functions (Functions and their graphs; Domain, range, even, odd, increasing, decreasing; Power functions, polynomials, rational functions; Combining functions, shifting and scaling graphs; Composite functions; Trigonometric functions sin, cos, tan; Double-angle and half-angle formulas; Exponential functions and their rules; Inverse functions, one-to-one; Logarithmic functions and their rules)

4.2.2 Academische Vaardigheden

Interactief hoorcollege (IA1). Leerdoelen: (Level 2 Skill Sheets)

- ability to apply the principles of project management
- giving and receiving feedback
- formations of groups (internally / externally induced)

Studiemateriaal

G1 Principles of Effective Team Management
G9 Feedback and Coaching + Oefening voor Feedback geven en ontvangen
G2 Forming: Members, Roles and Dependencies

4.2.3 De parel — Inleiding

Algoritmisch denken is één van de basisvaardigheden die in ieder geval van elke student verwacht mag worden. Daarbij is inzicht in de complexiteit van het betreffende algoritme van wezenlijk belang; de taal waarin geprogrammeerd wordt veel minder. In deze parel wordt een elementaire klasse algoritmen, die voor zoeken en sorteren, gebruikt om de studenten met deze concepten kennis te leren maken.

Als programmeertaal is de keuze op Python gevallen, om de volgende redenen:

- Het is een taal met een vrij “schone” syntax
- De taal ondersteunt de (elementaire) concepten die voor deze parel nodig zijn: control structures en eenvoudige typen
- De taal leent zich voor programmeerwerk in het kader van het project van deze module
- Het is een gangbare taal, waarvan het waarschijnlijk is dat studenten deze later in de studie/het werk ook tegenkomen

4.2.4 Leerdoelen

Na het absorberen van deze parel

- kan de student het belang van zoek- en sorteer-algoritmen uitleggen;
- kan de student de werking van en het verschil tussen lineaire en binaire zoekmethoden uitleggen, evenals tussen insertion sort, bubblesort, merge sort en quicksort;
- begrijpt de student de complexiteitsargumenten bij bovenstaande algoritmen en kan hij analyseren wat in welke context de beste oplossing is
- kan de student eenvoudige imperatieve programmeerconcepten toepassen: if/then, while, integervariabelen en arrays;
- kan de student bovenstaande algoritmen in Python uitprogrammeren.

4.2.5 Indeling

Ma 6–9 In een hcwordt de rol van (imperatief) programmeren uitgelegd en hoe een Python-algoritme geïnterpreteerd moet worden, aan de hand van lineair zoeken. Daarna moeten studenten in een iprdit en binair zoeken zelf in Python implementeren aan de hand van pseudo-code. Vervolgens leggen ze aan elkaar uit (pfb) wat de wezenlijke verschillen tussen lineair en binair zoeken zijn (efficiëntie! Het binair zoeken moet door de studenten zelf uitgezocht worden). *Uitdaging: Kan je uitdrukken wat het efficiëntieverschil is, d.w.z. hoeveel stappen heeft het algoritme nodig voor het zoeken in een lijst met n elementen?*

N.B.: Dit veronderstelt een programmeeromgeving of -raamwerk waarin ze hier snel mee aan de slag kunnen en een behoorlijk deel al voorgekookt is. Elke student moet dit (individueel) op zijn laptop draaiend hebben, bij wijze van voorbereiding op de module!

Di 1–4 In een hc wordt het belang van sorteeralgoritmen uitgelegd, met de nadruk op formele complexiteit (d.w.z., efficiëntie). Studenten worden uitgenodigd te bedenken/vertellen hoe ze een auto's op een parkeerplaats zouden sorteren.

In de iprmoeten studenten twee algoritmen (bubblesort + merge sort, of verschillende keuzes voor verschillende groepen?) uitwerken tot een Python-implementatie, en deze testen aan de hand van een voorgegeven dataset.

N.B.: Om de opdrachten aantrekkelijker te maken en de mogelijkheid tot variatie te vergroten is het misschien wenselijk ook struct-arrays te gebruiken.

Di 8–9 Studenten gaan verder met het ipr. Aan het eind van de dag vertellen ze elkaar hun bevindingen resp. laten hun uitgewerkte programma's aan elkaar zien.

Resultaten van het ipr moeten vóór donderdag afgetekend zijn; dan begint de opdracht, later aftekenen leidt onvermijdelijk tot vertraging.

Wo 6–9 Studenten bestuderen de opd (voorzover nog niet gebeurd; de omschrijving staat in de handleiding!) en maken een keuze; spreken af wie wat gaat doen. De eerste twee uur is hier begeleiding bij.

Do 1–4 Verder werken aan de opd. De tweede helft van de ochtend is er begeleiding aanwezig.

Vr 1–4 Zelfstudie gevolgd door toets. In de toets komen (bijvoorbeeld) aan de orde

- Scenario's voor zoeken en sorteren, daarbij wordt gevraagd welk algoritme het meest geschikt is, met onderbouwing aan de hand van complexiteitsargumenten;
- Het toepassen van diverse algoritmen op een gegeven array, door middel van invullen van de opeenvolging van stappen;
- Een gegeven (onbekend) algoritme in Python (niet persé zoek/sorteer!), daarbij wordt gevraagd uit te leggen wat er gebeurt.

Vr 6–9 Afmaken en opleveren opd. De eerste helft van de middag is er begeleiding aanwezig.

4.2.6 Opdracht

Basisidee: Zoekmachine-functionaliteit zelf implementeren.

Gegeven Een (grote) ongeordende array van woord/doc.nr.-paren.
Gevraagd Een algoritme dat, gegeven een woord, een gesorteerde lijst doc.nrs oplevert, van goed naar slecht

Om dit te bereiken moeten de studenten de volgende stappen programmeren:

- Twee lijsten maken: (1) een op woorden gesorteerde lijst met begin- en eind-indices in de tweede lijst, en (2) een lijst met doc.nr/frequentieparen.
- Gegeven een woord, binair zoeken in lijst (1) en vervolgens het bijbehorende fragment van lijst (2) gesorteerd opleveren, bijv. naar aflopende frequentie

De opdracht kan (voor ambitiauze studenten) naar believen worden uitgebreid met

- Andere (slimmere) datastructuren, bijv. geneste lijsten
- Verfijndere sorteringscriteria van lijst documenten
- Zoeken naar meerdere woorden tegelijk

4.2.7 Docenten

Voorstel: Arend Rensink

4.2.8 Studiemateriaal

Wikipedia komt een heel eind:

- Binair zoeken: zie `http://en.wikipedia.org/wiki/Binary_search_algorithm`
- Sorteren: zie `http://en.wikipedia.org/wiki/Sorting_algorithm`
- Ter verluchting: Dropsort (complexiteit: $O(n)$) en Bogosort (complexiteit: $O(n!)$)
- Referentiemateriaal over Python

4.3 Week 3: Green Pearl — Cryptography

Uur	Ma			Di			Wo			Do			Vr		
1	W	tts	D∞	P	col	D∞	W	zs	N	P	opd2	S24	P	zs	N
2	W	tts	D∞	P	zs	N	W	zs	N	P	opd2	S24	P	zs	N
3	W	hc	D∞	P	col	D∞	W	wc	D24	P	qa	D∞	P	tts	D∞
4	W	hc	D∞	P	zs	N	W	wc	D24	P	zs	N	P	tts	D∞
6	P	hc	D∞	A	wc	D24	P	hc	D∞	W	zs	N	W	zs	N
7	P	hc	D∞	A	wc	D24	P	hc	D∞	W	zs	N	W	zs	N
8	P	opd2	S24	P	opd2	S24	P	zs	N	W	wc	D24	W	tts	D∞
9	P	opd2	S24	P	opd2	S24	P	zs	N	W	wc	D24	W	tts	D∞

Afk	Werkvorm
col	Colstructie
hc	Hoorcollege
ipr	Instructie-practicum
opd	Opdracht
pfb	Peer feedback
prj	Project
qa	Vragenuur
wc	Werkcollege
tts	Toets
zs	Zelfstudie

Afk	Onderdeel
P	Parel
E	Eindproject
A	Academ. vaardigh.
W	Wiskunde

Afk	Begeleiding
D	Docent
S	Student-assistent
N	Geen

[Wiskunde] A Euclides: Techniques of proving theorems (Factorials, permutations and combinations; Binomial of Newton; Proofs (Continued))
[Wiskunde] B1 Leibniz: Differentiation (Differentation rules (sum rule, product rule, quotient rule); Second and higher-order derivatives, notations; Derivatives of the trigonometric functions sin, cos, tan; Derivative of a composite function, the chain rule)

AP 5: *Security (cryptography): reduce ambitions* **MvK,AP**

4.3.1 Inleiding

Status en tijdspad: gesproken met Andreas Peter (19 feb). Hij bereidt een eerste versie van leerdoelen en indeling voor. Hij bespreekt dit met Pieter Hartel op z'n eerste werkdag (1 mrt). Ik (MvK) overleg dan weer met Andreas in de week van 4 maart en zal in die week de leerdoelen en indeling invullen.

4.3.2 Leerdoelen

After completion of this pearl

- students know some basic fundamentals of cryptography, such as symmetric and asymmetric encryption, as well as hash functions and digital signatures,
- students know the details of RSA and the One-Time-Pad, and know weaknesses of certain symmetric encryption schemes (such as Vigenere)
- students know the necessary background on elementary number theory (modular arithmetic, Euclidean algorithm) and basic probability theory, for a proper understanding of the above mentioned cryptosystems.

4.3.3 Indeling

Mon 6–7 Introductory lecture on cryptography. This contains:

- Why is crypto important? Real-world examples: e-passport, online banking
- History (Caesar, Viginere, Enigma), the "old view" on cryptography
- Kerckhoff's principals
- Some background on probability theory
- One Time Pad (OTP), symmetric-key encryption, what is security (definition)?
- Introduction of the student company "ACMEäs a motivating example for the öpdracht": The company uses certain encryption schemes (that we, "the teachers", provide) and it wants to know whether these schemes are secure. Therefore, the company hires some hackers (the students) who should try to break the used ciphers.

Mon 8–9 Time for öpdracht". Students are assigned to break certain "old ciphers" (such as Vigenere) by using certain analytical techniques (e.g., frequency analysis). They learn why such ciphers are always insecure.

Tue 1–4 Motivation for asymmetric crypto: How to exchange a symmetric key? Some mathematical background needed to understand the RSA cryptosystem (modular arithmetic, factoring, Euclidean algorithm). Introduction of RSA. Assignments on modular arithmetic, Euclidean algorithm, and RSA.

Tue 8–9 This slot is reserved for the ”Opdracht”. More hacking into ACME (also RSA).

Wed 6–7 High-Level introductory lecture on hash functions, MACs, and digital signatures (with constructions, e.g., Merkle-Damgaard, Block-Cipher-Based MACs, RSA signature scheme)

Wed 8–9 Self-study on RSA, RSA-signature, hash functions, MACs (with assignments).

Thu 1–2 Time reserved for the ”Opdracht”. More hacking, also on signatures.

Thu 3–4 Q&A on all the above mentioned topics and some nice cryptanalytic riddles.

Fri 1–2 Self-Study on everything that has been done in the module (with dedicated material for the students to prepare for a short exam/test)

Fri 3–4 Short exam/test

4.3.4 Opdracht

The öpdrachtï̈s to hack into the ACME company: A student company that produces certain software. Of course, the company wants to use secure encryption and signature schemes for data confidentiality and integrity. Therefore, they hire a bunch of hackers (the students of this course) to try to hack the used schemes. Some ciphers will be harder to break than others (and students can get bonus points if they manage to break these as well).

4.3.5 Docent

Hoofddocent: Andreas Peter, PostDoc van DIES-groep. “Echte” cryptograaf. Start per 1/3/13.

4.3.6 Studiemateriaal

4.3.7 Academische Vaardigheden

Bouwopdracht: Bouw een zo hoog mogelijke toren van papier / brug van spaghetti. Leerdoel:

- eerste inzicht in het functionieren van teams

4.4 Week 4: White Pearl — Software Engineering

Uur	Ma			Di			Wo			Do			Vr		
1	**W**	**hc**	**D∞**	**P**	**col**	**D∞**	W	zs	N	P	pfb	N24	P	zs	N
2	**W**	**hc**	**D∞**	**P**	**col**	**D∞**	W	zs	N	P	pfb	N24	P	zs	N
3	**W**	**hc**	**D∞**	**P**	**ipr**	**S24**	**W**	**wc**	**D24**	E	prj6	N	**P**	**tts**	**D∞**
4	**W**	**hc**	**D∞**	**P**	**ipr**	**S24**	**W**	**wc**	**D24**	E	prj6	N	**P**	**tts**	**D∞**
6	**P**	**hc**	**D∞**	**A**	**col**	**D∞**	**P**	**opd6**	**S24**	W	zs	N	E	prj6	N
7	**P**	**hc**	**D∞**	**A**	**col**	**D∞**	**P**	**opd6**	**S24**	W	zs	N	E	prj6	N
8	E	prj6	N	P	opd6	N	P	opd6	N	**W**	**wc**	**D24**	**E**	**hc**	**D∞**
9	E	prj6	N	P	opd6	N	P	opd6	N	**W**	**wc**	**D24**	**E**	**hc**	**D∞**

Afk	Werkvorm
col	Colstructie
hc	Hoorcollege
ipr	Instructie-practicum
opd	Opdracht
pfb	Peer feedback
prj	Project
qa	Vragenuur
wc	Werkcollege
tts	Toets
zs	Zelfstudie

Afk	Onderdeel
P	Parel
E	Eindproject
A	Academ. vaardigh.
W	Wiskunde

Afk	Begeleiding
D	Docent
S	Student-assistent
N	Geen

[Wiskunde] B1 Leibniz: First-Order Differential Equations (Models of exponential change; First-order differential equations and first-order initial value problems; (Particular) solution and the general solution; Applications, examples)

4.4.1 Inleiding

Deze parel wordt gevolgd door studenten INF en BIT, gezamenlijk.

In deze parel staat requirements analyse centraal, een aspect van cross-cutting concern 'System Development'. Deze parel besteedt daarnaast aandacht aan projectmatig werken, hetgeen in de opleiding herhaaldelijk terugkomt en ook gezien kan worden als academische vaardigheid. Dit is de eerste parel waarin alvast gewerkt wordt aan het eindproject. In deze parel wordt de requirements specification opgesteld van het stuk software dat tijdens het eindproject gerealiseerd gaat worden. De requirements specification is zelf een artefact dat ontworpen moet worden en waarop je de ontwerpcyclus kunt loslaten, en wat je beheerst en systematisch wilt doen. In deze parel behandelen we daarom technieken uit requirements engineering, de ontwerpcyclus, en 'formeel' projectmanagement (technieken voor planning en monitoring) en passen deze toe op het resultaat dat voor deze parel gerealiseerd moet worden.

Toetsing

- Individuele component: multiple-choice toets op vrijdag.
- Groepscomponent: beoordeling (voldoende/onvoldoende) van de opgeleverde functionele specificatie, beoordeling eventuele bonuspunten.

4.4.2 Leerdoelen

- Studenten kennis laten maken met het beheerst en voorspelbaar realiseren van software/projectresultaat;
- Studenten kennis laten maken met enkele technieken uit projectmanagement;
- Studenten leren om requirements/acceptatiecriteria op te stellen;

Naast het bereiken van deze leerdoelen is een expliciet extra doel van deze parel het opzetten van structuur voor het uitvoeren van het eindproject van de module.

4.4.3 Indeling

Ma 6-7 Inleiding/theorie in HC of colstructievorm: functionaliteit en kwaliteit (ISO 9126) van software, user stories, issue trackers, continuous integration / continuous deployment als state-of-the-art, projectmanagement. Doen: opstellen eisen voor resultaat dat deze week opgeleverd moet worden. Intro van de tools (bijv. Trello, zoals in deze kleine tryout of zoals in een serieus project. Deadline voor vorming groepen; tijd vrijhouden om laatste deelnemers aan groepen toe te wijzen.

Ma 8-9 Groepen eindproject gaan voor het eerst als groep aan het werk. Bespreken wat men wil bereiken, opzetten werkomgeving, spelen met de tools.

Di 1-4 Specificatiefase: colstructie over planning. Opstellen planning rest van de week. Werkvormen uit 1.1 IEBIT gebruiken om studenten uit te laten zoeken wat er in de Batavierenrace allemaal gebeurt.

Di 8-9 Werken aan de opdracht voor deze parel: opstellen requirements specification (we zouden ook kunnen stellen dat dit voor het eindproject is; cel zou blauw en groen tegelijk moeten zijn).

Wo 6-9 Realisatiefase: studenten stellen spec op in de vorm van een aantal user stories, evt. op basis van interviews, en leggen ze vast in Trello. Dit is het resultaat dat gebruikt wordt tijdens het eindproject.

Do 1-4 Evaluatiefase: groepen wisselen specs uit voor een peer review. Daarna passen ze het projectresultaat aan aan de feedback.

Vr 1-4 Voorbereiden toets, toetsing theorie projectmanagement en requirements engineering.

Vr 6-9 Presentaties specificatie.

4.4.4 Opdracht

Tijdens het eindproject van deze module wordt een stuk software ontworpen dat moet voldoen aan vooraf gestelde eisen. Dit gebeurt in een ontwikkelproces dat zo voorspelbaar en beheerst mogelijk moet zijn: tijdens het proces is op elk moment voor zowel het ontwikkelteam als bepaalde andere stakeholders duidelijk wat de stand van zaken is. In deze parel laten we studenten kennis maken met een paar technieken/hulpmiddelen hiervoor, zonder dat we dit uitputtend doen in de zin van het integraal behandelen van een methode zoals RUP of SCRUM (dat gebeurt later in andere modules). De technieken/hulpmiddelen worden enkel aangereikt en gemotiveerd. Alvorens in te gaan op deze parel kijken we eerst naar de relatie met het eindproject.
Waar gaat het om tijdens het eindproject?

- Aan het einde van het eindproject zal iets opgeleverd worden. Op dit moment is nog niet precies bekend wat dat is, maar we kunnen er wel vanuit gaan dat dat bestaat uit een aantal componenten en/of dat een aantal stappen doorlopen moet worden (met elk een deelresultaat) om tot het eindresultaat te komen. We gaan er niet vanuit dat tijdens deze parel al bekend is wat die stappen zijn / uit welke onderdelen het resultaat bestaat, wel dat er stappen/onderdelen zijn.
- Het eindproject wordt uitgevoerd aan de hand van een methode en de daaruit voortvloeiende planning. Het ontwerpteam van module 1.1 bepaalt wat die methode zal zijn: 3 iteraties die gezien kunnen worden als sprints (a la SCRUM), van elk 1 week.
- Tijdens het eindproject is duidelijk hoe de voortgang is en wie wat doet, zowel voor de groepen als voor de begeleiding (heeft ook een functie in controleren op meelifters).

Waar gaat het om tijdens deze parel?

- Hoogste aggregatieniveau: de ontwerpcyclus (specificatie - realisatie - evaluatie);
- Volgende aggregatieniveau: decompositie in componenten;
- Maken van een planning om de componenten te realiseren en te integreren tot het eindresultaat. In dit document nog lastig uit te voeren omdat ook het ontwerpteam nog niet weet wat het eindproject gaat worden;
- Monitoren van de uitvoering van het plan voor het behalen van het resultaat van deze week;
- Dit alles toepassen in de parel zelf (als vingeroefening voor het eindproject): aan het einde van deze parel moet ook iets opgeleverd worden.

Wat zijn de verwachte deliverables?

- Noodzakelijke deliverables voor verkrijgen voldoende beoordeling:
  - Functionele specificatie van eindproduct van het eindproject in de vorm van user stories die vastgelegd zijn in Trello.
  - Presentatie tijdens de presentatiesessie op vrijdag.
- Mogelijkheden voor bonuspunten:
  - User stories uitbreiden met acceptatie-criteria: testbare criteria waarmee een vertegenwoordiger van de Batavierenrace kan vaststellen in hoeverre de functionele specificatie gerealiseerd is.
  - Uitbreiding van de functionele specificatie met non-functionele (ofwel: kwaliteits-) eisen die SMART gedefinieerd zijn.

4.4.5 Docenten

- Pascal van Eck (hoofddocent)
- Klaas Sikkel
- Christiaan Katsma

4.4.6 Materiaal

- Skills Sheets: aantal onderdelen van Skill Sheets Series G ('Team and Project Managament', is trouwens vooral team management en bijna geen project management). Eventueel ook uit Series C ('Listening', gaat voornamelijk over interviewtechnieken).
- Videos, korte artikelen (niet noodzakelijk wetenschappelijk) over user stories, MoSCoW.
- Verplichte tekst over requirements engineering, te kiezen in overleg met Klaas Sikkel i.v.m. afstemming met module 1.2.
- Hoofdstuk 2 van *The Mythical Man-Month* (F. Brooks): adding manpower to a late project makes it later.

4.4.7 Academische Vaardigheden

Interactief hoorcollege (IA2). Leerdoelen: (Level 2 Skill Sheets)

- Understand the basics of effective brainstorming and meetings
- Basic knowledge of meeting and decision-making techniques

Studiemateriaal:

G3 Brainstorming
G4 Norming: Tasks and Roles
G5 Decision-making

4.5 Week 5: Red Pearl — Operating Systems and the Internet

Uur	Ma	Di	Wo	Do	Vr
1	W tts D∞	P hc D∞	W zs N		P zs N
2	W tts D∞		W zs N		P zs N
3	W hc D∞		W wc D24		P tts D∞
4	W hc D∞		W wc D24		P tts D∞
6	P hc D∞	A wc D24		W zs N	W zs N
7		A wc D24		W zs N	W zs N
8				W wc D24	W tts D∞
9				W wc D24	W tts D∞

Afk	Werkvorm
col	Colstructie
hc	Hoorcollege
ipr	Instructie-practicum
opd	Opdracht
pfb	Peer feedback
prj	Project
qa	Vragenuur
wc	Werkcollege
tts	Toets
zs	Zelfstudie

Afk	Onderdeel
P	Parel
E	Eindproject
A	Academ. vaardigh.
W	Wiskunde

Afk	Begeleiding
D	Docent
S	Student-assistent
N	Geen

[Wiskunde] B1 Leibniz: Vectors and the Geometry of Space (Coordinate Systems; Vectors, vector operations, parallelogram law; Length of a vector, (standard) unit vector(s); Dot product, angle between two vectors, orthogonal; Cross product, torque vector)

4.5.1 Inleiding

Doel van deze parel is studenten kennis te laten maken met de “high-level” kant van computersystemen, namelijk de besturingssystemen en netwerken. De nadruk ligt op de netwerken, omdat het Internet en het WWW in het bijzonder toch wat meer parelglans hebben dan besturingssystemen.

4.5.2 Leerdoelen

Na het absorberen van deze parel

- kan de student de belangrijkste taken van een besturingssysteem identificeren en beschrijven;
- heeft de student inzicht in de werking en opbouw van computernetwerken en kan overweg met begrippen als packetswitching, protocol en layering;
- heeft de student kennis van de globale opbouw van het internet en internetapplicaties en kan overweg met begrippen als TCP/IP, IP-adres, poortnummer, HTML.

4.5.3 Indeling

In grote lijnen: twee halve parels, een over OS en een over internet, elk met hoor- en werkcollege en een opdrachtje.

AP 6: *Indeling parel 5 verder uitwerken.* **PTdB**

4.5.4 Opdracht

Aangezien studenten in dit stadium nog nauwelijks kunnen programmeren, is het niet doenlijk ze een (simpele) applicatie te laten bouwen. Wat wel kan, is observeren en onderzoeken. Een grote opdracht zoals in de andere parels (2 of meer dagdelen) is moeilijk te vinden; maar meerdere kleine opdrachten zou wel kunnen, gesplitst over OS en internet.
Ideeën voor OS-observaties:

- Zoek experimenteel uit welk interval je OS gebruikt voor scheduling (hier zit meteen een vleugje concurrency in!); dit is goed te doen door een programmaatje te schrijven dat voortdurende de tijd opvraagt en iets logt als er een gat valt.
- Zoek experimenteel uit wat voor caching-strategie je OS gebruikt; bijv. na hoeveel tijd of hoeveel data er naar disk geschreven wordt.
- Zoek experimenteel uit hoeveel geheugen je OS je laat alloceren.

Ideeën voor netwerk-observaties:

- Hoe groot is het internet? Probeer een zo lang mogelijk (= meeste hops) pad te vinden.
- Hoe langzaam is het internet? Probeer een zo langzaam mogelijk (= grootste round-trip-time) pad te vinden.
- Probeer te meten hoeveel buffergrootte er in een (door ons beschikbaar gestelde) switch zit.
- Observaties van parameters zoals initiele TTL van verschillende OS'en.
- Als twee laptops aan hetzelfde WiFi-accesspoint hangen en data uitwisselen, gaat dat dan via het accesspoint?

AP 7: *concrete opdracht(en) parel 5 bedenken en uitwerken.* **PTdB**

4.5.5 Docenten

Pieter-Tjerk de Boer e.a.

AP 8: *Mede-docent/ontwerper met meer OS-expertise zoeken* **PTdB**

4.5.6 Studiemateriaal

Voor de hand ligt (kopie in reader van (een deel van)) hoofdstuk 1 uit Peterson&Davie, dat we ook in module 1.3 gaan gebruiken.

4.5.7 Academische Vaardigheden

Team meeting: Generating Feedback on first phase of project (G3, G4, G5). Invullen Belbin Test en bespreken in de groep wat de uitkomsten voor een invloed op het project (kunnen) hebben. Leerdoelen:

- reflecteren op de eerste project week (zijn de technieken uit IA1 en IA2 toegepast ja / nee / waarom niet)
- toepassen van de Belbin rolen op de projectgroepen en inzicht in het eigen functioneren in teams

4.6 Week 6: Blue Pearl — Functional Programming

Uur	Ma	Di	Wo	Do	Vr
1	**W tts D∞**	**P hc D∞**	W zs N	P opd	P zs N
2	**W tts D∞**	P ipr	W zs N	P opd	P zs N
3	**W hc D∞**	P ipr	**W wc D24**	P opd	**P tts D∞**
4	**W hc D∞**	P ipr	**W wc D24**	P opd	**P tts D∞**
6	**P hc D∞**	**A wc D∞**	P opd	W zs N	
7	P ipr	**A wc D∞**	P opd	W zs N	
8	P ipr	P ipr	P zs	**W wc D24**	
9	P pfb	P pfb	P zs	**W wc D24**	

Afk	Werkvorm
col	Colstructie
hc	Hoorcollege
ipr	Instructie-practicum
opd	Opdracht
pfb	Peer feedback
prj	Project
qa	Vragenuur
wc	Werkcollege
tts	Toets
zs	Zelfstudie

Afk	Onderdeel
P	Parel
E	Eindproject
A	Academ. vaardigh.
W	Wiskunde

Afk	Begeleiding
D	Docent
S	Student-assistent
N	Geen

4.6.1 Wiskunde

B1 Leibniz Second-Order Differential Equations (Second-order linear differential equations, (non)homogeneous; The Superposition Principle; General solution of a constant-coefficient homogeneous linear differential equation, the characteristic equation; General solution of a constant-coefficient nonhomogeneous linear differential equation; Method of undetermined coefficients; Applications to vibrations or electric circuits)

4.6.2 Academische Vaardigheden

Interactief hoorcollege (IA3). Leerdoelen: (Level 3 Skill Sheets)

- Understand decision-making techniques and how to effectively organise meetings

Studiemateriaal:

G6 Basics of Effective Negotiations
G7 Group Contract
G8 Effective Meetings

4.6.3 Programmeertaal

Gekozen wordt voor Haskell als programmeeraal. Overwegingen hierbij zijn:

- De taal is wijd verbreid en wordt in de praktijk toegepast
- Het is een lazy taal
- Hoewel de foutmeldingen cryptisch kunnen zijn, blijkt dit in de praktijk van het practicum mee te vallen.

Als andere kandidaten zijn Scala en Amanda overwogen, maar tegen beide bestaan teveel bezwaren.
Overwegingen voor de te kiezen taal zijn:

Haskell *Vóór:* Wijd verbreid, in de praktijk gebruikt; lazy; web-interface beschikbaar. *Tegen:* Cryptische foutmeldingen.
Amanda *Vóór:* Geschikt voor onderwijs; lazy. *Tegen:* Verouderd, draait niet meer stabiel.
Scala *Vóór:* Gecompileerd naar JVM; wijd verbreid. *Tegen:* eager.

4.6.4 Leerdoelen

Na het absorberen van deze parel

- Kent de student de basisconcepten van de gekozen functionele taal
- Begrijpt de student de principes van recursie en het verband met inductie
- Kan de student het concept van functie-applicatie uitleggen
- Kan de student het verschil tussen applicatief (functioneel) en state-gebaseerd (imperatief) programmeren uitleggen;
- Kan de student standaard-sorteeralgoritmen in de gekozen functionele taal schrijven.

4.6.5 Docent

Jan Kuper.

4.6.6 Indeling

Ma 6–9 In een hc wordt de functioneel programmeren geïntroduceerd, en de essentie van het verschil met imperatief programmeren uitgelegd. Eenvoudige voorbeelden van functiedefinities worden besproken in samenhang met de typering daarvan. Recursie zal worden geïntroduceerd onder verwijzing naar het bij wiskunde behandelde principe van volledige inductie.

In een practicumomgeving zullen in onderlinge samenwerking verschillende elementaire opdrachten worden gedaan. Het practicum zal worden begonnen met het installeren op de laptop van de gekozen programmeeromgeving.

Di 1–4 In een hc worden lijsten en het definiëren van (recursieve) functies daarop besproken. Dit zal worden toegepast in het op verschillende manieren sorteren van lijsten, zodat aansluiting ontstaat bij de algoritmen besproken in de zwarte parel (imperatief programmeren). Indien de tijd het toelaat zullen enkele eenvoudige hogere orde functies worden besproken (map, filter) worden besproken.

In het ipr zal geoefend worden met het definiëren van recursieve functies op lijsten waarbij aandacht besteed zal worden aan het implementeren van sorteerfuncties.

Di 8–9 De studenten gaan verder met het ipr. De dag wordt afgesloten met een gezamenlijke bespreking van de resultaten en van de principes van de programmeerparadigma's.

Wo 6–9 De middag wordt beginnen met een MHC waarin enkele complexere voorbeelden van de besproken principes zullen worden behandeld zoals matrixvermeningvuldiging. Die voorbeelden zullen in een IPR door de studenten verder worden uitgewerkt. De middag wordt afgesloten met het bespreken van een grotere opdracht die de volgende dag zal moeten worden gemaakt.

Do 1–4 Uitwerken van de opdracht die de vorige dag is besproken.

Vr 1–4 Zelfstudie gevolgd door toets. In de toets komen kleine programmeeropdrachten aan de orde zoals ook bij de practische opdrachten geoefend zijn.

Vr 6–9 Afronden opdracht.

4.6.7 Opdracht

To be done

4.7 Week 7: Yellow Pearl — Visual and Intelligent

Uur	Ma			Di			Wo			Do			Vr		
1	W	tts	D∞	P	hc	D∞	W	zs	N	E	prj6	N	P	zs	N
2	W	tts	D∞	P	hc	D∞	W	zs	N	E	prj6	N	P	zs	N
3	W	hc	D∞	P	opd6	S	W	wc	D24	E	prj6	S	P	tts	D∞
4	W	hc	D∞	P	opd6	S	W	wc	D24	E	prj6	S	P	tts	D∞
6	P	hc	D∞	A	wc	D24	P	hc	D∞	W	zs	N	E	prj6	N
7	P	hc	D∞	A	wc	D24	E	prj6	N	W	zs	N	E	prj6	N
8	P	opd6	D24	P	opd6	N	E	prj6	N	W	wc	D24	E	hc	D∞
9	P	pfb6	D24	P	pfb6	S24	E	prj6	N	W	wc	D24	E	hc	D∞

Afk	Werkvorm
col	Colstructie
hc	Hoorcollege
ipr	Instructie-practicum
opd	Opdracht
pfb	Peer feedback
prj	Project
qa	Vragenuur
wc	Werkcollege
tts	Toets
zs	Zelfstudie

Afk	Onderdeel
P	Parel
E	Eindproject
A	Academ. vaardigh.
W	Wiskunde

Afk	Begeleiding
D	Docent
S	Student-assistent
N	Geen

[Wiskunde] B1 Leibniz: Complex numbers (Real part, imaginary part, complex conjugate; Absolute value, argument, polar form; Geometric representations of a complex number; Euler's formula, De Moivre's Theorem; Complex arithmetic operations and related geometric illustrations; Roots of a complex number and the Fundamental Theorem of Algebra)

4.7.1 Inleiding

Als uitgangspunt zijn de volgende hoofdonderwerpen voor deze parel gekozen:

- Intelligentie in interactie
- Data visualisatie (optioneel; bestaand stukje onderwijs van Chintan Amrit zou hier goed zou passen)

4.7.2 Leerdoelen

Na het absorberen van deze parel

- kan de student een stakeholder- en contextanalyse doen en de resultaten interpreteren en gebruiken bij het ontwerpen en genereren van sociaal intelligente interactie;
- heeft de student kennisgemaakt met de eerste beginselen van artificiële intelligentie en kan een eenvoudig regelgebaseerd sociaal intelligent systeem ontwerpen;
- heeft de student kennisgemaakt met enkele basispricipes van machine learning en heeft een classificatietaak bedacht en uitgevoerd met een (black-box) classifier.

4.7.3 Indeling

Ma 6+7 Hoorcollege "Socially Intelligent Interaction": stakeholders, context, applications. Schets van het ontwerptraject: analyse/perceptie; interpretatie; ontwerp/generatie. Voorbereiding voor de opdracht: te gebruiken tools (bijvoorbeeld woordteller, resultaat van sentiment mining).

Ma 8+9 (indien mogelijk al starten tijdens 7^{e} uur). Start opdracht; deze middag:

- Verfijning en of motivatie requirements vanuit raamwerk sociale karakteristieken van de doelgroep (= eerder gekozen stakeholder) en contextanalyse.
- Tweetanalyse.
- Bedenken kleine beperkte vraag — relevant voor de stakeholder — die de projectgroep deze middag wil beantwoorden met behulp van sociaal intelligentie technologie.

Werk in de projectgroepen van 6 personen, 4 projectgroepen samen in één werkcollegegroep met begeleiding door 1 docent. In 9e uur peer feedback/discussions onder leiding van de docent.

Di 1+2 Hoorcollege "Inleiding AI en Machine Learning" (light; al snel toespitsen op rule-based systems met event-condition-action (ECA) regels en classifiers)

Di 3+4 en 8+9 Continuering opdracht; vandaag:

- Bedenken vraag — relevant voor de stakeholder — die de projectgroep wil beantwoorden met ML-technieken.

- Opstellen/bedenken ECA-rules gerelateerd aan sociale interactie zodat deze te koppelen is aan het ML onderdeel (minstens één (of toch optioneel?). Voorbeeld: indien tweet *gezellig* is ...). Verder merendeels acties die gekoppeld zijn aan events zoals het voorkomen van woorden of aan het vaak (meer dan X keer) voorkomen van bepaalde woorden in korte tijd (Y minuten).
- Selectie training- en test-set ML.
- 9e uur: feedback (peer reviews olv. docent) op rules, training- en test-set.

Waarschijnlijk deel van de tijd zonder begeleiding of met begeleiding studentassistent ipv. docent.

Wo 6–9 Eventueel hier nog starten met HC over datavisualisatie? Verder werken aan de opdracht:

- Training classifier en rapportage daarover.
- Maken rule-based system.

Do 1–4 Werken aan de opdracht: Afmaken rule-based system en rapportage daarover.

Vr 1–4 Toets met daarvoor zelfstudie voor toets.

Vr 6+7 Voorbereiding presentaties.

Vr 8+9 Presentaties over het gedane werk en demo van de uiteindelijke oplossingen (weer in de werkcollegegroepen van 4 projectgroepen waar steeds in is samengewerkt)

4.7.4 Opdracht

Ontwerp een rule-set (event-action of event-condition-action regels) voor een intelligent agent tbv de stakeholder. Aan de event/condition-kant kunnen studenten gebruik maken van events zoals ”woord X in tweet”, ”woord uit verzameling S komt meer dan N keer voor in de afgelopen M minuten”, etc. Aan de action-kant kunnen studenten acties doen in de interface zoals ”signaleer alarm”, ”plaats X op kaartjeöf acties voor het bijhouden van tellertjes enzo.

Optioneel (en dus voor bonuspunt): er wordt ook een machine learning-gebaseerde classifier aangeboden (als black-box). Studenten kunnen een trainingset aanleveren van berichten (bijvoorbeeld, berichten die een ongeval aangeven). Voor bonuspunten kunnnen ze de uitspraken van die classifier gebruiken in hun ruleset.

De opdracht is om te komen tot een ontwerp van de user-interface voor de dashboard van het project (deliverable die nuttig is voor het project).

De rol van de stakeholder is zeer belangrijk in dergelijk ontwerp. Ons idee is om de projectgroepen in week 4 de opdracht te geven om een *representant* te vinden voor de gekozen stakeholder; iemand die zich daar goed in zou kunnen inleven. Denk bijvoorbeeld aan een politieagent in de familie, hardloper bij de buren, iemand van de muziekvereniging die wel eens met een bandje muziek maakt langs de kant van de marathon. Deze persoon moet bekend zijn in week 7 en moet beschikbaar zijn voor een overleg waarin de projectgroep de werkende intelligent agent demonstreert aan die representant tbv feedback.

4.7.5 Docenten

4.7.6 Studiemateriaal

4.7.7 Academische Vaardigheden

Team meeting: Generating Feedback on the project (and especially on G6, G7, G8) (in te plannen aan het einde van de week). Leerdoel:

- reflecteren op de afgelopen project week, zijn de geleerde technieken toegepast?

4.8 Week 8

Uur	Ma			Di			Wo			Do			Vr		
1	**W**	**tts**	**D∞**	E	prj6	N	W	zs	N	**E**	**prj6**	**S24**	E	prj6	N
2	**W**	**tts**	**D∞**	E	prj6	N	W	zs	N	**E**	**prj6**	**S24**	E	prj6	N
3	**W**	**hc**	**D∞**	**E**	**hc**	**D∞**	**W**	**wc**	**D24**	E	prj6	N	**E**	**hc**	**D∞**
4	**W**	**hc**	**D∞**	**E**	**hc**	**D∞**	**W**	**wc**	**D24**	E	prj6	N	**E**	**hc**	**D∞**
6	**E**	**hc**	**D∞**	**A**	**col**	**D∞**	E	prj6	N	W	zs	N	W	zs	N
7	E	prj6	N	**A**	**col**	**D∞**	E	prj6	N	W	zs	N	W	zs	N
8	E	prj6	N	**E**	**prj6**	**S24**	**E**	**prj6**	**D24**	W	zs	N	**W**	**tts**	**D∞**
9	**E**	**prj6**	**D24**	**E**	**prj6**	**S24**	**E**	**prj6**	**D24**	W	zs	N	**W**	**tts**	**D∞**

Afk	Werkvorm
col	Colstructie
hc	Hoorcollege
ipr	Instructie-practicum
opd	Opdracht
pfb	Peer feedback
prj	Project
qa	Vragenuur
wc	Werkcollege
tts	Toets
zs	Zelfstudie

Afk	Onderdeel
P	Parel
E	Eindproject
A	Academ. vaardigh.
W	Wiskunde

Afk	Begeleiding
D	Docent
S	Student-assistent
N	Geen

4.8.1 Opzet van deze week

Thema van de week: uitgaande van de specificatie die is opgesteld in de SE-parel een ontwerp maken voor het eindresultaat van de module (zie paragraaf 3.2.3 voor een omschrijving van het beoogde eindresultaat). Concreet betekent dit dat in deze week de (functionele) eisen die vastgesteld zijn bij de SE-parel vertaald worden in een globale opzet van de ECA-regels en interfaces met de aangeboden user-interface elementen en beschikbare data. Deel hiervan is ook het opstellen van een planning: het identificeren van de stappen/acties die nodig zijn om dat artefact te kunnen opleveren. We bieden theoretische ondersteuning in de vorm van net voldoende modellen/aanpakken om greep te krijgen op het ontwerpen van eenvoudige software (hierbij kunnen we technieken uit de huidige cursus Informatiesystemen van Klaas Sikkel gebruiken).

Deliverables:

- Document dat ontwerp(-keuzes) presenteert en motiveert, voorzien van planning voor het realiseren van het ontwerp.
- Presentatie van ontwerp.

4.8.2 Weekindeling

Randvoorwaarde bij het ontwerp van de weekindeling is dat het aantal contacturen excl. wiskunde en excl. AV niet hoger is dan 12, waarvan 4 uur inzet student-assistenten.

- Maandag: hele ochtend vrijhouden voor wiskunde (toets, feedback en HC terugkoppeling). Middag: project kickoff. Opstellen planning en feedback op planning.
- Dinsdag: eerste ronde ontwerp; dit moeten we structureren. Bijv. hoorcollege over de zogenaamde event list (techniek uit huidige cursus Informatiesystemen). Het 6e en 7e uur zijn voor Academische Vaardigheden.
- Woensdag: ochtend is voor wiskunde, middag voor uitwerken project.
- Donderdag 12:30u strikt: opleveren weekresultaat, middag is voor wiskunde.
- Vrijdag: ochtend maken en geven presentaties. Middag: wiskunde.

4.8.3 Toetsing en beoordeling

Indrukcijfer (telt niet mee) op basis van presentatie.

4.8.4 Academische Vaardigheden

Interactief hoorcollege (IA4). Leerdoelen: (Level 2/3 Skill Sheets)

- Identify free-riding behaviour in others and yourself
- Understand that free-rider behaviour takes different forms throughout the various project stages; take effective action

Studiemateriaal:

G10 Unhealthy Group Dynamics: Groupthink
G11 Unhealthy Group Dynamics: Free-riding
G12 Dealing with conflicts

4.9 Week 9

Uur	Ma	Di	Wo	Do	Vr
1	**P hc D∞**	E prj6 N	**E pfb D24**	E prj6 N	**P tts D∞**
2	**P hc D∞**	E prj6 N	**E pfb D24**	E prj6 N	**P tts D∞**
3	E prj6 N	E prj6 N	E prj6 N	**E prj6 S6**	**P tts D∞**
4	E prj6 N	E prj6 N	E prj6 N	**E prj6 S6**	**P tts D∞**
6	E prj6 N	**E prj6 D6**	E prj6 N	E prj6 N	E prj6 N
7	E prj6 N	**E prj6 D6**	E prj6 N	E prj6 N	E prj6 N
8	**E prj6 S6**	E prj6 N	**E prj6 D6**	E prj6 N	**E hc D∞**
9	**E prj6 S6**	E prj6 N	**E prj6 D6**	E prj6 N	**E hc D∞**

Afk	Werkvorm
col	Colstructie
hc	Hoorcollege
ipr	Instructie-practicum
opd	Opdracht
pfb	Peer feedback
prj	Project
qa	Vragenuur
wc	Werkcollege
tts	Toets
zs	Zelfstudie

Afk	Onderdeel
P	Parel
E	Eindproject
A	Academ. vaardigh.
W	Wiskunde

Afk	Begeleiding
D	Docent
S	Student-assistent
N	Geen

4.9.1 Opzet van deze week

Thema van de week: uitprogrammeren van het ontwerp dat in week 8 is gemaakt, resulterend in werkende software. Het docententeam van deze module bouwt een systeem dat al helemaal werkt op de invulling van de ECA-rules na, studenten van deze module voegen daaraan functionaliteit toe. Berkeley heeft hier goede ervaringen mee[3]. De docenten houden zich zelf ook aan wat we in de parels leren: we gebruiken één van de programmeertalen die in de parels ook gebruikt zijn (een andere taal invoeren is wat teveel van het goede, en studenten kunnen de indruk krijgen dat wat ze eerder geleerd hebben niet bruikbaar is als je daadwerkelijk iets niet-triviaals wilt maken), en we passen technieken uit de parel SE zelf toe.
Los van de manier waarop we laten uitprogrammeren is het idee dat in deze week een eerste release wordt opgeleverd die elke groep op z'n minst op een 6 brengt:

- Opleveren op donderdag is verplicht. Geen uitzonderingen, geen uitstel, geen excuses. Zoals bij de Europese projectaanvragen[4]. Niet inleveren is een 1, en dus geen eindcijfer voor de module (want de eis is dat elk deelcijfer minstens 5 is). Leeg document inleveren en presentatie geven op vrijdag is voldoende om in de race te blijven.
- Docenten houden de donderdagavond en/of vrijdagochtend vrij voor beoordelen; op vrijdagmiddag direct na de presentaties of maandagochtend is feedback beschikbaar en weten de groepen waar ze aan toe zijn. Dit is echter geen deelcijfer, enkel een voorlopige beoordeling (alleen een 1 bij niet inleveren is niet voorlopig).
- Groepen die nog niet op het niveau van een 6 zitten hebben nog één week om tot een 6 (of hoger) te komen. Groepen die al wel op een 6 zitten kunnen functionaliteit toevoegen voor een hoger cijfer (of zich richten op de wiskundetoets, zou zelfs per persoon kunnen verschillen).

Tijdens deze week is er een mini-parel Databases. Deze parel verschilt in twee opzicht van de overige zeven: (1) er zijn geen 'losse opdrachten': het eindproject zelf vormt de enige context waarin studenten het geleerde toepassen, en (2) er is geen individuele toets zoals bij de andere parels.

4.9.2 Weekindeling

Randvoorwaarde bij het ontwerp van de weekindeling is dat het aantal contacturen excl. wiskunde en excl. herkansingen niet hoger is dan 14, waarvan 4 uur inzet student-assistenten en 2 uur voor hoorcollege mini-parel.

- Maandag: kickoff week 2 en MHC/HC Databases. Rest van de dag: werken aan eindresultaat voor vrijdag.
- Dinsdag: werken aan eindresultaat voor vrijdag
- Woensdag: ochtend: sessies met begeleiders en eventueel andere groepen: hoe gaat het? Waar loopt men tegenaan? Reflectie op proces, eventueel bijsturen.
- Donderdag 17:30u strikt: opleveren resultaat 1e iteratie.
- Vrijdag: ochtend is voor herkansingen parel-toetsen, middag is voor maken en geven presentaties.

[3] Zie Armando Fox and David Patterson (2012): Crossing the Software Education Chasm: An Agile approach that exploits cloud computing. *Comm. ACM*, 55(5):44–49, rechterkolom p. 47.

[4] We moeten er voor zorgen dat studenten al ruim vóór de deadline kunnen inleveren, en dat ze ingeleverd werk voor de deadline nog kunnen herzien. Net als bij de Europese projectaanvragen.

4.9.3 Toetsing en beoordeling

Voorlopig groepscijfer voor opgeleverd eindresultaat.

4.9.4 Academische Vaardigheden

Geen bijeenkomst deze week.

4.10 Week 10

Uur	Ma	Di	Wo	Do	Vr
1	W zs N	E prj6 N	**E pfb D24**	zs N	**P tts D∞**
2	**W tts D∞**	E prj6 N	**E pfb D24**	zs N	**P tts D∞**
3	**W tts D∞**	**E prj6 S6**	E prj6 N	zs N	**P tts D∞**
4	**W tts D∞**	**E prj6 S6**	E prj6 N	zs N	**P tts D∞**
6	**E hc D∞**	**A col D∞**	E prj6 N	zs N	zs N
7	E prj6 N	**A col D∞**	E prj6 N	zs N	zs N
8	E prj6 N	E prj6 N	**E hc D∞**	zs N	zs N
9	**E prj6 D6**	E prj6 N	**E hc D∞**	zs N	zs N

Afk	Werkvorm
col	Colstructie
hc	Hoorcollege
ipr	Instructie-practicum
opd	Opdracht
pfb	Peer feedback
prj	Project
qa	Vragenuur
wc	Werkcollege
tts	Toets
zs	Zelfstudie

Afk	Onderdeel
P	Parel
E	Eindproject
A	Academ. vaardigh.
W	Wiskunde

Afk	Begeleiding
D	Docent
S	Student-assistent
N	Geen

4.10.1 Opzet van deze week

Deze week is minder intensief dan de tweede projectweek. Strikt genomen zou een groep die al tevreden is met de voorlopige beoordeling en geen herkansingen heeft vrij kunnen nemen (al moet men wel verplicht opnieuw inleveren en een presentatie geven). In het midden van de week is er weer een presentatiesessie: eis is dat dit een andere presentatie is, gegeven door andere leden van het team. Zo dwingen we af dat er nog wat gebeurt in deze week. De laatste twee dagen zijn cf. TOM-beleid vrijgehouden voor aanvulling en herkansing.

Inhoudelijk bestaat deze week uit een iteratie: teams gaan gewoon verder waar ze vorige week gebleven zijn in het werken naar een definitief eindresultaat. Hierin zou nog wat meer vrijheid kunnen zitten: wat wij in de projectomschrijving eisen zou te doen moeten zijn tijdens de eerste iteratie, deze iteratie is voor de extra's die groepen tijdens de parel SE (en tijdens de eerste iteratie) bedacht hebben.

Er is een strikte deadline donderdag 17:30u (met mogelijkheid om al vanaf maandag in te leveren). Niets inleveren is een 1, ook al was de voorlopige beoordeling hoger. Docenten houden de vrijdagochtend vrij om na te kijken, zodat al op vrijdagmiddag een beoordeling gegeven kan worden.

4.10.2 Weekindeling

Randvoorwaarde bij het ontwerp van de weekindeling is dat het aantal contacturen excl. wiskunde, excl. herkansingen, en excl. AV niet hoger is dan 14, waarvan 4 uur inzet student-assistenten.

- Maandag: hele ochtend vrijhouden voor wiskundetoets. Middag: kickoff week 3 en feedback week 2 voor groepen die dat nodig hebben, groepen gaan verder waar ze gebleven waren.
- Dinsdag: werken aan eindresultaat voor vrijdag. Het 6e en 7e uur zijn voor Academische Vaardigheden.
- Woensdag: ochtend: sessies met begeleiders en eventueel andere groepen: welke puntjes moeten nog op welke i's? Middag: maken en geven presentaties.
- Donderdag: vrij gehouden, 17:30u strikt: opleveren eindresultaat.
- Vrijdag: ochtend is voor herkansingen parel-toetsen, middag is vrij gehouden; deze tijd is beschikbaar voor oplossingen voor studenten waarvoor individueel iets herkanst moet worden, bijv. i.v.m. ziekte (denk bijv. aan een extra sessie Academische Vaardigheden voor een klein groepje studenten dat te vaak afwezig is geweest, of een mondeling wiskunde).

4.10.3 Toetsing en beoordeling

Groepscijfer voor opgeleverd eindresultaat.

4.10.4 Academische Vaardigheden

(Na afloop) Team meeting: Generating Feedback on the project (and especially on G10, G11, G12) / Debriefing. Leerdoelen:

- reflecteren op de afgelopen project weeken, zijn de geleerde technieken toegepast?
- defineren van persoonelijke leerdoelen voor de komende projecten

4.11 Week 11

Uur	Ma	Di	Wo	Do	Vr
1	W tts D∞	P col D∞	W zs N	E prj6 N	P zs N
2	W tts D∞	P col D∞	W zs N	E prj6 N	P zs N
3	W hc D∞	P ipr D24	W wc D24	P pfb S24	P tts D∞
4	W hc D∞	P ipr D24	W wc D24	P ipr S24	P tts D∞
6	P hc D∞	A col D∞	E prj6 N	W zs N	W zs N
7	P col D∞	A col D∞	E prj6 N	W zs N	W zs N
8	P ipr D24	P opd6 N	E prj6 N	W zs N	W tts D∞
9	P ipr D24	P opd6 N	E prj6 N	W zs N	W tts D∞

Afk	Werkvorm
col	Colstructie
hc	Hoorcollege
ipr	Instructie-practicum
opd	Opdracht
pfb	Peer feedback
prj	Project
qa	Vragenuur
wc	Werkcollege
tts	Toets
zs	Zelfstudie

Afk	Onderdeel
P	Parel
E	Eindproject
A	Academ. vaardigh.
W	Wiskunde

Afk	Begeleiding
D	Docent
S	Student-assistent
N	Geen

4.11.1 Inleiding

De BIT-studenten krijgen in hun module 1.3 basiskennis over databases te weten: databaseschemaontwerp met ER-diagrammen, databasecreatie en SQL. Aangezien BIT en INF module 1.4 “Data en Informatie” delen, hebben de BIT-studenten hiermee een voorsprong in kennis op de INF-studenten. Een logisch uitgangspunt voor deze parel is om de INF-studenten wat voorkennis te geven op dezelfde onderwerpen. Daarnaast willen we in deze parel, omdat het de laatste is, ruimte geven voor het werken aan het project.

4.11.2 Leerdoelen

Na het absorberen van deze parel

- kan de student een ontwerp maken van een databaseschema voor een eenvoudige casus mbv ER-diagrammen.
- kan de student dat ontwerp realiseren mbv SQL.
- kan de student eenvoudige vragen en gegevensmanipulaties uitdrukken in SQL.
- kan de student deze SQL queries vanuit Python uitvoeren en in geval van een query, het resultaat in Python uitlezen.

4.11.3 Indeling

Ma 6+7 Hoorcollege “Databases” dat het belang van databases duidelijk maakt, de basisbegrippen uitlegt, alsmede de beginselen van SQL query’s en updates.

Ma 8+9 Instructiepracticum waar de studenten oefenen met SQL query’s en updates op de Twitter database.

Di 1+2 Hoorcollege “Datamodellering” dat ER-modellen uitlegt en interactief voor een paar voorbeeldcasussen een model maakt. Ook legt he uit hoe je zo’n model realiseert in een database mbv CREATE TABLE statements.

Di 3+4 Instructiepracticum waar de studenten oefenen met het maken van een ER-model en die realiseren en vullen met wat data.

Di 8+9 Uitleg benodigd bij uitvoeren opdracht. Start maken met de opdracht.

Wo 6–9 Ruimte voor werken aan het eindproject.

Do 1+2 Ruimte voor werken aan het eindproject.

Do 3+4 Peer-feedbacksessie mbt de opdracht. Daarnaast uitleg en oefenen met uitbreiden ECA-rule engine met extra acties en user-interface componenten. Dit laatste alleen bedoeld voor geïnteresseerden.

Vr 1–4 Toets met daarvoor zelfstudie voor toets.

4.11.4 Opdracht

De opdracht is om tbv het eigen project de database uit te breiden met eigen informatie en bijbehorende uitbreiding van de ECA-rule engine met een event en/of condition dat de gegevens uit die tabel(len) gebruikt. De informatie moet ontworpen zijn met een ER-model en de tabellen moeten corresponderen met dit model.

4.11.5 Docenten

4.11.6 Studiemateriaal

4.11.7 Academische Vaardigheden

Team meeting: Generating Feedback on the project (and especially on G6, G7, G8) (in te plannen aan het einde van de week). Leerdoel:

- reflecteren op de afgelopen project week, zijn de geleerde technieken toegepast?

A Open Issues

- Studiemateriaal / blokboek
- Zwakheid: als je in één week op een onderwerp concentreert en er daarna niet meer op terugkomt, dan beklijft het geleerde mogelijk niet goed genoeg.
- Zijn we niet te ambitieus? Kijk maar eens wat je per week moet doen inclusief wiskunde.
- Leerdoelen en cijfers: leerdoelen zijn minimum-eis? Toetsvragen puur op leerdoelen? Perfecte toets = 10? Of 8 + bonuspunten? Of bonuspunten voor all parels verzamelen voor verhoging module eindcijfer.

B Actiepunten

1: MvK : Bij de opdrachten van specifieke parels (1,2,5?) concurrency-georiënteerde bonusonderdelen opnemen. *7*
2: MvK : Ontwerp voor "Organisatie tijdens module" maken *8*
3: PTdB : Streven is om het "interview een wetenschapper" uit AVI1 te behouden, mogelijk ook "interview met een alumnus", maar waar geven we dat een geschikte plek in de module? *9*
4: PTdB : Leerdoelen toetsbaarder maken, waarmee ook inhoud van toets en werkcollege duidelijker wordt *15*
5: MvK,AP Security (cryptography): reduce ambitions *20*
6: PTdB : Indeling parel 5 verder uitwerken. *25*
7: PTdB : concrete opdracht(en) parel 5 bedenken en uitwerken. *26*
8: PTdB : Mede-docent/ontwerper met meer OS-expertise zoeken *26*