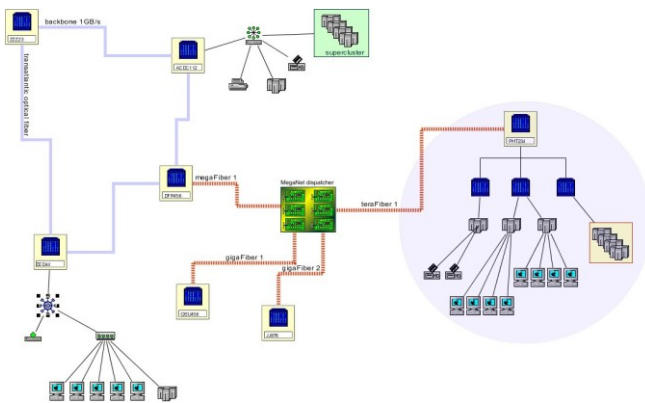


# Distributed Systems Chat (DSC)

*Protocol & Eisen versie 1.7b, 29 maart 2010*



## Inhoudsopgave

- 1) Inleiding
- 2) Netwerkarchitectuur
- 3) De Cliënt
- 4) De Server
- 5) Te verzamelen en gebruiken data
- 6) Berichten
- 7) Cliënt-Server communicatie
- 8) Server communicatie
- 9) Eindnotities

### **(1) Inleiding**

Dit document beschrijft de opdracht voor het practicum gedistribueerde systemen 2009-2010. De opdracht behelst het ontwerpen, bouwen en testen van een server waarmee een backbone van servers voor een gedistribueerd chatsysteem (Distributed Systems Chat, DSC) kan worden gebouwd. Het protocol dat gebruikt dient te worden voor een correcte implementatie van een DSC server vormt hier een belangrijk onderdeel van. De DSC server is een programma dat tijdens het college distributed systems dient te worden ontwikkeld door teams van practicanten.

Voor het programmeren van de server is geen programmeertaal of platform voorgeschreven, deze wordt in overleg met de assistent bepaald. Wel moeten alle ontwikkelde servers en cliënten in een netwerk kunnen samenwerken.

**Samenwerking.** De opdracht moet met groepjes van drie of vier studenten worden gemaakt, waarbij natuurlijk onderling goed en controleerbaar moet worden samengewerkt. In verband met deze controleerbaarheid is aanwezigheid op het practicum verplicht.

**Ontwerp.** Hoewel hieronder een vrij ver uitgewerkt protocol staat, moet er nog goed worden nagedacht over de afhandeling van de diverse boodschappen zodat een robuust en betrouwbaar systeem ontstaat. Omdat ook wordt geëist dat de gebouwde server kan samenwerken met die die worden gebouwd door de andere groepen, moet het protocol nauwgezet worden geïmplementeerd. Een document met een uitgewerkte analyse van de werking van het protocol en een eerste ontwerp van je server moet aan de assistent worden voorgelegd voordat je begint met programmeren. Je mag pas in de tweede week beginnen met programmeren – ook komt de controlserver (zie onder) pas dan on-line.

**Eindproduct.** Het eindproduct dat je zal moeten leveren bestaat uit de volgende onderdelen:

- Een demonstratie van de werking van je server op het practicum op woensdag 12 mei.
- Een verslag, waarin het ontwerpdocument, aangepast op basis van je ervaring bij de implementatie, tevens een korte handleiding en een verslag van de wijze waarop je server is getest. Inleveren op of voor 16 mei via blackboard.
- De goed gedocumenteerde code. Inleveren op of voor 16 mei via blackboard.

## ***(2) Netwerkarchitectuur***

Tijdens dit practicum dient een gedistribueerd communicatiemedium gebouwd te worden. Dit houdt in dat servers en cliënten die op verschillende platforms worden ontwikkeld moeten communiceren en samenwerken. De cliënten dienen maar met één server tegelijk actief verbinding te maken. Passieve verbindingen zijn altijd mogelijk. De servers moeten aan elkaar gelinkt kunnen worden in een samenhangende boom (dus zonder cyclen). Omdat we in ons protocol niet zullen bijhouden langs welke servers een boodschap al is gekomen, zou een lus in de topologie al gauw tot gevolg hebben dat boodschappen langs die lus blijven rondreizen. Met welk adres (welke reeds aanwezige server) een nieuwe server actief verbinding maakt wordt bepaald door de control server, zie ook verderop.

## ***(3) De cliënt***

De cliënt voor dit practicum hoeft niet zelf te worden geschreven, maar is reeds door de practicumleiding ontwikkeld. Er zijn 2 smaken van de cliënt: een nogal buggy ncurses cliënt en een goede grafische cliënt. Beiden werken onder linux, er zijn geen windows cliënten. De cliënten zijn te vinden in /opt/stud/opsys op sremote.

## ***(4) De Server***

De server die door de practicanten dient te worden ontwikkeld, moet aan een aantal eisen voldoen:

- De servers moeten elk verbindingen van meerdere cliënten en meerdere servers toestaan
- De servers moeten berichten kunnen broadcasten naar alle servers en unicasten per verbonden cliënt.
- De servers moeten unicast berichten naar cliënten die niet aan hemzelf verbonden zijn, kunnen relays/doorsturen naar de juiste andere server

- De interne database van aan het netwerk verbonden cliënten en servers dient consistent te zijn over alle servers, waarbij aangetekend wordt dat niet alle servers dezelfde informatie hebben.
- De servers dienen te voldoen aan het verderop beschreven protocol, uitbreidingen zijn niet verboden, maar worden afgeraden ivm consistentie
- De servers dienen een configuratiebestand in te kunnen lezen waarin staat:
  - De poort waarop geluisterd moet worden (standaard opsys-net)
  - De namen van de beheerders van de server, inclusief wachtwoord (cleartext)
  - Het adres met informatie van de controlserver
  - De door deze server te gebruiken identificatie
  - Eventuele uitbreidingen....Het formaat van dit configuratiebestand mag zelf worden gekozen en hoeft niet uniform te zijn over alle servers.
- De server mag niet interactief via de console worden gebruikt, alles dient via het configuratiebestand of via commando's uit de cliënten te geschieden. Ook mag er niets naar stdout/stdin geschreven worden, maar dient er een logfile aangemaakt te worden.
- De server mag zelf maar actief naar één andere server linken, maar er mogen meer servers naar dezelfde linken

### ***(5) Te verzamelen en gebruiken data***

Elke server dient tenminste de volgende data bij te houden voor een goede werking van het protocol:

- Voor elke direct aan deze server verbonden server tenminste het socketnummer van die server
- Voor alle andere servers in het systeem: een gedeeltelijke route naar die server.
- Voor elke cliënt tenminste
  - De autorisatiestatus van de cliënt
  - Als die cliënt direct met de huidige server is verbonden: het socketnummer van de cliënt
  - In andere gevallen: de server via welke de cliënt te bereiken is

### ***(6) Berichten***

De uitwisseling van informatie en berichten tussen servers en cliënten alsmede tussen servers onderling geschiedt via berichten volgens een vast protocol. Hiervan mag niet afgeweken worden, daar dit communicatie tussen verschillende implementaties moeilijk of onmogelijk maakt.

#### ***(6.1) Berichtformaat***

De berichten hebben het volgende formaat

- De eerste 16 bits zijn een integer die de lengte van het record inclusief deze 2 bytes aangeeft
- De volgende 2 bytes zijn een integer die het type bericht aangeeft
- De 2 voornoemde integers dienen in network byte order verzonden te worden

- Daarna volgt de rest van het bericht, een bericht is maximaal 200 bytes lang, inclusief de eerste 4 bytes. Deze bytes moeten allemaal “human-readable” ascii karakters bevatten. Dus: letters, cijfers, spaties, en de tekens `~!@#\$\$%^&\*() - \_+=+[{]};:'"\|,<.>/?`
- Een bericht wordt niet afgesloten met een bepaald karakter. Een ‘\0’ of ‘\n’ aan het einde van een bericht is expliciet niet toegestaan.

**Fouten:** Bij ontvangst van foute data mag onmiddellijk de socketverbinding worden verbroken.

**Woord:** Een woord zoals verderop gebruikt, is een reeks tekens gescheiden door één of meerdere spaties en/of tabs.

**Naam:** Een naam voor een cliënt mag alleen bestaan uit letters, cijfers en underscores. De maximale lengte van een naam is 12 tekens.

**Serverident:** Een server identificeert zich met een combinatie van zijn ipadres en de poort waarop hij luistert aan de controlserver. Voorbeeld: 146.50.18.121:2000 Aan andere servers zal een server zich identificeren dmv een cookie of tag die van de server ontvangen wordt. Dit cookie is een reeks van 32 bytes. De rol van de tag is ervoor te zorgen dat een herstarte server met hetzelfde ipadres en poortnummer toch als een andere server kan worden herkend. De server moet deze tag dus zelf genereren en wel zo dat hij uniek is. Gebruik als ingrediënten b.v. ipadres, poortnummer en de tijd tot op microseconden. Je kan b.v. ook gebruik maken van /dev/random of /dev/urandom Het kan ook handig zijn een voor jezelf herkenbaar stukje tekst op te nemen – dan is jouw server herkenbaarder in de log van de control server.

## **(6.2) Berichttypen**

Hieronder volgt een lijst met de mogelijke berichttypen, tussen welke partijen deze berichten verzonden worden en de data die ze verder bevatten.

*Berichttype:* Registreer cliënt  
*Typenummer:* 100  
*Tussen:* Cliënt – server  
*Parameters:* naam [wachtwoord]

De rest van het bericht dient 1 of 2 woorden te bevatten: het eerste is de naam van de cliënt, het tweede woord een eventueel wachtwoord. De naam moet uniek zijn voor het hele systeem en de server zal dan ook pas een “registratie gelukt” bericht sturen als dit zo is.

*Berichttype:* Cliënt toegevoegd  
*Typenummer:* 110  
*Tussen:* Server - server en server - cliënt  
*Parameters:* naam

De rest van het bericht bestaat uit slechts de naam van de toegevoegde cliënt. Aan de toegevoegde cliënt of server worden in een serie berichten van dit type alle namen van verbonden cliënten weergegeven.

*Berichttype:* Cliënt verwijderen  
*Typenummer:* 120  
*Tussen:* Cliënt-server  
*Parameters:* woord\*

Dit bericht wordt verzonden als de cliënt-server verbinding verbroken wordt. De rest van het bericht bevat een eventuele tekst die naar iedereen als groet verzonden wordt. Mocht de cliënt deze niet zelf invullen of wordt de verbinding onverwachts verbroken dan dient de server dit zelf in te vullen

*Berichttype:* Cliënt verwijderen  
*Typenummer:* 130  
*Tussen:* Server-server en server-client  
*Parameters:* woord\*

Dit bericht wordt verzonden als de cliënt-server verbinding verbroken is. De rest van het bericht bevat de naam van de verwijderde cliënt en een tekst die naar iedereen als groet verzonden wordt. Mocht de cliënt deze niet zelf invullen of wordt de verbinding onverwachts verbroken dan dient de server dit zelf in te vullen

*Berichttype:* Echo request (ping)  
*Typenummer:* 140  
*Tussen:* Server-server, cliënt-server en server-client  
*Parameters:* identificatie

De rest van het bericht dient 1 woord te bevatten: een identificatienummer / tekenreeks. Het antwoord op dit bericht dient binnen 2 seconden verstuurd te zijn, anders mag de verbinding verbroken worden.

*Berichttype:* Echo response (pong)  
*Typenummer:* 150  
*Tussen:* Server-server, server-client en cliënt-server  
*Parameters:* identificatie

De rest van het bericht dient 1 woord te bevatten: het identificatienummer / tekenreeks van het voorafgaande Echo Request. Bericht dient binnen 2 seconden na het ontvangen van een ping verstuurd te zijn, anders mag de verbinding verbroken worden.

*Berichttype:* Wijzig naam  
*Typenummer:* 160  
*Tussen:* Cliënt-server  
*Parameters:* newname

Dit bericht wordt verzonden als een cliënt zijn naam wijzigt. Het eerste en enige woord van het bericht is de nieuwe naam. De cliënt mag pas aannemen dat het gelukt is als een correcte statusmelding is ontvangen.

*Berichttype:* Wijzig naam  
*Typenummer:* 170  
*Tussen:* Server-server, server-client  
*Parameters:* oldname newname

Dit bericht wordt aan alle aangesloten cliënten verzonden als een cliënt zijn naam heeft gewijzigd. Het eerste woord in het datadeel is de oude naam, het tweede de nieuwe naam.

*Berichttype:* Tekstbericht  
*Typenummer:* 200  
*Tussen:* Cliënt-server  
*Parameters:* bestemming woord\*

De rest van het bericht bestaat uit 2 delen: het eerste woord is de naam van degene aan wie het bericht gericht is of de tekst #all als het bericht naar iedereen gaat. De rest is de door de persoon ingevoerde tekst.

*Berichttype:* Actiebericht  
*Typenummer:* 210  
*Tussen:* Cliënt-server  
*Parameters:* bestemming woord\*

De rest van het bericht bestaat uit 2 delen: het eerste woord is de naam van degene aan wie het bericht gericht is of de tekst #all als het bericht naar iedereen gaat. De rest is de door de persoon ingevoerde tekst. Voor de servers is er geen verschil in behandeling van tekstberichten en actieberichten.

*Berichttype:* Tekstbericht  
*Typenummer:* 300  
*Tussen:* Server-server, server-cliënt  
*Parameters:* bron bestemming woord\*

De rest van het bericht bestaat uit 3 delen: het eerste woord is de naam van de afzender, het tweede woord is de naam van degene aan wie het bericht gericht is of de tekst #all als het bericht naar iedereen gaat. De rest is de door de persoon ingevoerde tekst.

*Berichttype:* Actiebericht  
*Typenummer:* 310  
*Tussen:* Server-server, server-cliënt  
*Parameters:* bron bestemming woord\*

De rest van het bericht bestaat uit 3 delen: het eerste woord is de naam van de afzender, het tweede woord is de naam van degene aan wie het bericht gericht is of de tekst #all als het bericht naar iedereen gaat. De rest is de door de persoon ingevoerde tekst.

*Berichttype:* Registratie gelukt  
*Typenummer:* 500  
*Tussen:* Server-cliënt  
*Parameters:* (geen)

Wordt verzonden als de registratie gelukt is, de rest van het bericht kan leeg zijn.

*Berichttype:* Registratie mislukt  
*Typenummer:* 510  
*Tussen:* Server-cliënt  
*Parameters:* reden (=woord woord\*)

Wordt verzonden als de registratie mislukt is. De rest van het bericht bevat de reden van het mislukken van de inlogpoging.

*Berichttype:* Naamwijziging gelukt  
*Typenummer:* 520  
*Tussen:* Server-cliënt  
*Parameters:* (geen)

Wordt verzonden als de naamswijziging gelukt is, de rest van het bericht kan leeg zijn.

*Berichttype:* Naamwijziging mislukt  
*Typenummer:* 530  
*Tussen:* Server-cliënt  
*Parameters:* reden (=woord woord\*)

Wordt verzonden als de naamswijziging mislukt is. De rest van het bericht bevat de reden van het mislukken van de poging.

*Berichttype:* Registreer server  
*Typenummer:* 600  
*Tussen:* Server-server  
*Parameters:* ipadres en poortnummer (in een woord), gevolgd door een dubbele punt en de eigen tag.

Dit verstuurt een server om zich aan te melden bij een andere server. Het is dus de server-server versie van berichttype 100.

*Berichttype:* Adres server  
*Typenummer:* 601  
*Tussen:* Server-control server  
*Parameters:* ipadres en poortnummer (in een woord), gevolgd door een dubbele punt en de eigen tag.

Dit verstuurt een server bij aanmelding naar de control server.

*Berichttype:* Adres server  
*Typenummer:* 602  
*Tussen:* Control server- server  
*Parameters:* ipadres, poortnummer en tag van de toegewezen parent(in een woord) of “none”

Dit verstuurt een control server bij aanmelding naar de nieuwe server. Parameter is of een serverident en tag, gescheiden door een dubbele punt, of de tekenreeks “none” (zonder “”). Voorbeeld:

146.50.19.123:2000:7a788f56fa49ae0ba5ebde780efe4d6a89b5db47

Deze tag geeft dus een unieke identificatie van de “parent” server.

*Berichttype:* Peer verbinding verbroken  
*Typenummer:* 603  
*Tussen:* Server-control server  
*Parameters:* Tag van de uitgevallen verbinding (in een woord)

Dit verstuurt een server naar de control server als een van zijn partner servers niet reageert. Dat kan dus de parent of een van de kinderen zijn.

*Berichttype:* Hergroeppeer  
*Typenummer:* 604  
*Tussen:* Control server- server  
*Parameters:* tag van de uitgevallen server (in een woord) en eventueel een tweede woord equivalent aan de parameter van een 602 bericht.

Deze wordt door de controlserver verstuurd als een peer niet meer reageert. Wanneer dit bericht ontvangen wordt dienen alle cliënten die via de dode server verbonden zijn afgemeld te worden. Als de uitgevallen node de huidige parent is zal er een tweede woord aanwezig zijn wat de nieuwe parent is. Met deze zal verbinding gemaakt moeten worden om het netwerk te herstellen.

*Berichttype:* Adres server  
*Typenummer:* 610  
*Tussen:* Cliënt-control server  
*Parameters:* geen

Dit verstuurt een cliënt bij aanmelding naar de control server om een serveradres op te vragen.

*Berichttype:* Adres server  
*Typenummer:* 611  
*Tussen:* Control server- cliënt  
*Parameters:* ipadres en poortnummer toegewezen server (in een woord) of “none”

Antwoord op een 610. Dit verstuurt de control server bij aanmelding naar een cliënt. Parameter is een serveridentificatie zonder tag. Als er geen servers bekend zijn, wordt er noodgedwongen geen adres opgestuurd, maar “none”.

*Berichttype:* Stop server  
*Typenummer:* 700  
*Tussen:* Cliënt-server  
*Parameters:* geen

De berichttypen 700-799 zijn gereserveerd voor uitbreidingen die server-specifiek zijn. Alleen het commando 700 dient gebruikt te worden om een server te kunnen stoppen. Dit commando mag uiteraard alleen door beheerders gegeven worden. Beheerders identificeren zichzelf tijdens het aanmelden of een naamswijziging. Daarom is hier geen extra authenticatie nodig.

### ***(7) Cliënt-Server communicatie***

Een cliënt maakt verbinding met de server door een TCP/IP verbinding te maken naar het ip-adres van de server. De poort waarop de server luistert staat niet vast, maar veelal zal opsys-res (2001) worden gebruikt. De cliënt kan eventueel aan deze gegevens komen door verbinding te maken met de control server en die na het ontvangen van een ip adres weer onmiddellijk te verbreken (boodschappen 610 en 611).

Zodra de verbinding is gemaakt verstuurt de cliënt een loginbericht, de server verstuurt daarop een statusmelding (OK, naam onjuist, wachtwoord onjuist) en indien OK zal hij de lijst met namen sturen. Ook zal de server de naam van de nieuwe cliënt verspreiden over het netwerk. Als het na 3 loginpogingen of 15 seconden nog niet is gelukt om in te loggen zal de server de verbinding verbreken.

Vervolgens zal er een reeks berichten uitgewisseld worden van de types ‘bericht’, ‘actie’ en ‘naamswijziging’. De server zal deze naar alle aangesloten cliënten versturen, inclusief de zendende cliënt.

Bij wijziging van namen zal de server dat aan alle aangesloten cliënten/servers melden. Deze dienen daarop hun namenlijst aan te passen. Als een naamswijziging niet correct is (er is al iemand met die naam bijvoorbeeld) zal de server een errorcode terugsturen. Ook is het mogelijk om administratieve handelingen te doen zoals het afsluiten van de server via een bepaald commando. Verdere uitbreidingen staan vrij.

De sessie wordt normaliter beëindigd met een quit commando, welke de server naar alle cliënten doorstuurt. Als een cliënt of server meer dan 30 seconden stil blijft dient er een ping verzonden te worden. Wordt deze niet binnen 2 seconden beantwoordt, dan is de cliënt niet meer beschikbaar en mag de verbinding verbroken worden. Ook bij foutief verzonden data mag de verbinding worden verbroken. Als de server zelf de verbinding verbreekt moet er wel naar alle cliënten en servers een ‘quit’ bericht worden verzonden. Daarin staat de reden van het verbreken.

### ***(8) Server-Server communicatie***

De servers vinden elkaar uiteraard niet vanzelf. Hiervoor is de control server. Deze draait op een nader op te geven machine op poort opsys-res (2001). Een server dient hiermee verbinding te maken en een paar 601 en 602 berichten uit te wisselen. Zodra de verbinding tot stand is gebracht zal de control server een adres van een reeds actieve server geven. Dit adres zal een ip4-adres zijn in ascii tekens, gevolgd door een poortnummer en een tag (zie 6.1), bijvoorbeeld “146.50.9.13:2000:7a788f56fa49ae0ba5ebde780efe4d6a89b5db47”. Als de server de string “none” teruggeeft is er nog geen server actief en blijf je standalone. De verbinding met de control server dient in stand te blijven zolang de server online is. Elke 10 seconden zal er gepingd worden volgens het eerder genoemde protocol. Andere data mag niet verzonden worden.

Een server meldt zich niet aan met een code 100, maar met een code 600. We noemen de server waarbij hij zich aanmeldt de “parent” van een server. Als een server zich aanmeldt zal hij van zijn parent een lijst met namen ontvangen. Hij moet dan de namen doorsturen naar alle reeds verbonden cliënten en servers (alleen van belang bij heraanmelding). Mocht hij al cliënten hebben met dezelfde naam, dan dient hij deze te verwijderen en de verbindingen te verbreken.

Vervolgens stuurt hij zelf een lijst met de namen met aan of via hem verbonden cliënten aan de parent. Ook deze stuurt de namen door naar alle verbonden cliënten en servers. De communicatie tussen 2 servers lijkt erg op die tussen server en cliënt, met het verschil dat er veel berichten een ander typenummer en andere parameters hebben.

De berichten die voor cliënten bestemd zijn die niet direkt met de huidige server verbonden zijn, moeten allen naar de server verzonden worden via welke de cliënt te bereiken is.

Als voor een server de verbinding met zijn parent wordt verbroken (door een fout of door het afmelden van de parent), dan meldt hij zich als volg opnieuw aan bij het systeem:

- Hij markeert de informatie over alle via de parent te bereiken cliënten en servers als ongeldig.

- Hij maakt een nieuwe verbinding met de control server om een nieuwe parent toegewezen te krijgen (zoals boven beschreven).
- Hij verbreekt daarna onmiddellijk de oude verbinding met de control server.
- Hij meldt zich eventueel aan bij de nieuw toegewezen parent volgens eerder genoemd protocol, waarbij servers en cliënten weer als geldig gemarkeerd kunnen worden.
- Hij stuurt naar alle van hem afhankelijke cliënten en servers een quit bericht voor de cliënten en servers die daarna nog als ongeldig gemarkeerd overblijven.
- Hij stuurt informatie over nog onbekende deelnemers door naar alle van hem afhankelijke deelnemers.

Als de verbinding met een child server wegvalt, dan stuurt een server een quit bericht voor alle langs die weg te bereiken (nu onbereikbare) cliënten en servers aan de nog wel verbonden partijen.

### ***(9) Eindnotities***

Zoals alles in het leven staat niets vast: het protocol zal aan veranderingen onderhevig zijn, ook tijdens de looptijd van het practicum. De te ontwikkelen programma's dienen te voldoen aan de laatst bekende versie van het protocol. Mocht iemand onvolkomenheden / fouten in het protocol vinden, dan zal dit zeker aangepast worden.. Voor alle vragen mbt het practicum kun je je wenden tot de assistent. In het cursusjaar 2009-2010 is dat Roy Bakker. Hij is te bereiken via mail([bakkerr@science.uva.nl](mailto:bakkerr@science.uva.nl)).

De practicumssessies zijn voor iedereen verplicht, niet aanwezig betekent dat je geen cijfer krijgt voor het practicum. Wel kan in overleg met de assistent voor teams een andere tijd gekozen worden. De beoordeling van de resultaten ligt in handen van de assistent. Deze zal kijken naar o.a. inzet, voortgang, correctheid en volledigheid. Het eindresultaat zal een cijfer zijn op de schaal van 0 t/m 10. Het cijfer moet voldoende zijn, en telt dan voor 50% mee in het eindcijfer.

#### **Belangrijke mededeling:**

De berichten die servers naar cliënten versturen moeten **ook** verstuurd worden naar de cliënt die ze veroorzaakt. Een cliënt mag er ook niet van uitgaan dat een bericht goed ontvangen is voordat hij hem terugkrijgt.

*Maart 2010*

*Dick van Albada ([dick@science.uva.nl](mailto:dick@science.uva.nl))*

*Roy Bakker ([bakkerr@science.uva.nl](mailto:bakkerr@science.uva.nl))*