

Project Logisch Programmeren

Thomas Raes
traes@vub.ac.be
75972

15 januari 2006

Samenvatting

Verslag project Logisch Programmeren academiejaar 2005-2006 aan de Vrije Universiteit Brussel.

Inhoudsopgave

1	Inleiding	1
2	Gebruik	1
2.1	Starten	1
2.2	Personen	2
2.3	Groepen	2
2.4	Tijd	2
2.5	Voorkeuren	3
2.6	Reschedulen	3
2.7	Database	4
2.8	Testen	4
3	Implementatie	4
3.1	Bestanden	4
3.2	Multithreading	5
3.2.1	Synchronisatie	5
3.2.2	Communicatie	5
3.3	Human Language Interface	6
3.4	Afspraken	6
3.5	Scheduler	6
3.5.1	Constraint conflicten	6
3.5.2	Constraint schendingen	6
4	Besluit	7
A	Voorbeeld	7

1 Inleiding

Het project is een Multi-Organizer en regelt afspraken tussen verschillende personen. De interface van het programma is een Human Language interpreter. Het programma kan zowel als single-user programma en als server gebruikt worden.

In dit verslag wordt eerst de functionaliteit uitgelegd aan de hand van voorbeelden. Vervolgens wordt de implementatie besproken. Tenslotte volgt er een volledige voorbeeldinteractie met het programma.

2 Gebruik

2.1 Starten

Verschillende mensen kunnen samenwerken als men het programma als server start. Dit doet men met het script *start-server*. Gebruikers connecteren via telnet en krijgen vervolgens een command prompt.

```
telnet localhost 5000
Trying 127.0.0.1...
Connected to HAL9000.
Escape character is '^]'.
=== Multi Organizer Server ===
#:
```

Als men geen netwerk wil gebruiken, kan men het programma ook starten in single-user mode. Dit doet men met het script *start-single-user*. De server en single-user mode bieden dezelfde functionaliteit.

2.2 Personen

Het programma maakt onderscheid tussen interne en externe personen. Men kan afspraken maken waar zowel interne als externe personen aan deelnemen. Bij interne personen wordt gecontroleerd of ze wel degelijk bestaan.

Men kan de namen van alle interne personen opvragen met *show intern persons*. Nieuwe interne personen kan men aanmaken met *make intern person*. Enkel interne personen moeten eerst aangemaakt worden. Voor de naam van externe personen moet *extern* staan.

```
#:make intern person alice
#:make intern person bob
#:make appointment between alice bob extern alfred as soon as possible
```

2.3 Groepen

Men kan personen samenbrengen in groepen, waarbij personen tot meerdere groepen tegelijk kunnen behoren. Bij het aanmaken van afspraken kan men dan groepsnaam gebruiken en hoeft men niet alle namen afzonderlijk in te geven. De groepsnaam moet dan voorafgegaan worden door *group*.

Men kan personen toevoegen aan en verwijderen uit groepen. Groepen worden automatisch aangemaakt vanaf ze leden hebben. Men kan groepen verwijderen met het commando *delete group*.

```
#:add bob charlie alice to group sales
#:remove charlie from group sales
#:show group sales
alice
bob
#:make appointment between group sales extern customer
as soon as possible
#:delete group sales
```

2.4 Tijd

Datums worden ingegeven met de dag als nummer, de naam van de maand en vervolgens het jaar. Het programma kent automatisch de dag van de week met behulp van het Doomsday algoritme [1].

Men kan de huidige tijd opvragen met het commando *show current time*:

```
#:show current time
thursday 5 january 2006 at 19:30
```

De kleinst mogelijke duur van een afspraak is 30 minuten. Alle tijdstippen worden bijgevolg afgerond tot veelvoud van 30 minuten.

De tijdspanne van een afspraak wordt bepaald door het beginmoment en het eindmoment of een vermelding van de lengte. Als men deze informatie niet expliciet meedeelt, gebruikt het programma de huidige tijd als beginmoment en 30 minuten als de duur van de afspraak.

```
#:make appointment between alice bob as soon as possible
#:make appointment between alice bob on 5 january 2006 at 21 : 00
with duration 2 hours and 30 minutes
#:make appointment between alice bob from 5 january 2006 at 19 : 30
till 5 january 2006 at 20 : 00
```

2.5 Voorkeuren

Gebruikers kunnen voorkeuren bepalen waaraan een bepaalde afspraak moet voldoen. Er kunnen 0, 1 of meerdere voorkeuren opgelegd worden aan elke afspraak.

```
#:make appointment between group sales as soon as possible
constrained on a holiday during office hours not on a weekend
```

De volgende constraints zijn beschikbaar:

- **(not) on a holiday** (niet) op een feestdag
- **(not) on a weekend** (niet) op zaterdag of zondag
- **(not) on a weekday** omgekeerde van weekend
- **begin before** begin voor een gegeven moment
- **begin after** begin na een gegeven moment
- **later than** begin na een bepaald uur
- **earlier than** begin voor een bepaald uur
- **(not) during office hours** (niet) tussen 09u00 en 17u00

2.6 Reschedulen

Als het niet mogelijk is om een afspraak aan de vereiste voorkeuren te laten voldoen, zoekt de scheduler samen met de gebruiker naar een alternatief. De scheduler zoekt een andere afspraak die verschoven kan worden. Indien er een dergelijke afspraak gevonden wordt, wordt er aan de gebruiker gevraagd of die afspraak verschoven mag worden. Indien de gebruiker akkoord gaat, wordt de oorspronkelijke afspraak verwijderd en wordt de nieuwe afspraak vastgelegd. Vervolgens gaat de scheduler op zoek naar een nieuwe plaats voor de oorspronkelijke afspraak. Hierdoor kan het zijn dat de scheduler nog een te verschuiven afspraak voorstelt. Als de gebruiker geen toestemming geeft om een andere afspraak te verplaatsen,

kan de afspraak niet vastgelegd worden. In dit geval wordt de afspraak afgelast.

Voorbeeld van het reschedulen:

```
#:show appointments
Appointment between alice and bob from sunday 8 january 2006 at 19:00
till sunday 8 january 2006 at 19:30 with 0 constraints
Appointment between alice and bob from sunday 8 january 2006 at 19:30
till sunday 8 january 2006 at 20:00 with 0 constraints
Appointment between alice and bob from sunday 8 january 2006 at 20:00
till sunday 8 january 2006 at 20:30 with 0 constraints
Appointment between alice and bob from sunday 8 january 2006 at 20:30
till sunday 8 january 2006 at 21:00 with 0 constraints
Appointment between alice and bob from sunday 8 january 2006 at 21:00
till sunday 8 january 2006 at 21:30 with 0 constraints
Appointment between alice and bob from sunday 8 january 2006 at 21:30
till sunday 8 january 2006 at 22:00 with 0 constraints
#:make appointment between alice dave as soon as possible constrained
begin before 8 january 2006 at 20 : 00
'can I reschedule:'
Appointment between alice and bob from sunday 8 january 2006 at 19:00
till sunday 8 january 2006 at 19:30 with 0 constraints
'yes/no'yes.
#:show appointments
Appointment between alice and bob from sunday 8 january 2006 at 19:30
till sunday 8 january 2006 at 20:00 with 0 constraints
Appointment between alice and bob from sunday 8 january 2006 at 20:00
till sunday 8 january 2006 at 20:30 with 0 constraints
Appointment between alice and bob from sunday 8 january 2006 at 20:30
till sunday 8 january 2006 at 21:00 with 0 constraints
Appointment between alice and bob from sunday 8 january 2006 at 21:00
till sunday 8 january 2006 at 21:30 with 0 constraints
Appointment between alice and bob from sunday 8 january 2006 at 21:30
till sunday 8 january 2006 at 22:00 with 0 constraints
Appointment between alice and dave from sunday 8 january 2006 at 19:00
till sunday 8 january 2006 at 19:30 with 1 constraints
Appointment between alice and bob from sunday 8 january 2006 at 22:00
till sunday 8 january 2006 at 22:30 with 0 constraints
```

2.7 Database

De database bevat alle personen, groepen en afspraken. Men kan deze informatie opslagen in en laden uit een bestand:

```
#:load database from db1
#:save database as backup
```

2.8 Testen

Er zijn enkele voorzieningen om het testen en debuggen te vereenvoudigen. Met de commandos *enable debug* en *disable debug* kan men het programma debug informatie laten weergeven.

3 Implementatie

3.1 Bestanden

- **project.pl** code ivm opstarten programma
- **input.pl** code ivm ontvangen en parsen gegevens
- **output.pl** code ivm uitvoer van gegevens
- **time.pl** code ivm tijd
- **holidays.pl** code ivm feestdagen
- **appointments.pl** code ivm afspraken
- **constraints.pl** voorstelling en controle constraints
- **persons.pl** code ivm personen
- **scheduling.pl** scheduling algoritme
- **server.pl** code ivm server en sockets
- **start-server** script dat programma als server start
- **start-single-user** script dat programma single-user start
- **db1** voorbeeld database

3.2 Multithreading

Threads laten toe dat er meerdere gebruikers tegelijk op het systeem werken. Er moeten wel maatregelen genomen worden om de verschillende threads correct samen te laten werken.

3.2.1 Synchronisatie

Alle threads hebben toegang tot dezelfde database. Als een thread gegevens wijzigt, zijn deze aanpassingen ook zichtbaar voor alle andere threads.

Dit kan problemen veroorzaken als een thread gegevens gebruikt die een andere thread aan het wijzigen is. Het kan bijvoorbeeld zijn dat 2 threads hetzelfde tijdsslot willen reserveren voor verschillende afspraken. De eerste thread controleert de beschikbaarheid van het tijdsslot en gebruikt het vervolgens voor zijn afspraak. Een tweede thread controleert hetzelfde tijdsslot nadat de eerste thread dit gecontroleerd heeft maar voor dat de eerste thread dit tijdsslot effectief gebruikt. Dit heeft als gevolg dat een tijdsslot meerdere keren gebruikt wordt, wat conflicten kan geven als er personen aan beide afspraken deelnemen.

Om de consistentie van de data te garanderen, moeten de verschillende threads dus gesynchroniseerd worden. Dit gebeurt met behulp van locks. Vermits het controleren en verkrijgen van een lock als een atomaire operatie uitgevoerd wordt, kan men met behulp van locks de toegang tot gemeenschappelijke data regelen. Voordat een thread aanpassingen aan de database kan uitvoeren, moet hij in het bezit zijn van de lock. Als een andere thread de lock bezit, wacht de eerste thread totdat de lock terug vrijgegeven wordt. Na het uitvoeren van de operaties op de database geeft de thread op zijn beurt ook de lock vrij.

Bij de oorspronkelijke toepassing van deze methode in de multi-organizer trad er geregeld starvation op. Een thread die een lock vrijgaf en deze later terug aanvroeg, was meestal sneller dan een andere thread die op dezelfde lock aan het wachten was. Hierdoor duurde het lang voordat een

andere thread toegang tot de database verkreeg. Concreet kwam dit er op neer dat de gebruiker die eerst was ingelogd, veel meer aan de beurt kwam dan de andere gebruikers.

Een oplossing is om een actieve thread die een lock vrijgeeft een lagere prioriteit in de scheduler te geven. Vermits SWI Prolog threads implementeert met behulp van native OS-threads, heeft men echter geen controle over de scheduler. Dit probleem werd opgelost door een thread 0.1 seconde te laten wachten (mbv *sleep*) nadat hij een lock vrijgeeft. Op deze manier geeft de scheduler voorrang aan een andere thread.

3.2.2 Communicatie

Wanneer het programma in single-user mode gebruikt wordt, moet alle invoer en uitvoer naar de terminal van de gebruiker gestuurd worden. Als het programma in server mode gebruikt wordt, moet de in- en uitvoer met de juiste netwerksockets verbonden worden. Om dit transparant te laten verlopen heb ik de predikaten *show/1,ask/1* en *newline/0* gemaakt die automatisch de juiste streams gebruiken.

Verschillende threads vertegenwoordigen verschillende sessies. Dit geeft als probleem dat de in- en uitvoer op het scherm van de juiste gebruiker moet terechtkomen. De verschillende threads gebruiken *assert* en *retract* om hun input- en outputstream zichtbaar te maken voor *show/1, ask/1* en *newline/0*. Vermits deze regels globaal zijn, wordt er gebruik gemaakt van het eerder beschreven locking-systeem om de synchronisatie te regelen.

Een groter probleem is dat het SWI Prolog predikaat *readln* niet werkt met netwerk-sockets. Dit predikaat wordt in de single-user versie gebruikt om een hele lijn gebruikersinvoer ineens in te lezen. In de server-versie van het programma, kan men enkel *read* gebruiken om de invoer van een socket stream te lezen. Concreet betekent dit dat de gebruiker zijn invoer als lijst moet ingeven.

Het predikaat *read_line_to_codes* werkt wel goed met netwerkverbindingen. Dit predikaat heeft echter als nadeel dat de verkregen invoer, inclusief spaties, voorgesteld wordt als een lijst van getallen. Deze getallen moeten dus met extra regels omgezet worden naar een lijst van atomen. In tegenstelling tot *readln* houdt *atom_codes* er geen rekening mee dat er ook getallen voorgesteld kunnen worden. Hierdoor waren er aanpassingen nodig in de human language parser om hierop te controleren en zelf de juiste conversies uit te voeren.

3.3 Human Language Interface

De gebruikers communiceren met het programma in het Engels. Deze tekstuele commandos worden vertaald naar prolog code met behulp van een definite clause grammar.

3.4 Afspraken

Afspraken zijn abstract voorgesteld als lijsten. Ze bevatten de deelnemende personen, de constraints die van toepassing zijn, de aangevraagde tijdsspanne en de werkelijke tijdsspanne. De werkelijke tijdsspanne kan nooit vroeger beginnen dan de aangevraagde tijdsspanne. Hiervoor moet de aangevraagde tijdsspanne bijgehouden worden zodat hiermee rekening gehouden kan worden tijdens het scheduleren.

3.5 Scheduler

De scheduler probeert de afspraak zo dicht mogelijk bij de aangevraagde tijdsspanne te laten plaatsvinden. Hierbij houdt hij rekening met de andere afspraken van al de deelnemers, en de eventuele extra constraints.

3.5.1 Constraint conflicten

Gebruikers kunnen meerdere constraints opleggen voor een zelfde afspraak. De afspraak moet dan aan al deze constraints voldoen. In sommige gevallen is dit onmogelijk, bijvoorbeeld als zowel de constraints *on a weekend* en *not on a weekend* opgelegd worden.

Hierbij ontstaat er dus een conflict tussen de constraints van een zelfde afspraak. Als er aan de ene constraint voldaan is, wordt de andere geschonden en omgekeerd. De scheduler controleert daarom afspraken eerst op dergelijke conflicten alvorens ze te proberen schedulen. Indien een conflict gevonden wordt, wordt de gebruiker gewaarschuwd en stopt de scheduler.

3.5.2 Constraint schendingen

Constraints kunnen wanneer ze geschonden worden een betere tijdsspanne voorstellen. Op deze manier moet de scheduler minder mogelijkheden onderzoeken en werkt het programma sneller.

Als er voor een afspraak geen geschikte tijdsspanne gevonden kan worden, herschikt de scheduler reeds eerder gemaakte afspraken. Dit kan gebeuren wanneer er een afspraak moet plaatsvinden tijdens een periode die al helemaal volzet is. Tijdens het herschedulen worden er voor alle betrokken afspraken oplossingen gezocht die zo dicht mogelijk bij de oorspronkelijk aangevraagde tijdsspanne liggen. De scheduler vraagt altijd eerst toestemming voordat er een reeds eerder gemaakte afspraak verschoven wordt.

4 Besluit

Het oplossen van scheduling-problemen en het parsen van tekst gaat vlot in Prolog. Threads veroorzaken soms moeilijkheden die niet eenvoudig op te sporen zijn, maar dit is geen Prolog-specifiek probleem.

A Voorbeeld

In het volgende voorbeeld logt een gebruiker in, en maakt vervolgens enkele groepen en afspraken. Vervolgens log een andere gebruiker in een bekijkt de gemaakte afspraken. De database met deze afspraken is beschikbaar als het bestand *testdb*.

```
Trying 127.0.0.1...
Connected to HAL9000.
Escape character is '^]'.
=== Multi Organizer Server ===
#:make intern person alice
#:make intern person bob
#:make intern person carol
#:make intern person dave
```

```
#:make intern person eve
#:make intern person charlie
#:make intern person lucy
#:make intern person linus
#:make intern person sally
#:make intern person marcie
#:make intern person mark
#:make intern person jimmy
#:make intern person alex
#:make intern person sid
#:add alice bob carol to group research
#:add carol dave eve to group management
#:add lucy linus sally marcie to group customers
#:add mark jimmy alex sid to group staff
#:make appointment between group research group management
group customers group staff as soon as possible
#:make appointment between group research group management
as soon as possible
#:make appointment between group research group customers
as soon as possible
#:make appointment between group research group staff
as soon as possible
#:make appointment between group research group management
as soon as possible
#:make appointment between group research group management
group customers group staff as soon as possible
#:make appointment between group research group management
as soon as possible
#:make appointment between group research group management
as soon as possible
#:make appointment between group research group customers
as soon as possible
#:make appointment between group research group staff
as soon as possible
#:make appointment between group research group management
group customers group staff as soon as possible
#:make appointment between group research group customers
as soon as possible
#:make appointment between group research group staff
as soon as possible
#:make appointment between group research group customers
as soon as possible
#:make appointment between group research group management
group customers group staff as soon as possible
#:make appointment between group research group management
group customers group staff as soon as possible
#:make appointment between group research group management
as soon as possible constrained on a weekend
#:make appointment between group research group customers
as soon as possible constrained on a holiday
#:make appointment between group research group staff
as soon as possible constrained not on a weekend
#:make appointment between alice bob carol from 17 january 2006 at 14 : 00
till 17 january 2006 at 16 : 00
```



```
#:make appointment between alice bob carol from 17 january 2006 at 14 : 00
till 17 january 2006 at 16 : 00
#:make appointment between alice bob carol from 17 january 2006 at 14 : 00
till 17 january 2006 at 16 : 00
#:make appointment between alice bob carol from 17 january 2006 at 14 : 00
till 17 january 2006 at 16 : 00
#:make appointment between alice bob carol from 17 january 2006 at 14 : 00
till 17 january 2006 at 16 : 00
#:make appointment between alice bob carol from 17 january 2006 at 14 : 00
till 17 january 2006 at 16 : 00
#:save database as testdb
#:stop
Goodbye!
```

Een andere gebruiker die inlogt, kan de aangemaakte personen en afspraken bekijken.

```
Trying 127.0.0.1...
Connected to HAL9000.
Escape character is '^]'.
=== Multi Organizer Server ===
#:show intern persons
alice
bob
carol
dave
eve
charlie
lucy
linus
sally
marcie
mark
jimmy
alex
sid
#:show appointments
Appointment between carol, bob, alice, eve, dave, carol, marcie, sally, linus,
lucy, sid, alex, jimmy and mark from sunday 15 january 2006 at 00:00 till
sunday 15 january 2006 at 00:30 with 0 constraints
Appointment between carol, bob, alice, eve, dave and carol from sunday 15 january
2006 at 00:30 till sunday 15 january 2006 at 01:00 with 0 constraints
Appointment between carol, bob, alice, marcie, sally, linus and lucy from sunday
15 january 2006 at 01:00 till sunday 15 january 2006 at 01:30 with 0 constraints
Appointment between carol, bob, alice, sid, alex, jimmy and mark from sunday 15
january 2006 at 01:30 till sunday 15 january 2006 at 02:00 with 0 constraints
Appointment between carol, bob, alice, eve, dave and carol from sunday 15 january
2006 at 02:00 till sunday 15 january 2006 at 02:30 with 0 constraints
Appointment between carol, bob, alice, eve, dave, carol, marcie, sally, linus,
lucy, sid, alex, jimmy and mark from sunday 15 january 2006 at 02:30 till sunday
15 january 2006 at 03:00 with 0 constraints
Appointment between carol, bob, alice, eve, dave and carol from sunday 15 january
2006 at 03:00 till sunday 15 january 2006 at 03:30 with 0 constraints
Appointment between carol, bob, alice, eve, dave and carol from sunday 15 january 2006
```

at 03:30 till sunday 15 january 2006 at 04:00 with 0 constraints
Appointment between carol, bob, alice, marcie, sally, linus and lucy from sunday 15
january 2006 at 04:00 till sunday 15 january 2006 at 04:30 with 0 constraints
Appointment between carol, bob, alice, sid, alex, jimmy and mark from sunday 15 january 2006
at 04:30 till sunday 15 january 2006 at 05:00 with 0 constraints
Appointment between carol, bob, alice, eve, dave, carol, marcie, sally, linus, lucy,
sid, alex, jimmy and mark from sunday 15 january 2006 at 05:00 till sunday 15
january 2006 at 05:30 with 0 constraints
Appointment between carol, bob, alice, marcie, sally, linus and lucy from sunday
15 january 2006 at 05:30 till sunday 15 january 2006 at 06:00 with 0 constraints
Appointment between carol, bob, alice, sid, alex, jimmy and mark from sunday
15 january 2006 at 06:00 till sunday 15 january 2006 at 06:30 with 0 constraints
Appointment between carol, bob, alice, marcie, sally, linus and lucy from sunday
15 january 2006 at 06:30 till sunday 15 january 2006 at 07:00 with 0 constraints
Appointment between carol, bob, alice, eve, dave, carol, marcie, sally, linus, lucy,
sid, alex, jimmy and mark from sunday 15 january 2006 at 07:00 till sunday
15 january 2006 at 07:30 with 0 constraints
Appointment between carol, bob, alice, marcie, sally, linus and lucy from sunday
15 january 2006 at 05:30 till sunday 15 january 2006 at 06:00 with 0 constraints
Appointment between carol, bob, alice, sid, alex, jimmy and mark from sunday 15
january 2006 at 06:00 till sunday 15 january 2006 at 06:30 with 0 constraints
Appointment between carol, bob, alice, marcie, sally, linus and lucy from
sunday 15 january 2006 at 06:30 till sunday 15 january 2006 at 07:00 with 0 constraints
Appointment between carol, bob, alice, eve, dave, carol, marcie, sally,
linus, lucy, sid, alex, jimmy and mark from sunday 15 january 2006 at 07:00 till
sunday 15 january 2006 at 07:30 with 0 constraints
Appointment between carol, bob, alice, eve, dave, carol, marcie, sally, linus, lucy,
sid, alex, jimmy and mark from sunday 15 january 2006 at 07:30 till sunday
15 january 2006 at 08:00 with 0 constraints
Appointment between carol, bob, alice, eve, dave and carol from sunday 15 january 2006
at 08:00 till sunday 15 january 2006 at 08:30 with 1 constraints
Appointment between carol, bob, alice, marcie, sally, linus and lucy from sunday
16 april 2006 at 00:00 till sunday 16 april 2006 at 00:30 with 1 constraints
Appointment between carol, bob, alice, sid, alex, jimmy and mark from monday
16 january 2006 at 00:00 till monday 16 january 2006 at 00:30 with 1 constraints
Appointment between alice, bob and carol from tuesday 17 january 2006 at 14:00
till tuesday 17 january 2006 at 16:00 with 0 constraints
Appointment between alice, bob and carol from tuesday 17 january 2006 at 16:00
till tuesday 17 january 2006 at 18:00 with 0 constraints
Appointment between alice, bob and carol from tuesday 17 january 2006 at 18:00
till tuesday 17 january 2006 at 20:00 with 0 constraints
Appointment between alice, bob and carol from tuesday 17 january 2006 at 20:00
till tuesday 17 january 2006 at 22:00 with 0 constraints
Appointment between alice, bob and carol from tuesday 17 january 2006 at 22:00
till wednesday 18 january 2006 at 00:00 with 0 constraints
Appointment between alice, bob and carol from wednesday 18 january 2006 at 00:00
till wednesday 18 january 2006 at 02:00 with 0 constraints
#:stop
Goodbye!
Connection closed by foreign host.

Referenties

- [1] Doomsday Algorithm, R. Limeback, <http://rudyc.ca/doomsday.html>
- [2] Simply Logical, Intelligent Reasoning by Example, P. Flach, 1994, John Wiley & Sons, Chichester
- [3] SWI-Prolog 5.6.0 Reference Manual, J. Wielemaker, <http://gollem.science.uva.nl/SWI-Prolog/Manual/>