

Meewerken aan het ExpertSysteem

Een opdracht voor mijn oriëntatiestage



door Alex Goldhoorn

30 maart 2001

Meewerken aan het ExpertSysteem

Een opdracht voor mijn oriëntatiestage

Het eindverslag voor de oriëntatiestage van Alex Goldhoorn.

Door	Alex Goldhoorn
Student aan	De Hanzehogeschool, Hogeschool van Groningen, Faculteit Techniek, afdeling informatica
Opleiding	Hogere Informatica
Studiejaar	2
Stagedocent	H.J. Stapelkamp
Stagebedrijf	Rijksuniversiteit Groningen, Faculteit Rechten, sectie Rechtsinformatica
Bedrijfsmentor	dr. mr. C.N.J. de Vey Mestdagh
Begeleider in de praktijk	Peter Veentjer
Uitgebracht op	Vrijdag 30 maart 2001

Inhoudsopgave

Samenvatting	iii
1. Inleiding	4
2. De Rijksuniversiteit Groningen	5
2.1. Vroeger en nu	5
2.2. De organisatiestructuur van de RuG	5
2.3. Faculteit Rechten	6
2.4. Sectie Rechtsinformatica	6
2.5. De automatisering	7
3. Het ExpertSysteem	9
3.1. Het ExpertSysteem	9
3.2. Het nieuwe ExpertSysteem	11
3.3. The LawGame	14
3.4. Het juristenprogramma	14
4. Mijn opdrachten	15
4.1. De databases bij het ExpertSysteem	15
4.2. FactEditor	15
4.3. RelationEditor	16
4.4. Functies en operatoren	19
4.5. RMI-server	20
4.6. Conclusie van de stageopdrachten	20
5. Professioneel Persoonlijke Vaardigheden	21
5.1. Mijn PPV bij mijn stage	21
5.2. Mijn PPV-plan	21
5.3. Mijn PPV-plan voor de toekomst	22
6. Conclusie	23
Literatuurlijst	24
Bijlage 1 – Weekrapport 4	25
Bijlage 2 – Weekrapport 5	26
Bijlage 3 – Weekrapport 6	27
Bijlage 4 – Weekrapport 7	28
Bijlage 5 – Weekrapport 8	29
Bijlage 6 – Weekrapport 9	30
Bijlage 7 – Organogram RuG	31

Samenvatting

De Rijksuniversiteit Groningen (RuG) bestaat sinds 1614, ook de Faculteit Rechtsgeleerdheid bestaat sindsdien. Sinds de negentiger jaren bestaat de sectie Rechtsinformatica onder leiding van dhr. De Vey Mestdag, een onderdeel van de vakgroep Rechtstheorie. En houdt zich bezig met onderwijs en onderzoek.

Onderwijs over Rechtsinformatica, bijvoorbeeld over auteursrechten, Internet, privacy, telecommunicatie en dergelijke.

Qua onderzoek is de sectie Rechtsinformatica sinds 1991 bezig met het maken van een ExpertSysteem. De bedoeling van dit systeem is om de computer conclusies te laten trekken uit de voor hem beschikbare wetten (regels), feiten en doelen. Die doelen worden beredeneerd uit de regels en feiten. Voor dit redeneren wordt de niet-monotone logica gebruikt.

Het ExpertSysteem is destijds (1991 gemaakt) en wordt nu opnieuw gemaakt in de programmeertaal Java, waarbij Peter Veentje de projectleider is qua programmeren.

Mijn opdrachten waren voornamelijk in de vorm van programmeeropdrachten aan het ExpertSysteem, onderdelen toevoegen. In het begin van de stage moest ik eerst “kennis maken” met het systeem, daarna lukte dit wel grotendeels zelfstandig en ook met hulp van Peter.

Mijn Professioneel Persoonlijke Vaardigheden heb ik niet erg goed kunnen verbeteren.

1. Inleiding

Op 23 augustus 1614 gaan de eerste studenten naar de toen net opgerichte Rijksuniversiteit Groningen. Nu, in 2001, gaan er ruim 22.000 studenten naar de Rijksuniversiteit Groningen, RuG afgekort.

Sinds begin negentiger jaren bestaat de sectie Rechtsinformatica aan de RuG, dit is een onderdeel van de vakgroep Rechtstheorie die weer een onderdeel is van de Faculteit der Rechtsgeleerdheid of Faculteit Rechten zoals deze tegenwoordig wordt genoemd. Sindsdien is dhr. De Vey Mestdagh het hoofd van deze afdeling.

Vanaf 1991 zijn ze op de sectie Rechtsinformatica bezig met het maken van het ExpertSysteem, bedoeld om op de wet los te laten waardoor we uiteindelijk een geautomatiseerd rechtssysteem kunnen krijgen wat als voordeel heeft en misschien ook wel als nadeel dat het alleen naar de regels en feiten kijkt.

Het doel van dit verslag is om een beeld te geven van mijn werkzaamheden tijdens de stage. Ook om informatie te geven over de RuG en specifiek over de sectie Rechtsinformatica.

In dit verslag zal eerst de Rijksuniversiteit Groningen worden besproken in het tweede hoofdstuk. Hierin wordt het ontstaan, de toekomst, de automatisering en de organisatiestructuur van de universiteit besproken. En er wordt aandacht besteed aan de Faculteit Rechten en de Sectie Rechtsinformatica.

In hoofdstuk drie worden het idee om het ExpertSysteem te maken, het oude en het nieuwe ExpertSysteem, the LawGame en het juristenprogramma besproken.

Daarna worden de opdrachten besproken die ik tijdens mijn stage heb gemaakt: de FactEditor, RelationEditor, functies en operatoren en de RMI-server.

De Professioneel Persoonlijke Vaardigheden (PPV) worden in hoofdstuk vijf besproken. Hier ga ik in op mijn PPV-doelen, in hoeverre ik deze gehaald heb en mijn PPV-plan voor de toekomst.

In het laatste hoofdstuk staan de conclusies. En in de bijlagen staan de weekrapporten van week vier tot en met week negen en het organogram van de RuG.

2. De Rijksuniversiteit Groningen

In dit hoofdstuk wordt de Rijksuniversiteit Groningen (RuG) besproken. In de volgende paragraaf wordt het ontstaan van de RuG en hoe het nu is besproken. In de tweede paragraaf wordt de organisatiestructuur besproken. De derde paragraaf gaat over de Faculteit Rechten en de vierde gaat over de Sectie Rechtsinformatica en de laatste over de automatisering binnen de RuG en de sectie Rechtsinformatica

2.1. Vroeger en nu

De RuG is de op een na oudste universiteit van Nederland, opgericht op 23 augustus 1614. Het bestaan van de universiteit was niet altijd even zeker, maar in 1876 garandeerde de wet op het Hoger Onderwijs het voortbestaan van de universiteit. De universiteit had toen vier faculteiten en zes hoogleraren. Nu, in 2001, zijn dit tien faculteiten, gehuisvest op ongeveer 150 verschillende locaties in en rond Groningen. Totaal heeft de RuG meer dan 5.000 personeelsleden, waaronder 300 hoogleraren en 22.000 studenten. De studentinstroom per jaar ligt rond de 3.500 studenten en het jaarlijks budget is ongeveer 600 miljoen gulden.

De RuG vindt het gebruik van Informatie en Communicatie Technologie (ICT) bij onderzoek en onderwijs belangrijk. Ook wil de RuG een bijdrage leveren aan het ontwikkelen van toepassingsmogelijkheden hiervoor.

Steeds meer studenten kiezen ervoor om hun academische opleiding gedeeltelijk in het buitenland te volgen. Ook de RuG biedt Engelstalige cursussen aan voor internationale studenten aan. Er worden bij veel onderzoeksgroepen al gebruikt gemaakt van jonge onderzoekers uit het buitenland om een team aan te vullen.

De bedrijfscultuur van de RuG en van een universiteit in het algemeen zal anders zijn dan dat van een bedrijf. Dit omdat een bedrijf gericht is op commerciële doeleinden en een universiteit gericht is op onderzoek en onderwijs. Maar bij de universiteit is ook een hiërarchie aanwezig (zie de bijlage) waarbij vrijwel iedereen verantwoording moet afleggen en goedkeuring moet vragen. En in een universiteit heb je nog een andere soort hiërarchie, wat als oorzaak vooral het onderwijsgevende karakter heeft, het niveau van kennis. Voor elk vak is er een of zijn er meer hoogleraren, studenten volgen deze vakken om hun kennis te vergroten.

De universiteit bevindt zich niet op een bedrijfsmarkt en er is ook niet zo'n commerciële concurrentie bij de universiteiten. Maar er is wel een onderwijsmarkt en de universiteiten proberen wel een goed imago te krijgen en te behouden, door de internationalisering is dit steeds belangrijker geworden.

Het doel van de Rijksuniversiteit Groningen en die van een universiteit over het algemeen zijn onderwijs en onderzoek.

2.2. De organisatiestructuur van de RuG

De organisatiestructuur van de universiteit bestaat uit drie lagen met elk hun eigen taken:

- **Centraal niveau:**

- *het College van bestuur*

Dit is het centrale bestuursorgaan van de universiteit, deze regelt de centrale taken die niet door de universiteitsraad worden geregeld. Deze bestaat uit de Rector Magnificus en twee andere leden.

- *de Universiteitsraad*

Deze bestaat uit 29 leden en is verantwoordelijk voor de begroting van het bestuursreglement en het kiesreglement.

- **Facultair niveau:**

- *de Faculteitsbesturen*
Deze zijn onder andere verantwoordelijk voor de organisatie en de coördinatie van het onderwijs en de wetenschapsbeoefening van de betreffende faculteit.
- *de Faculteitsraden*
Deze zorgen voor het vaststellen van het faculteitsreglement, onderwijs- en examenregelingen, richtlijn voor de wetenschapsbeoefening en het facultair begrotingsplan. En de faculteitsraad heeft een rol bij de benoeming van hoogleraren.
- **Vakgroep niveau**
 - *de vakgroepen*
De vakgroepen verzorgen het onderwijs en verrichten onderzoek. De onderzoeksprogramma's worden uiteindelijk door het bestuur van de vakgroep vastgesteld en deze moet ook door de faculteitsraad worden goedgekeurd. De uitvoering van het onderzoeksprogramma en de examenregelingen moet gedaan worden door de vakgroep. De vakgroep bestaat voornamelijk uit wetenschappelijk personeel.

Zie voor het organogram van de Rijksuniversiteit Groningen de bijlage.

2.3. Faculteit Rechten

De faculteit Rechtsgeleerdheid of Rechten is een van de eerste vier faculteiten van de RuG. Deze faculteit staat bekend als een moderne, internationaal georiënteerde instelling.

De faculteit onderhoudt intensieve contacten met toekomstige studenten, alumni, juridische beroepsgroepen, opdrachtgevers van onderzoek en juridische faculteiten in binnen- en buitenland. De Groningse juridische opleiding staat bekend als degelijk en van goede kwaliteit. Het onderwijs richt zich veelal op:

- de systematiek van het recht;
- het leren formuleren, analyseren en oplossen van juridische problemen;
- het kritisch beoordelen van rechtsontwikkelingen;
- internationale en rechtsvergelijkende vlakken;
- samenwerking en uitwisseling met buitenlandse universiteiten.

Veel onderwijs wordt gegeven in de vorm van werkgroepen en practica die een actieve deelname van de student vergen waardoor de student zich de vaardigheden kan aanleren die een jurist nodig heeft.

De faculteit voelt zich en is verbonden met het noorden. Het opdrachtonderzoek en het postacademisch onderwijs zijn afgestemd op de vragen en behoeften van klanten in dit gebied. Tweederde van de studenten komt uit de noordelijke provincies. Aan het eind van 1998 had de faculteit Rechten 3.117 studenten.

2.4. Sectie Rechtsinformatica

De sectie Rechtsinformatica is een onderdeel van de vakgroep Rechtstheorie. De vakgroep rechtstheorie bevat de volgende secties:

- Rechtsinformatica
- Rechtssociologie
- Rechtsfilosofie
- Politieke wetenschappen

De sectie Rechtsinformatica houdt zich bezig met het verzorgen van onderwijs en verrichten van onderzoek op de gebieden rechtsinformatica en informaticarecht onder leiding van dhr. De Vey Mestdag. Die tevens budgethouder van de sectie is.

Het informaticarecht behandelt de rechtsvragen die worden opgeroepen door het toenemende gebruik van IT in onze samenleving, bijvoorbeeld computercriminaliteit, e-commerce en digitale burgerrechten.

De rechtsinformatica behandelt het ontwikkelen en gebruiken van IT voor het uitvoeren van juridische werkzaamheden, bijvoorbeeld digitale ontsluiting van rechtsbronnen en juridische expertsystemen.

Sinds 1992 houdt de sectie zich bezig met de ontwikkeling en implementatie van een kennissysteem voor de uitvoering van wet- en regelgeving. Dit project heeft inmiddels geleid tot het ESM (ExpertSysteem Milieuvergunningrecht) dat nu wordt herschreven. De programmeurs overleggen met dhr. De Vey Mestdag hoe het ExpertSysteem ongeveer moet werken en vooral hoe de gebruiker ermee kan werken. Dus vooral de werking voor de gebruiker, hoe die ermee moet werken en ook de logica erachter.

De faculteit heeft op dit moment het volgende aanbod van onderwijs op het gebied van recht en informatietechnologie:

- *Recht en Informatietechnologie (Rechtsinformatica 1)*
Over de functie en de betekenis die IT voor juristen kan hebben.
- *Rechtsinformatica 2*
Er wordt ingegaan op de werking, betrouwbaarheid en de juridische toepassingen van het internet en van juridische kennissystemen. Studenten mogen met deze systemen werken.
- *Informaticarecht*
Er komen allerlei onderwerpen aan de orde, zoals: informatie en recht, internet en recht, bescherming van auteursrechten, privaatrechtelijke en strafrechtelijke aspecten van het computergebruiken, de privacyaspecten en telecommunicatierecht.
- *Informatierecht voor Bedrijfskundigen*
- *Scriptie schrijven bij de sectie Rechtsinformatica*

2.5. De automatisering

Bij de RuG worden apparatuur, programmatuur, software en internet (voornamelijk) aangeboden door het RekenCentrum (RC). Het RekenCentrum levert de volgende automatiseringsdiensten:

- leveren van computerapparatuur voor de RuG;
- leveren van computerapparatuur voor medewerkers, via een PC-privé project;
- RuGNet, dit is de backbone van de RuG en deze is verbonden met het SURF-Net, hierdoor kunnen medewerkers en studenten op de universiteit gebruik maken van internet;
- Oprit, via deze service kunnen studenten van de RuG en van de Hanzehogeschool thuis gebruik maken van het Internet;
- Electronic Publishing Center (EPC), via deze service kunnen dingen op het Internet worden gepubliceerd;
- Expertisecentrum Computerondersteunend Onderwijs (ECCOO), hierbij kan een elektronische leeromgeving worden gemaakt;
- High-Performance Computing (HPC), deze levert zware rekenkracht.

De standaardsoftware die gebruikt wordt zoals: Microsoft Windows NT en 2000, Microsoft Office en allerlei andere hulpmiddelen zijn centraal beschikbaar voor de universiteit. En een universiteit hoeft vaak niets of minder te betalen voor het registreren van een

softwarepakket of –programma. En dit wordt samen met het netwerk beheerd door het RekenCentrum.

De faculteiten zelf hebben ook hun eigen servers met hun eigen software, zo heeft de Faculteit Rechten software waar wetten in kunnen worden gezocht en bekeken en programma's om rechtenidoom in op te zoeken en andere dergelijke programma's.

Op de sectie Rechtsinformatica staat ook een server waarop de website van de sectie staat, software die bestemd is voor studenten (van Rechtsinformatica) en software voor de medewerkers van de sectie. Er staan dus ook programma's op voor de programmeurs. Deze server en de accounts om gebruik te kunnen maken van het netwerk van de sectie wordt voornamelijk beheerd door Rob Sabelis.

De software die door de programmeurs (Peter Veentjer de hoofdprogrammeur en andere informaticastagiaires) worden gebruikt zijn:

- Kawa, dit is een multifunctionele editor waarin onder ander in de programmeertaal Java kan worden gewerkt;
- JDK 1.3 (Java Development Kit), voor het compileren;
- Jikes, een snellere Java-compiler;
- Sybase, een databasepakket die ik voornamelijk heb gebruikt om mijn programmaonderdelen te testen;
- Interbase, ook een databasepakket;
- Antler, de parser om de taal van het ExpertSysteem te definiëren.

Voor de documentatie wordt voornamelijk gebruik gemaakt van Java-sites op het Internet zoals die van SUN (<http://java.sun.com>) en Java lobby (<http://www.javalobby.com>) en er is gebruik gemaakt van boeken.

Voor er nieuwe software of boeken worden aangeschaft waarvoor betaald moet worden, moet dit eerst overlegd worden met dhr. De Vey Mestdagh.

3. Het ExpertSysteem

Het idee om een juridisch kennissysteem te ontwikkelen kwam van dhr. De Vey Mestdagh in 1988, er is toen praktijk- en literatuuronderzoek gedaan. In 1991 is men begonnen met de uitvoering en in 1996 is het onderzoek afgerond.

In dit hoofdstuk wordt het ExpertSysteem behandeld. In paragraaf zal het doel en het oude ExpertSysteem worden behandeld. De tweede paragraaf gaat over het ExpertSysteem waaraan nu wordt gewerkt. Daarna komt de paragraaf over the LawGame en de laatste paragraaf gaat over het ExpertSysteem als juristenprogramma.

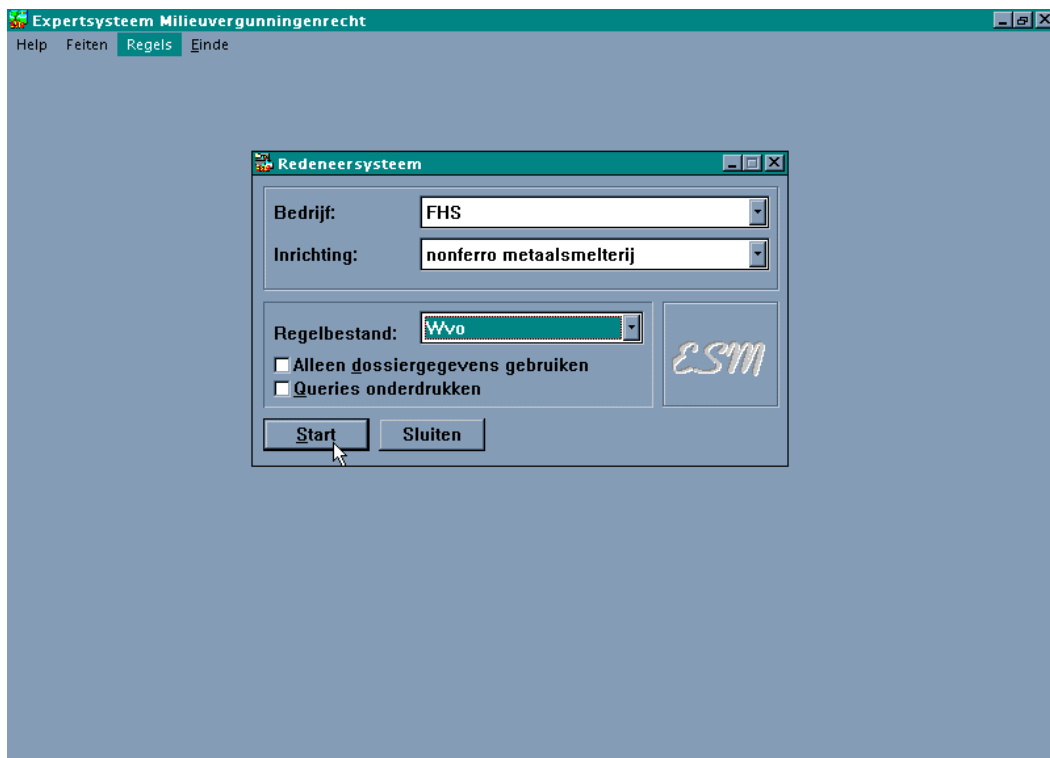
3.1. Het ExpertSysteem

Het ExpertSysteem is bedoeld om de computer te laten helpen met het nemen van beslissingen. Dit gebeurt door het ExpertSysteem een rulebase (lijst met regels), een database met feiten en de doelen te geven en het systeem daaruit de gemotiveerde conclusies te laten trekken. Om dit systeem te testen heeft men er voor gekozen om dit eerst te testen met het verlenen van milieuvergunningen. Om te testen of het ExpertSysteem goede conclusies trok en misschien zelfs aanvullend werden 40 milieucasussen door het ExpertSysteem verwerkt en daarna gecontroleerd door achttien onafhankelijke beoordelaars. Hieruit bleek dat alle conclusies (met uitzondering van enkele) goed waren en soms zelfs aanvullend dan die van mensen.

De eerste versie van het ExpertSysteem is gemaakt in de programmeertalen C++ en Delphi, en deze versie maakt gebruik van ODBC van Windows (voor databasekoppelingen).

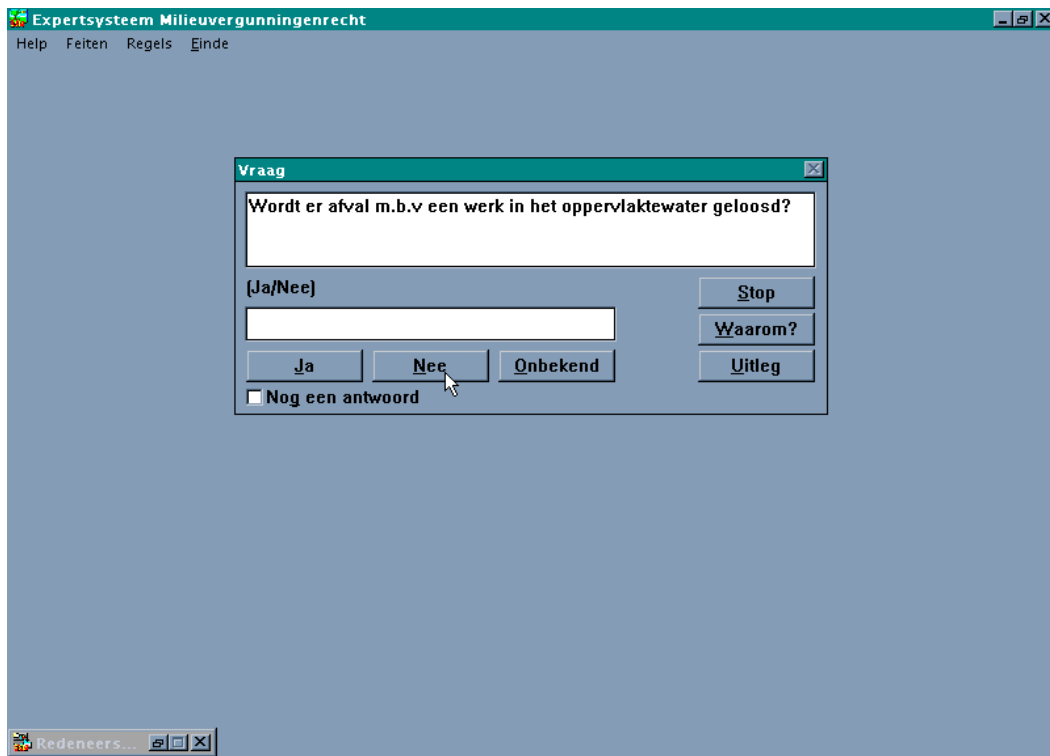
Op de webpagina van de sectie Rechtsinformatica staat een voorbeeld voor het weergeven van de werking van het ExpertSysteem.

Eerst moet voor het nemen van conclusies het bedrijf (de casus) worden geselecteerd, het bedrijf, de inrichting en het regelbestand, zie figuur 3.1.



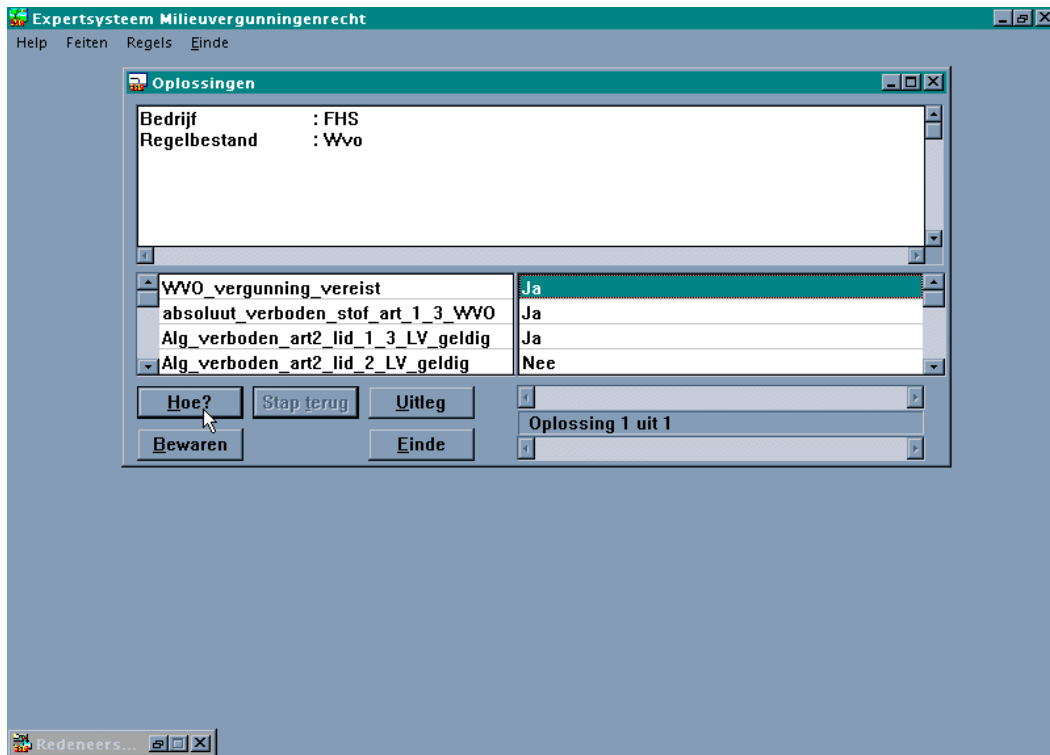
Figuur 3.1: Het bedrijf, de inrichting en het regelbestand moeten worden opgegeven bij het redeneersysteem.

Daarna moeten vragen over het bedrijf worden beantwoord om tot een conclusie te komen, zie figuur 3.2.



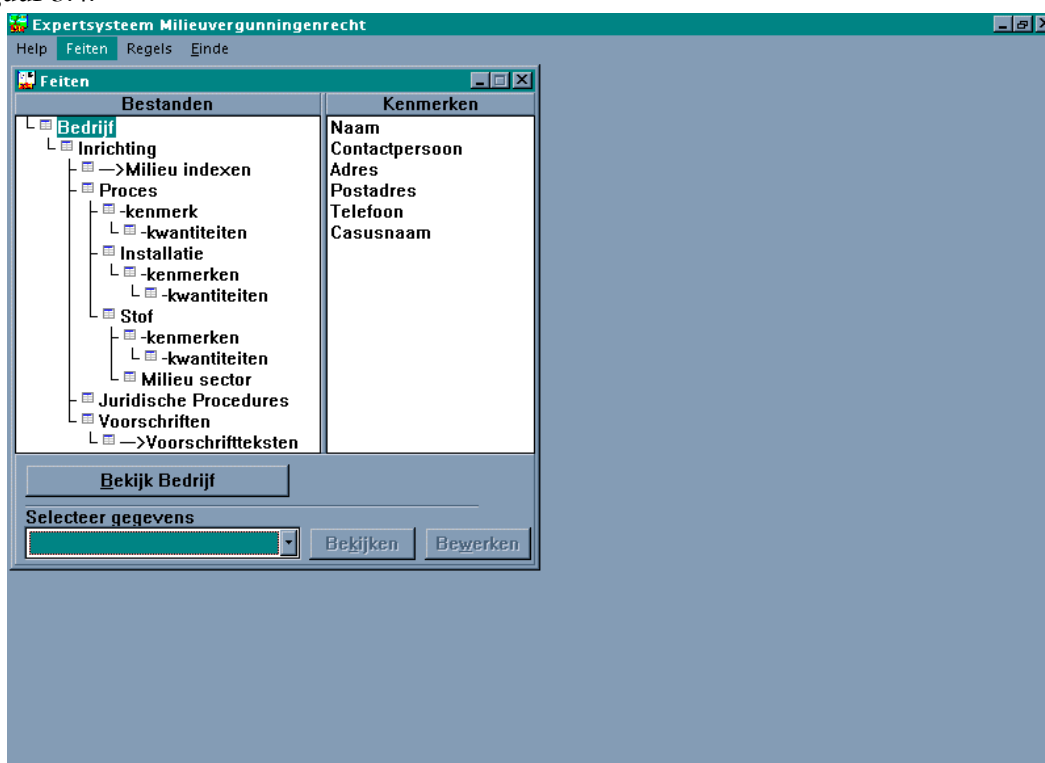
Figuur 3.2: Er moeten vragen worden gesteld om tot een conclusie te komen.

En dan worden de conclusies weergegeven, die zijn ontstaan door het gebruik van de door de gebruiker ingevoerde antwoorden (feiten) en de regels uit het regelbestand (de wetten), zie figuur 3.3.



Figuur 3.3: De conclusies worden als laatst weergegeven.

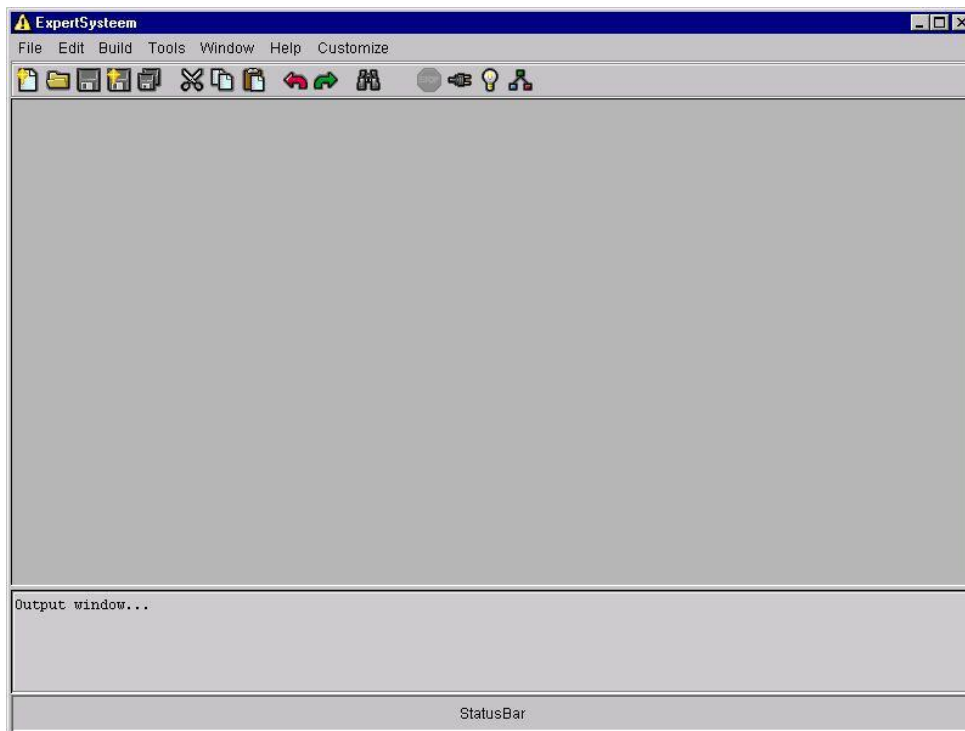
In het feitenscherm kunnen de tabellen en de kolommen van die tabellen worden bekeken, zie figuur 3.4.



Figuur 3.4: Het feitenscherm.

3.2. Het nieuwe ExpertSysteem

Nu wordt sinds de zomervakantie gewerkt aan een nieuw ExpertSysteem in Java, zie figuur 3.5. Dit ExpertSysteem zal na de zomervakantie van 2001 eerst worden gebruikt door studenten als the LawGame, hierover meer in de volgende paragraaf. Daarna zal het programma geschikt worden gemaakt voor juristen.



Figuur 3.5: Het opstartscherm van het ExpertSysteem.

Peter Veentjer heeft tijdens zijn laatste stage geholpen met het nieuwe ExpertSysteem, eerst was het de bedoeling dit verder in Delphi en C++ te doen, maar hij besloot toch over te gaan naar Java. Deze taal heeft hij vooral gekozen, vanwege het objectgeoriënteerde programmeren.

Het nieuwe ExpertSysteem is tot nu toe alleen nog maar een ontwikkelomgeving. Voor het definiëren van feiten, regels en doelen is een taal gedefinieerd die vrij veel op de programmeertaal Pascal lijkt.

Functies en operatoren moesten opnieuw worden gemaakt, ondanks dat deze veelal in Java zitten. Dit moet, omdat er gebruik wordt gemaakt van de niet-monotone logica. In deze logica wordt inconsistentie toegestaan door het creëren van meer werelden. Bijvoorbeeld als je zegt ‘alle koeien zijn groen’, en je zegt ‘alle koeien zijn rood’. Dan maak je twee werelden, een met waar alle koeien groen zijn en een waar alle koeien rood zijn. Er zijn ook situaties met bijvoorbeeld: “alle vogels kunnen vliegen”, maar dat klopt niet bij een pinguïn. Je kunt dan zeggen dat de regel “pinguïns kunnen niet vliegen” belangrijker is. Ook is het toegestaan om te zeggen dat iets onbekend is, daarom moeten de operatoren en functies aangepast worden, zodat ze ook met een of meer onbekenden kunnen werken. Een paar voorbeelden:

```
2 + 3 = 5
2 + onbekend = onbekend.
2 * onbekend = onbekend
0 * onbekend = 0
```

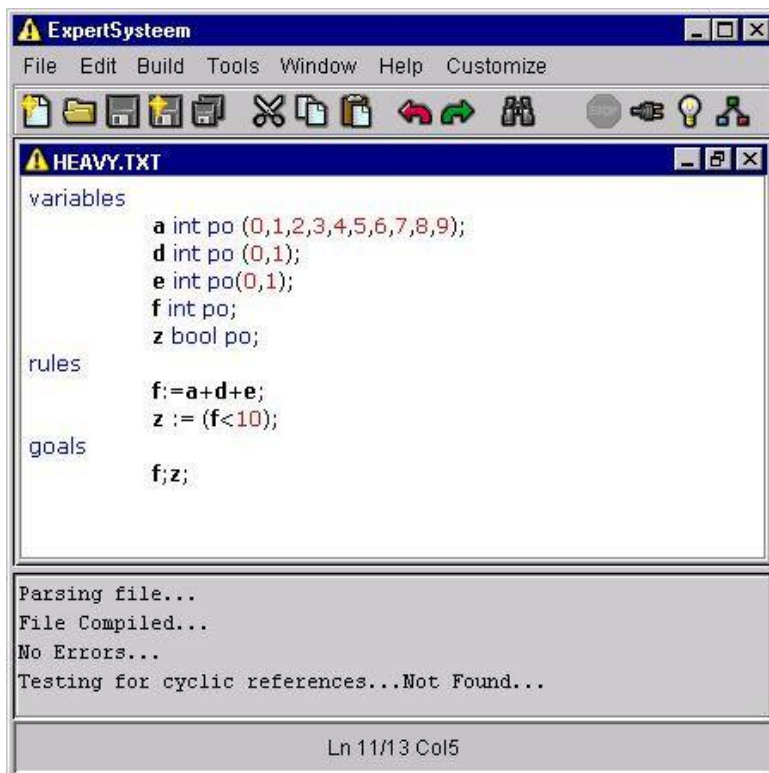
dit gaat ook op met logische formules:

```
waar EN onwaar = onwaar
waar EN onbekend = onbekend
onwaar EN onbekend = onwaar
waar OF onbekend = waar
```

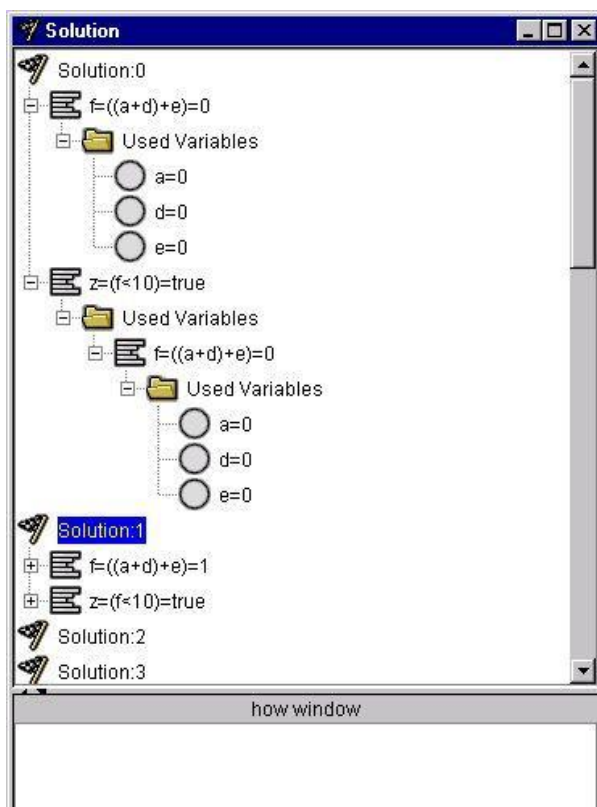
De gebruiker kan dan in een tekstverwerker in het systeem de variabelen, regels en doelen definiëren, zie figuur 3.6. Daarna moet het worden geparst, dan wordt de syntax gecontroleerd, wordt er naar cyclische referenties gezocht en dan wordt de of worden de oplossing(en) in een boom weergegeven in het Solution venster, zie figuur 3.7.

Voor het maken van een parser zijn verschillende programma's. Peter gebruikte het programma Antler. Dit zet een bestand dat volgens een bepaalde opmaak en syntax is gemaakt om naar twee Java-bestanden.

En in de database kunnen dan regels en feiten komen te staan. En er is dan de mogelijkheid om in het systeem de database te bewerken. In de MultiDatabaseEditor, FactEditor en RelationEditor.



Figuur 3.6: De tekstverwerker binnen het ExpertSysteem waarin variabelen (variables), regels (rules) en doelen (goals) kunnen worden bewerkt.



Figuur 3.7: Het Solution venster, de oplossingen worden getoond.

3.3. The LawGame

De LawGame zal gemaakt worden aan de hand van het ExpertSysteem. De LawGame is bedoeld voor rechtenstudenten (die enige interesse in rechtsinformatica hebben). Deze studenten kunnen dan met het programma een wereld creëren met bepaalde wetten. Door het gebruik van e-mail en chatfunctie in het programma kunnen ze met elkaar overleggen of men een bepaalde wet gaat aannemen of verwijderen. Het systeem kan dan redeneren wat er gebeurd als men een wet aanneemt of verwijderd.

Het ExpertSysteem waar nu aan gewerkt wordt zal vooral bedoeld zijn voor de beheerder van het hele systeem. Voor de studenten zal waarschijnlijk een speciale client-versie worden gemaakt. Er wordt dus ook gebruikt gemaakt van client/server-technologie, met de RMI-server (zie paragraaf 4.5).

Om studenten toegang te geven en ze overal te kunnen laten spelen wordt alles op de server opgeslagen. In een speciale database die meta-informatie bevat staan dingen als info over de casussen en de databases en over de gebruikers en hun rechten (lezen, schrijven, wijzigen en wissen).

Voor een spel moet eerst een bepaalde casus worden gekozen, bijvoorbeeld een metaalsmelterij of een varkenshouderij. Bij elke casus hoort een eigen rulebase en die kan dan worden aangevuld door de studenten

3.4. Het juristenprogramma

Na het maken van het programma voor de studenten, zal er een programma worden gemaakt voor juristen. Juristen kunnen dan met dit programma de benodigde feiten invoeren, daarna kan er door het systeem naar geldige conclusies worden geredeneerd waarbij gebruik wordt gemaakt van wetten die gedefinieerd zijn, dit kunnen dus alle overheidswetten zijn.

4. Mijn opdrachten

Tijdens mijn stage moest ik als stageopdracht verschillende dingen programmeren voor het ExpertSysteem, in de programmeertaal Java. Deze opdrachten kreeg ik voornamelijk van Peter Veentjer. Hij was en is degene die het project op programmeergebied leidt. Dhr. De Vey Mestdagh beslist uiteindelijk hoe het programma er ongeveer uit komt te zien, wat er in moet zitten, hoe dit er ongeveer uit moet zien en de logische werking (het redeneren). Voor mijn opdrachten heb ik dus eigenlijk alleen maar met Peter te maken gehad en soms moest dhr. De Vey Mestdagh (mee)beslissen over de werking en/of weergave van een onderdeel uit het programma.

In de eerste twee weken had ik vrij veel hulp van Peter nodig, omdat ik nog niet goed wist hoe het ExpertSysteem in elkaar zat. Ook moest hij mij uitleggen wat ik moest doen, en maakte een beginnetje voor mij. Na de eerste weken ging het steeds beter en had ik veel minder hulp van Peter nodig.

In dit hoofdstuk wordt in de eerste paragraaf het gebruik van databases bij het ExpertSysteem besproken. In de paragrafen daarna worden de opdrachten besproken die ik gemaakt heb: de FactEditor, de RelationEditor, functies en operatoren en de conclusie over mijn opdrachten

4.1. De databases bij het ExpertSysteem

De feiten die in het ExpertSysteem gebruikt worden, zullen voornamelijk in een database komen te staan. En het programma zal daarom functionaliteit bieden om de database te bewerken en te bekijken. In het programma was er al een mogelijkheid aanwezig, namelijk de MultiDatabaseEditor. Hierin staat de multidatabase in een boom. Boven in de boom zit de multidatabase, die de database als takken heeft, de database heeft de tabellen en de tabellen hebben de relatie en kolommen in hun takken. In de takken staan ook opties om eigenschappen aan te passen zoals de naam en help-URL.

Om het de gebruiker makkelijk te maken, wordt er nog een optie aan de gebruiker geboden om de multidatabase te tonen en te wijzigen. Dit gebeurt dan door de FactEditor en de RelationEditor.

Voor het testen heb ik bij beide, de FactEditor en de RelationEditor, het databasesysteem Sybase SQL Anywhere 7.0 (shareware versie) gebruikt. En bij Sybase zit een voorbeelddatabase die ik gebruikt heb om de FactEditor en de RelationEditor uit te testen.

Voor the LawGame zullen meerder multidatabases worden gebruikt. Per casus (een specifieke 'wereld', bijvoorbeeld een varkenshouderij of een staalsmelterij). En per casus zijn er dan standaard bepaalde bedrijven uit die sector die in een database staan en ook bepaalde regels. In een andere database zullen die multidatabases gaan staan als zogenaamde blobs, dit zijn Java-objecten die door het databaseprogramma kunnen worden ingelezen. En er moet ook een tabel van gebruikers in staan met hun naam, wachtwoord en restricties.

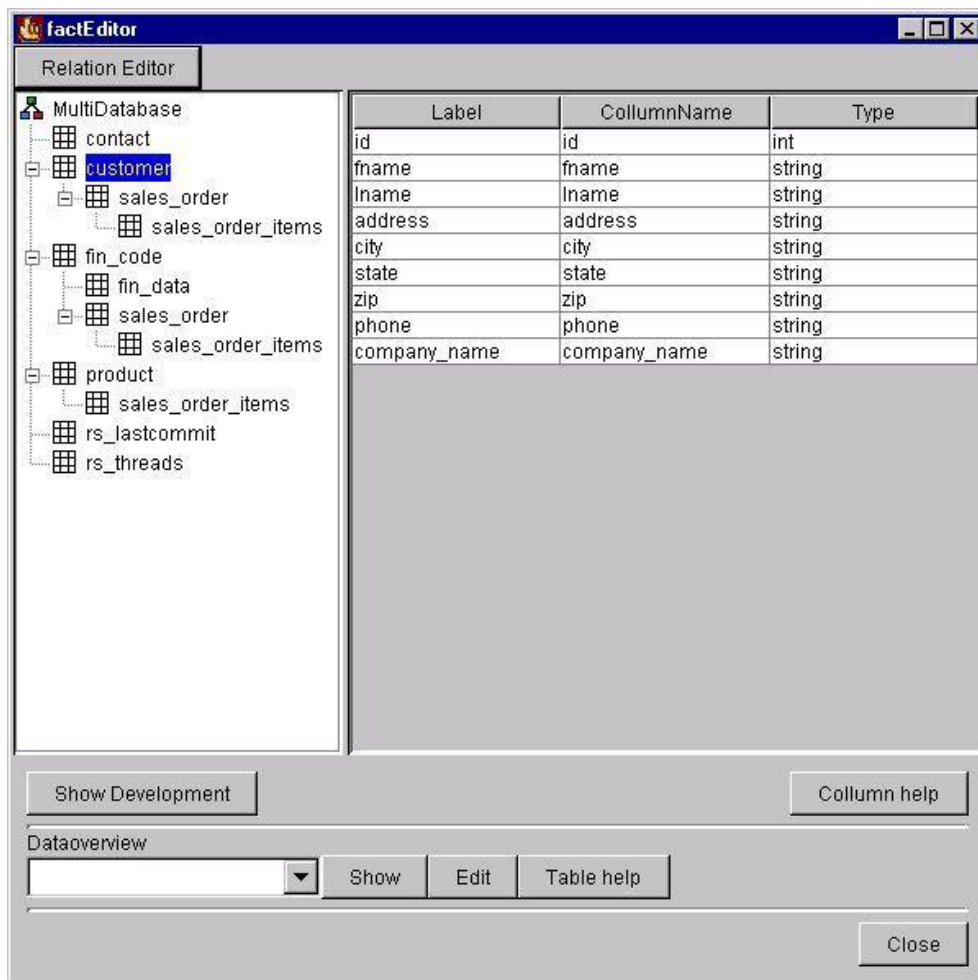
4.2. FactEditor

De FactEditor, of FeitenBewerker, is zoals de naam al zegt bedoeld om feiten te bewerken, zie figuur 4.1. Deze feiten staan in een database. Deze feiten staan in een database en het ging er daarom vooral om dat het kom omgaan met multidatabases.

De FactEditor is een scherm die uit drie delen bestaat. Links staat de boom met alle tabellen van de multidatabase. Deze tabellen staan zo in de boom als hun relatie is. In de root staan alle tabellen die niet voorkomen als secundaire tabel in een relatie. En onder elke tabel staat de secundaire tabel.

Rechts worden de kolomnamen en –typen neergezet van de geselecteerde tabel in de boom. En onderaan zitten de knoppen.

Deze opdracht was voor mij niet heel makkelijk, vooral omdat dit de opdracht was waarmee ik begon en toen had ik nog geen kennis van het ExpertSysteem, de benodigde tools en de werkwijzen van Peter. Wat ik een heel irritante klus vond was om de knoppen onderaan het scherm goed op het formulier te plaatsen en er voor te zorgen dat ze ongeveer zo blijven staan bij het vergroten of verkleinen van het venster.



Figuur 4.1: De FactEditor. Links staat de boom met tabellen, rechts de kolommen van de tabel customer

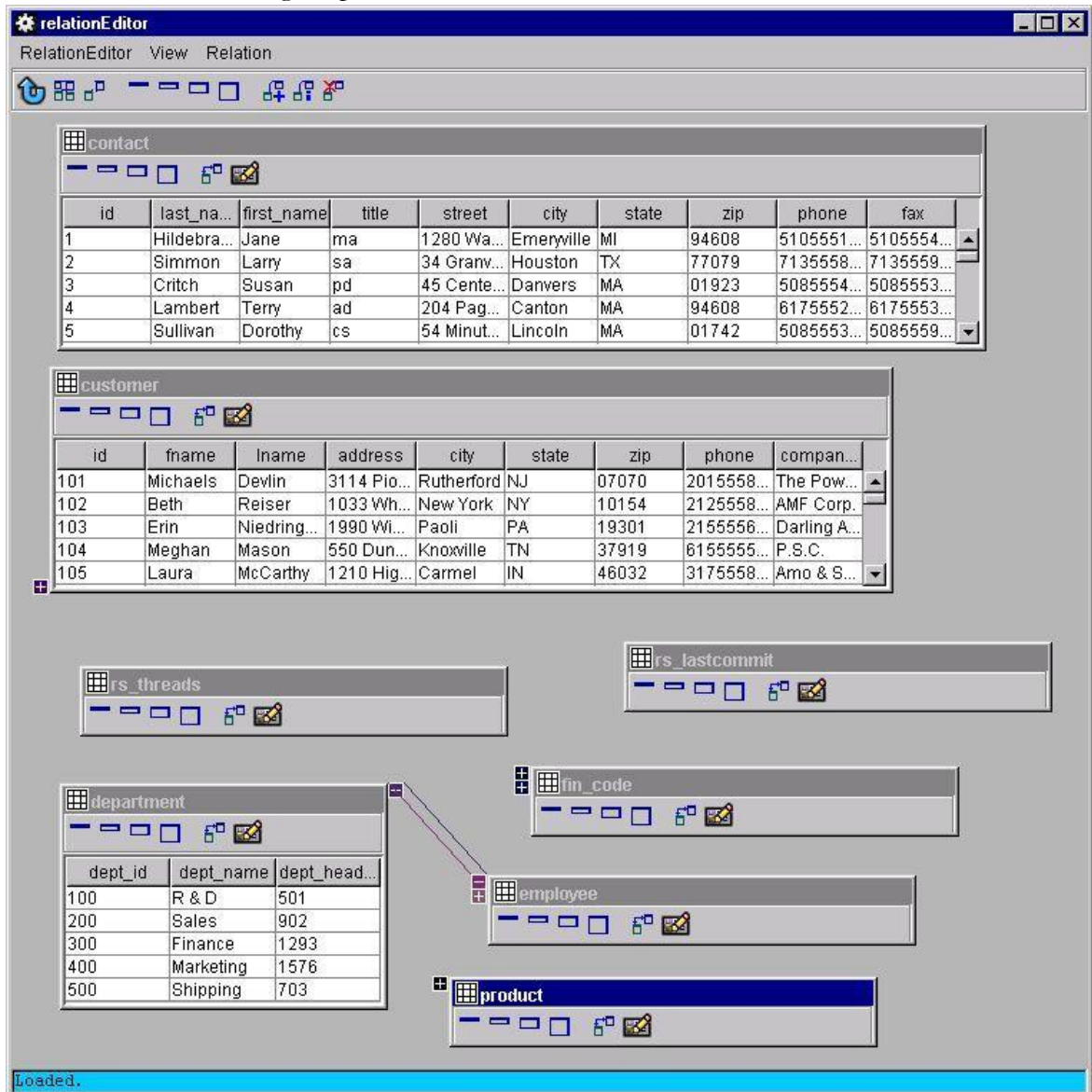
4.3. RelationEditor

De RelationEditor, of RelatieBewerker, is zoals de naam al zegt bedoeld om relaties te bewerken. Deze relaties staan in de database. En deze relaties kunnen gewist worden, bekeken worden en er kan informatie over getoond worden, zie figuur 4.2.

Bij het opstarten van de RelationEditor worden alleen de zogenaamde roottabellen (zie figuur 4.2) (de tabellen die niet als secundaire tabel voorkomen in een relatie) weergegeven. Ze worden allemaal naast elkaar en onder elkaar gezet. Bij het opstarten of herladen van de RelationEditor worden alleen circulaire relaties getoond, de overige relaties worden niet getoond, maar alleen een [+].

De relaties worden weergegeven door lijntjes die van de primaire naar de secundaire tabel gaan. Deze lijntjes zijn zo kort mogelijk. Aan de kant van de primaire tabel zit een vierkantje (later aangeduid als []), hierin zit een + of een – ([+] of [-]), dit hangt er van af of de secundaire

tabel wordt weergegeven of niet. Van alle niet zichtbare tabellen worden de relatielijnen ook niet weergegeven en ook niet als de primaire tabel alleen zichtbaar is. Als de primaire tabel zichtbaar is dan staat er sowieso een [+] of [-] om aan te geven dat er een relatie is, waarbij die tabel de primaire is. Als de primaire tabel alleen zichtbaar is en de secundaire niet, dan staat er een [+], wat aangeeft (zoals in een boom) dat je de relatie kunt 'uitklappen', als je hierop klikt dan wordt de secundaire tabel zichtbaar. En er staat een [-] als beide tabellen in de relatie zichtbaar zijn, als je hierop klikt wordt de relatie ingeklapt en wordt de secundaire tabel onzichtbaar.



Figuur 4.2: De RelationEditor. Alleen de roottabellen zijn zichtbaar (en circulaire relaties)

Ook kan er op een relatie worden geklikt. Bij een klik op de relatielijns wordt of worden deze geselecteerd. Met een rechtermuisklik op een relatielijns worden opties gegeven als het verwijderen en informatie weergeven van de relatie, deze opties zitten ook in het menu en op de toolbar van de RelationEditor. De opties in het menu zoals het verwijderen en informatie geven over een relatie gebruiken de geselecteerde relatielijns. Voordat een relatie wordt gewist, wordt er nog wel om een bevestiging gevraagd.

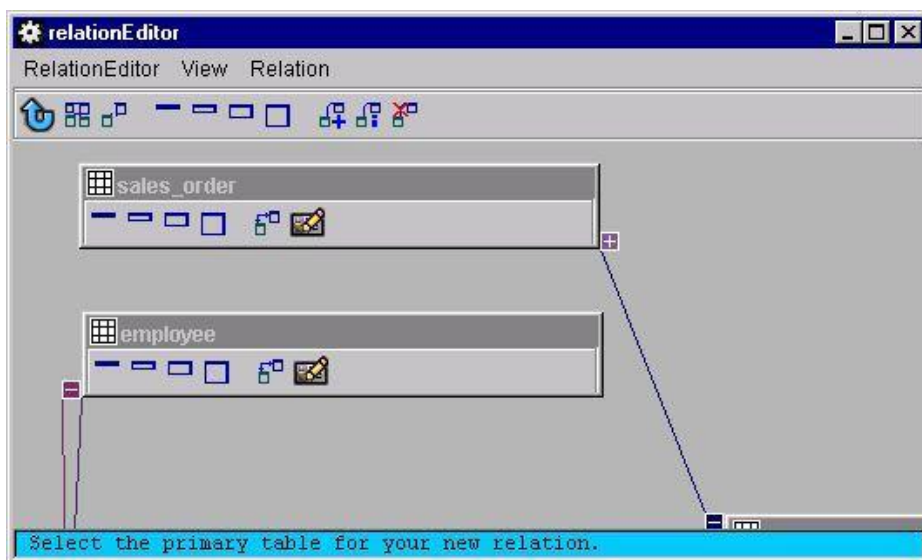
Op de toolbar en in het menu zit ook de optie om een relatie toe te voegen. Bij het toevoegen van een relatie wordt eerst gevraagd om de primaire tabel te selecteren (zie figuur 4.3) en daarna de secundaire tabel. Dit selecteren is alleen maar de focus zetten op een tabelvenster. Als de tabellen zijn geselecteerd, wordt de RelationWizzard gestart (zie figuur 4.4) en deze gaat

verder met het scherm waar de gebruiker de kolom van de secundaire tabel kan selecteren die een relatie heeft met de sleutel van de primaire tabel.

Voor de rest heeft de RelationEditor nog enkele opties om het overzichtelijk te houden. Hiervoor zijn er verschillende views, die betrekking hebben op de interne vensters met tabellen, namelijk:

- titlebarview , alleen de titelbalk en toolbar worden weergegeven
- minimalview , van de tabel wordt alleen een regel tegelijk laten zien
- normalview , er worden vijf regels laten zien
- maximalview , alle regels worden tegelijk laten zien

Elk intern venster met een tabel heeft knoppen om de view te wijzigen en in de toolbar van de RelationEditor zelf zitten ook knoppen hiervoor, die ervoor zorgen dat elke tabel in die view wordt gezet.



Figuur 4.3: Een relatie toevoegen, de primaire tabel moet worden geselecteerd.



Figuur 4.4: De RelationWizzard. De kolommen voor de relatie moeten worden geselecteerd.

Ook is er een optie om alle tabellen weer te geven, dus ook alle relaties. En er is een optie om alleen de roottabellen weer te geven en de anderen te verbergen.

In het venster met de tabel is ook een optie om de records weer te geven, na het drukken op die knop wordt er een nieuw venster getoond met de inhoud van het geselecteerde record of als er geen is geselecteerd het nieuwe record. In dat venster, de TableEditor, staan de naam van de kolommen aan de linker kant en aan de rechter kant staat de waarde van de kolom. Door middel van knoppen kunnen de records bekeken worden. Met de knop  ga je naar de eerste record, met  ga je naar de vorige knop, met  ga je naar de volgende en met de knop  ga

je naar het laatste record. En je kunt het recordnummer invoeren en daarna op de knop drukken om er naar toe te gaan.

Dit was mijn tweede en grootste opdracht, deze was voor mij een hele uitdaging. In het project van het vierde kwartaal van het eerste jaar had ik al eens samen met mijn projectleden een RDBMS (Relationeel Database Management Systeem) gemaakt en daarbij zat ook een scherm waarin je relaties kon toevoegen, bewerken en verwijderen. Maar aan dat onderdeel heb ik toen zelf niet veel gedaan. Nu moest ik het echter alleen maken. Dit vond ik wel een leuke opdracht, vooral toen ik na zes week het resultaat zag, waar ik best wel trots op was.

4.4. Functies en operatoren

In het ExpertSysteem kunnen regels, feiten en doelen worden gedefinieerd. Dit gebeurt in een interne tekstverwerker.

Alle operatoren en functies moeten opnieuw gemaakt worden, dit moet omdat er gebruik is gemaakt van de niet-monotone logica (in deze logica is het mogelijk om variabelen onbekend te laten).

Voor het ExpertSysteem heb ik de logische operatoren dagger of dolk in het Nederlands en de sheffer stroke geïmplementeerd. De waarheidstabel van de dagger staat in tabel 4.1 en de waarheidstabel van de sheffer stroke staat in tabel 4.2.

Tabel 4.1: De waarheidstabel van de logische operator dagger of dolk.

Linker_operand dagger Rechter_operand = Antwoord

Linker operand	Rechter operand	Antwoord
onbekend	onbekend	onbekend
onbekend	onwaar	onbekend
onbekend	waar	onwaar
waar	onbekend	onwaar
waar	waar	onwaar
waar	onwaar	onwaar
onwaar	onbekend	onbekend
onwaar	onwaar	waar
onwaar	waar	onwaar

Tabel 4.2: De waarheidstabel van de logische operator sheffer stroke.

Linker_operand sheffer_stroke Rechter_operand = Antwoord

Linker operand	Rechter operand	Antwoord
onbekend	onbekend	onbekend
onbekend	waar	onbekend
onbekend	onwaar	waar
onwaar	onbekend	waar
onwaar	waar	waar
onwaar	onwaar	waar
waar	onbekend	onbekend
waar	onwaar	waar
waar	waar	onwaar

En ook heb ik de logaritme functie gemaakt. Deze logaritme krijgt twee parameters mee, het grondgetal en het argument. Deze functie geen onbekende meekrijgt.

Dit was een kleine opdracht om even iets anders te doen tussendoor en duurde ook niet zo lang. Wat ik hier wel van geleerd heb of ten minste kennis mee gemaakt heb is de parser, waarmee een hele taal kan worden gedefinieerd. En ik moest daar mijn operatoren en functie aan toevoegen.

4.5. RMI-server

Voor the LawGame moeten studenten overal in de universiteit (en thuis) gebruik kunnen maken van het programma en vooral de gegevens over hun spel kunnen raadplegen en er moeten meer studenten gebruik kunnen maken van hetzelfde spel. Dit is mogelijk door gebruik te maken van een client/server model. De server zorgt dan voor de toegang en beheer van de database en de spellen en de student maakt dan gebruik van de client.

In Java kan op verschillende manier gebruik worden gemaakt van de client/server technologie. Dit kan door Corba, dit is een protocol waar Java maar ook andere systemen gebruik van kunnen maken, dus dan kan een in C++ geschreven programma communiceren met een in Java geschreven programma.

En het is mogelijk met behulp van RMI (Remote Method Invocation), dit is een protocol alleen gemaakt voor Java. Omdat de applicaties uitsluitend in Java zijn en worden geschreven, wordt er gebruik gemaakt van RMI.

Ik heb zo'n RMI-server geschreven. Eerst heb ik hiervoor een tutorial doorgelezen en wat mee geoefend. Ik heb toen een systeem gemaakt waarbij ik verschillende Java Virtual Machines op mijn PC draaide waarvan één de server en de andere(n) de client(s). Hiervoor moest er een RMI-register actief zijn, hierbij moet de server zich eerst aanmelden onder een bepaalde naam (zoals bijv rmi://ExpertSysteemServer) en met een verwijzing naar een bepaald Java-object op de server. Dan pas kan de client zich aanmelden door eerst aan het RMI-register de verwijzing naar het Java-object te 'vragen' die bij de naam hoort. Als de client de verwijzing naar dat object heeft kan de client dat gebruiken. In mijn systeem moest er eerst ingelogd worden, dus moest de client eerst inloggen. En in mijn systeem was dat object het serverobject zelf. De client roept dan een methode aan in het java-object. Het serverobject controleerde de gebruikersnaam en het bijbehorende wachtwoord uit een database die op de server staat (die wordt aangesproken met behulp van JDBC). Als de server de gebruikersnaam en het wachtwoord heeft goedgekeurd, wordt de client aan een lijst van clients in de server toegevoegd, zodat de server weet welke clients ingelogd zijn. Als de client ingelogd is dan kan hij gegevens van het serverobject vragen uit een database. En dat moeten dan dus de gegevens zijn die voor het spel nodig zijn. Voordat een client afsluit of onder een andere naam opnieuw inlogt moet dit aan de server worden doorgegeven. Maar stel nou dat de client crasht of de server, hoe weet de ander nou of dat het geval is? Daarvoor heb ik bij zowel de client als de server een klasse gemaakt die test of de ander nog leeft, dit gebeurt door een methode in de ander aan te roepen, als er dan een exceptie wordt opgeworpen dan is er iets niet goed en moet hij uitgelogd worden (als de client niet reageert bij de server) of opnieuw inloggen (als de server niet reageert bij de client).

Deze opdracht vond ik een van de leukste. Ik had nog nooit eerder iets op school gehad of thuis geprobeerd met client/server en deze opdracht was daarom ook zeer leerzaam.

4.6. Conclusie van de stageopdrachten

Ik vond de meeste opdrachten leuk om te doen. En ik heb zeker wat van de opdrachten geleerd, namelijk hoe je te werk moet gaan als je moet programmeren. Dit hebben we natuurlijk ook op school geleerd en bij het project van het vierde kwartaal van het eerste jaar, maar alleen in de praktijk is toch wel iets anders. Ik werd wel geholpen, maar je bent toch aardig op jezelf aangewezen.

Eveneens heb ik van Peter allerlei tips gekregen qua programmeren, voor opmaak van programmeercode, sites over Java en dergelijke.

De RMI-server vond ik het leukst om te doen, omdat ik nog nooit iets had gedaan met client/server of communicatie tussen twee computers.

5. Professioneel Persoonlijke Vaardigheden

In dit hoofdstuk vertel ik over mijn Professioneel Persoonlijke Vaardigheden (PPV). De eerste paragraaf gaat over de PPV op mijn stage. De volgende paragraaf gaat over het PPV-plan dat ik voor de stage heb opgesteld en voor hoeverre ik dit gehaald heb. En de laatste paragraaf gaat over mijn PPV-plan voor de toekomst.

5.1. Mijn PPV bij mijn stage

Tijdens mijn stage was er niet veel plaats om mij te ontwikkelen op het gebied van PPV, omdat ik het grootste deel van de tijd alleen bezig was met het programmeren.

Vooraf in de eerste weken hield ik mij voornamelijk bezig met het maken van de programmeeropdrachten en richtte ik mij daar volledig op. De programmeeropdrachten kreeg ik direct van Peter Veentjer. Hij legde mij uit wat de bedoeling was wat ik zou moeten maken en hoe ongeveer. Hij vertelde wat ik kon gebruiken (qua Java-klassen) en gaf mij wat tips. Als ik tijdens het programmeren ‘vast’ kwam te zitten doordat ik niet een oplossing of oorzaak kon vinden van een bepaalde fout, kon ik het Peter altijd direct vragen en hij hielp me dan weer verder. Soms zijn dit hele ‘domme’ fouten, zoals het vergeten van bijvoorbeeld een puntkomma of juist het teveel plaatsen daarvan. Ook kwam Peter regelmatig even bij mij kijken om te vragen hoe het ging.

Met mijn stagebegeleider, dhr. De Vey Mestdagh, had ik niet elke dag contact, maar wel wekelijks. Elke week dan liet ik hem mijn weekrapport zien en tekenen. Ook hebben we wel eens een paar kleine gesprekjes gehad. Eveneens werd hem wel door mij maar voornamelijk door Peter dingen gevraagd over het ExpertSysteem. Dit waren dingen als de logica en voor mij het eruit zien van de RelationEditor en zijn functionaliteit. Hij kwam tevens met het idee om relaties te kunnen ‘uit-’ en ‘inklappen’, zodat er bij het opstarten van de RelationEditor niet alle tabellen op het scherm staan waardoor het minder druk wordt.

Rob Sabelis zorgde voor het maken van een account voor mij op het netwerk van de sectie en als ik vragen of problemen met Windows NT of Novell had, kon ik hem altijd vragen. Ook heeft hij mij geholpen met de opdrachten voor de stage over de RuG.

Verder was de sfeer op de sectie wel goed. Meestal was het toch behoorlijk rustig omdat iedereen zelfstandig aan het werk was. Maar tegen het einde van de dag werden er vaak wel MP3'tjes gedraaid. Wat me wel tegenviel in het begin was om de hele dag achter de computer te zitten.

5.2. Mijn PPV-plan

In deze paragraaf zal ik per PPV-punt, die ik voor de stage heb aangegeven, bespreken of dit daadwerkelijk gelukt is:

- *Initiatief tonen*
Ik heb hiervoor niet echt de mogelijkheid gehad dit te verbeteren, omdat er niet veel gelegenheden hiervoor waren. Maar wel heb ik af en toe een e-mail naar mijn stagedocent gestuurd. En als ik tijdens een opdracht ‘vast’ kwam te zitten ging ik wel gewoon bij Peter vragen om hulp. En elke week liet ik mijn stagementor het weekrapport zien.
- *Assertief zijn*
In het begin van de stage moest ik eerst het ExpertSysteem leren kennen. De RelationEditor was voor mij een uitdaging en toen het af was ik er ook wel trots op en tevreden over.
- *Doorzettingsvermogen bezitten*
Bij het stuiteren op fouten probeerde ik deze eerst zelf op te lossen en te zoeken naar oplossingen in de documentatie of op internet. En als dat niet lukte ging ik naar Peter.

- *Creativiteit, originaliteit tonen*

Ik heb niet echt veel creativiteit getoond tijdens mijn stage, omdat dit ook niet echt nodig was.

Wat ik wel vrij goed deed was het rapporteren, een probleemoplossende houding hebben, doorzettingsvermogen bezitten, zelfstandig kunnen werken en kwaliteitsbewustzijn bezitten.

5.3. Mijn PPV-plan voor de toekomst

De professioneel persoonlijke vaardigheden van mij en de meeste studenten zijn wel uit te breiden. Omdat ik tijdens mijn stage niet aan alle PPV-punten kon werken, simpelweg omdat sommige punten niet aan bod kwamen tijdens de stage (bijvoorbeeld vergaderen en samenwerken), zal ik deze in de toekomst van mijn studie moeten verbeteren. Ik zal hier de PPV-punten noemen die ik in de toekomst zou moeten veranderen en hoe ik dit van plan ben te doen:

- *Initiatief tonen*
Omdat dit in de stage niet erg veel aan bod is gekomen moet dit nog wel wat verbeterd worden. Dit zou ik vooral kunnen doen in projecten en misschien in bepaalde practicalessen
- *Assertief zijn*
Dit zou ik nog wel wat kunnen verbeteren. Tijdens projecten en practica misschien.
- *Creativiteit, originaliteit tonen*
Dit zou ik natuurlijk bij veel dingen kunnen doen maar vooral bij practica en projecten.

Er zijn ook bepaalde punten waar ik nog niet veel mee te maken heb gehad en daarom ook niet zeker weet of ik deze zou moeten verbeteren:

- *taken kunnen verdelen;*
- *kunnen coördineren;*
- *kunnen omgaan met conflicten;*
- *kunnen onderhandelen;*
- *ondernemend ambitieus zijn.*

6. Conclusie

Al met al was het toch een interessante stage waar ik vrij veel van geleerd heb. Op de afdeling kon ik goed werken, het was er altijd rustig. En als ik wat wou vragen kon ik dat ook doen. Voor programmeervragen ging ik naar Peter Veentjer en voor ander vragen kon ik bij Rob Sabelis terecht.

En aan het einde van die negen week vond ik het toch snel gaan, dit kwam waarschijnlijk vooral omdat ik me vrijwel nooit verveelde op mijn stage. Ik had er altijd wel iets te doen qua programmeren en dat lukte meestal ook wel goed, want als het programmeren niet lukt dan kan het nog zeer vervelend werk zijn.

De opdrachten die ik voor het ExpertSysteem moest maken waren over het algemeen leuk om te doen en de meeste ook vrij leerzaam.

De punten voor de Professionele Persoonlijke Vaardigheden heb ik niet allemaal volledig gehaald. Ik heb wel punten aangegeven die ik in de toekomst zou moeten verbeteren.

Voor mijn volgende stage zou ik wel weer zoiets willen, alleen met wat meer afwisseling. Want ik zit nu vrijwel de hele dag achter de computer te programmeren. Hoewel ik de laatste weken het wel iets afwisselender had met de RMI-server en dergelijke.

Literatuurlijst

Vey Mestdagh, C.N.J. de, *Rekentuig of rekenmeester? Het onderbrengen van juridische kennis in een expertsysteem voor het Milieuvergunningenrecht*, Deventer, 1997.

Vey Mestdagh, C.N.J. de, *Samenvatting Expertsysteem Milieuvergunningenrecht*, z.pl., z.j.

Sinnema, J., *ExpertSysteem Milieuvergunningenrecht; Verslag van een stageproject in het kader van de studie Hogere Informatica aan de Noordelijke Hogeschool Leeuwarden*, Leeuwarden, 2001

Rijksuniversiteit Groningen, 'Rijksuniversiteit Groningen', 2001, <http://www.rug.nl>

Rijksuniversiteit Groningen Sectie Rechtsinformatica, 'Sectie Rechtsinformatica', 2001, <http://rint.rechten.rug.nl>

Bijlage 1 – Weekrapport 4

Bijlage 2 – Weekrapport 5

Bijlage 3 – Weekrapport 6

Bijlage 4 – Weekrapport 7

Bijlage 5 – Weekrapport 8

Bijlage 6 – Weekrapport 9

Bijlage 7 – Organogram RuG

