



Stichting NIOC en de NIOC kennisbank

Stichting NIOC (www.nioc.nl) stelt zich conform zijn statuten tot doel: het realiseren van congressen over informatica onderwijs en voorts al hetgeen met een en ander rechtstreeks of zijdelings verband houdt of daartoe bevorderlijk kan zijn, alles in de ruimste zin des woords.

De stichting NIOC neemt de archivering van de resultaten van de congressen voor zijn rekening. De website www.nioc.nl ontsluit onder "Eerdere congressen" de gearchiveerde websites van eerdere congressen. De vele afzonderlijke congresbijdragen zijn opgenomen in een kennisbank die via dezelfde website onder "NIOC kennisbank" ontsloten wordt.

Op dit moment bevat de NIOC kennisbank alle bijdragen, incl. die van het laatste congres (NIOC2023, gehouden op donderdag 30 maart 2023 jl. en georganiseerd door NHL Stenden Hogeschool). Bij elkaar bijna 1500 bijdragen!

We roepen je op, na het lezen van het document dat door jou is gedownload, de auteur(s) feedback te geven. Dit kan door je te registreren als gebruiker van de NIOC kennisbank. Na registratie krijg je bericht hoe in te loggen op de NIOC kennisbank.

Het eerstvolgende NIOC vindt plaats op donderdag 27 maart 2025 in Zwolle en wordt dan georganiseerd door Hogeschool Windesheim. Kijk op www.nioc2025.nl voor meer informatie.

Wil je op de hoogte blijven van de ontwikkeling rond Stichting NIOC en de NIOC kennisbank, schrijf je dan in op de nieuwsbrief via

www.nioc.nl/nioc-kennisbank/aanmelden_nieuwsbrief

Reacties over de NIOC kennisbank en de inhoud daarvan kun je richten aan de beheerder:

R. Smedinga kennisbank@nioc.nl.

Vermeld bij reacties jouw naam en telefoonnummer voor nader contact.

Computerondersteund Onderwijs: interactie tussen snelle jongens en artificiële intelligentie

Frans A.M. Schaars

K.U. Nijmegen, Recht en Informatica (tot en met 30-10-1990)

Hogeschool Katholieke Leergangen Tilburg (per 1-11-1990)

Postbus 90900

5000 GA Tilburg

Samenvatting

In deze bijdrage wordt aandacht besteed aan enkele problemen en mogelijke oplossingen binnen c.o.o.

Als we gangbare opvattingen over c.o.o. vergelijken met onze ervaringen uit het AMOR-R.I.N.project^[1] en het project Briefopener^[2], blijkt dat er twee uitersten zijn aan te wijzen binnen het in Nederland verrichte onderzoek naar c.o.o. Noch de op het A.I.paradigma gebaseerde theorie noch een radicale pragmatische aanpak bieden voldoende houvast om onze bevindingen systematisch te beschrijven.

We laten zien dat we wat betreft de theoretische uitgangspunten van beide projecten niet uit de voeten kunnen met de eisen zoals die binnen theoretisch onderzoek naar A.I.-achtig c.o.o. worden geformuleerd. Omdat we menen dat de interactie een veel zinvoller uitgangspunt is voor theorievorming stellen we voor een eenvoudig communicatiemodel als uitgangspunt te nemen voor de beschrijving van theoretische aspecten van c.o.o. die we op een drietal punten uitwerken.

1 Inleiding

Omdat bij grotere c.o.o.projecten zoals AMOR-R.I.N. en Briefopener een systematische werkwijze noodzakelijk is, is ook theoretische bezinning vereist. Dat is zo vanzelfsprekend dat we in deze bijdrage geen aandacht zullen besteden aan de opvatting dat niet systematisch gezocht hoeft te worden naar goede ideeën.

Momenteel bestaat AMOR-R.I.N.^[3] uit een serie vragen die de indeling van het boek *Rechtsvorming in Nederland* (vgl. noot 1) volgen. Via menu's kiest de student eerst welk hoofdstuk, dan per hoofdstuk via submenu's welke paragrafen gerepeteerd moeten worden. Als een onderdeel gekozen is, start de repetitie en stelt een reeks van vragen, waarna het programma in een menu terugkeert. De

student kan dan een ander hoofdstuk kiezen, een andere overhoring of het programma beëindigen. De opzet van *Briefopener* komt in hoofdlijnen overeen met die van AMOR-R.I.N., zij het natuurlijk dat er nu geen paragrafen uit een leerboek, maar een achttal casus bestudeerd worden, op basis waarvan een specifieke brief, bijvoorbeeld een offerte of een sollicitatiebrief, in een samenspraak tussen gebruiker en computer opgesteld wordt.

2 Theoretische aspecten

In het kader van het AMOR-R.I.N.project hebben we ondermeer gewerkt aan een c.o.o.systeem dat in hoge mate onderscheid maakt tussen enerzijds een onderwijskundig systeem (waarin verschillende typen vraag zijn ontwikkeld waarmee kennisoverdracht kan geschieden) en anderzijds een juridisch-inhoudelijk gedeelte dat bestaat uit 1) een vraag op het juridisch terrein, soms voorafgegaan door inleidende tekst, 2) een combinatie van testen van de antwoorden, 3) feed-back en 4) doorverwijzingen naar vervolgvragen. Door de object-onafhankelijke opzet van het onderwijskundige systeem konden een aantal AMOREenheden probleemloos overgeheveld worden naar de *Briefopener*-applicatie.

2.1 Gangbare opvattingen over c.o.o.: twee uitersten

Recente literatuur met betrekking tot c.o.o.programma's spreekt in dit opzicht tamelijk dwingend van componenten en verwacht van een c.o.o.programma dat er bijvoorbeeld een domeincomponent, een leerlingcomponent, een interactiecomponent en een onderwijskundige component aanwezig moet zijn, wil men spreken van een 'intelligent onderwijssysteem' [4].

Dergelijke algemeen geformuleerde eisen ten aanzien van c.o.o.programma's lijken rechtstreeks ontleend te zijn aan onderzoek op het gebied van artificiële intelligentie [5]. Binnen de huidige aandachtsvelden van juridisch c.o.o. blijkt dat bijvoorbeeld uit Van den Berg e.a. (1988) [6] waar onderzoek naar juridische kennissystemen beschreven wordt in eenzelfde context als die waarin onderzoek naar juridisch c.o.o. wordt geplaatst c.q. beschreven. [7]

Die eisen zijn methodologisch vooral van belang voor de ontwikkeling van c.o.o.theorie, maar gaan al snel een eigen leven leiden, hetgeen de ontwikkeling van c.o.o. niet per definitie bevordert. Te weinig oog voor de concrete c.o.o.praktijk en te veel aandacht voor theorievorming blijkt nu al in de c.o.o.-werkelijkheid. [8] Het andere uiterste wordt verwoord door medewerkers van de R.U. Limburg, die op het standpunt staan dat er zo snel mogelijk toepassingen voor de computer bedacht en uitgewerkt moeten worden [9]. Fernhout e.a. (1988) [10] waar-

schuwen voor het debacle van de vertaalprogramma's en claimen expliciet een *pragmatische aanpak*, waarvoor zij een viertal punten aangeven.

Naar mijn mening behoort een concreet c.o.o.project zich tussen deze beide uitersten te plaatsen. Theoretische bezinning en een systematische werkwijze zijn zonder meer noodzakelijk. Dat betekent echter niet dat ze hoofddoel van de c.o.o.werkzaamheden zullen zijn. Immers, een c.o.o.project wil een concrete bijdrage aan het onderwijs leveren. Het is precies deze intentie die maakt dat de eis van de pragmatische aanpak gesteld wordt. Het gevaar dat zich dan ogenblikkelijk aandient, is dat programmeurs steeds meer aandacht aan oppervlakkige zaken besteden zoals schermopbouw en leuk ogende animaties, zonder zich te bezinnen over de formalisatie van de interactie tussen docent en leerling.

2.2 Bezwaren tegen de c.o.o.theorie die zich baseert op het A.I.paradigma

In een c.o.o.programma moet de kennis zodanig zijn opgenomen, dat het programma via interactie ook uitleg kan geven. Als dat is gerealiseerd spreekt men volgens het A.I.paradigma van een domeincomponent. Binnen een automatische repetitor blijkt de domeincomponent zich echter op meer dan één plaats te bevinden. Zo zit specifieke juridische kennis opgesloten in de vraag die de repetitor aan de student stelt. Maar de juridische kennis kan ook opgenomen zijn in de feed-back. Dit kan zowel gebeuren wanneer uitleg over kennis als ook wanneer uitleg over gebrek aan kennis relevant is. Maar de juridische kennis moet ook gerelateerd kunnen worden aan de inhoud van de antwoorden die de student geeft. Bij een 'bijna helemaal goed' antwoord komt een andere reactie als bij een 'bijna helemaal fout' antwoord.

Uit onze praktijkervaringen blijkt dat de theoretische construct domeincomponent binnen het juridisch c.o.o. zich nooit als een eenheid manifesteert. Daarmee verliest het binnen de theorievorming zijn externe waarde. Maar ook de interne eisen die vanuit het A.I.paradigma gesteld worden, zijn niet altijd relevant. Pijls en Sandberg (1989:10) de eisen die o.a. aan de domeincomponent gesteld zouden moeten worden. Op p.15 stellen zij dat de werkelijkheid ver achter loopt bij het geschetste ideaal. Onbedoeld roepen de auteurs de vraag op of het geschetste ideaal wel levensvatbaar is als het (zoals zij zelf voorstellen) inhoudelijk wordt vergeleken met 'de rest van dit boek'. Het relevante punt zijn de noodzakelijke eisen, die o.i. allerm minst noodzakelijk hoeven te zijn. Met name de vereisten m.b.t. oplossingsstrategieën zijn zonder afdoende argumentatie normatief voor de ordening van kennis in een domeincomponent opgenomen. Overtuigen doen ze echter allerm minst als het onderwijssysteem een

repetitorfunctie moet vervullen. Het aanleren van oplossingsstrategieën is dan nl. zeker geen hoofddoel. Uit onze werkzaamheden blijkt dat de domeincomponent (maar die niet alleen) in de praktijk niet één fysieke eenheid binnen het programma vormt, maar verdeeld wordt over verschillende plaatsen. Daar hebben de opstellers van het 'ideaal' niet bij stil gestaan. Tevens dringt zich in dit verband de vraag op: zijn het de componenten uit het A.I. paradigma en geen andere die de basis moeten vormen voor de theorie met betrekking tot c.o.o.? Op basis van onze ervaringen zijn wij geneigd deze vraag met NEE te beantwoorden. Daarnaast willen wij belangstellenden op het gebied van c.o.o. deze vraag voorleggen: is het niet zinvoller om de interactie tussen computer en student als uitgangspunt te nemen voor theorievorming van c.o.o.? Voor c.o.o.toepassingen als een repetitor is dat zeker gewenst.

2.3 Het nieuwe uitgangspunt: de interactie

Laten we de concrete communicatie bij c.o.o. eens nader bekijken. Bij een automatische repetitor zien we dat de basissituatie is:

- 1) De computer stelt een vraag om kennis te toetsen.
- 2) de student typt een reactie in.
- 3) de computer
 - a) evalueert het gegeven antwoord.
 - b) geeft de beoordeling van dat antwoord, waarbinnen inhoudelijke feedback gegeven kan worden, bijvoorbeeld als het antwoord niet helemaal goed is.
 - c) bepaalt afhankelijk van het gegeven antwoord welke volgende vraag gesteld zal worden. In principe keert het systeem dan weer terug naar 1.

Nu schuilt de kracht van een repetitor niet zozeer in de domeincomponent alswel in het samenspel tussen de factoren a, b en c zoals die onder nr. 3 zijn gegeven. Hoe precieser de evaluatie door de computer kan worden gemaakt, des te precieser kan de feedback worden en des te adequater de verwijzing naar een volgende vraag. De eisen voor een intelligent systeem zullen dan ook op dit terrein geformuleerd moeten worden.

2.4 Analyse van de communicatie leidt tot een strategie van tests

Een eerste mogelijkheid om aan die vereiste gehoor te geven is de strategie van de tests die onder 3.a door de computer worden uitgevoerd.

Uitgangspunt in het nuvolgende is dat de computer binnen de doelstellingen van een c.o.o.programma de volgende vraag mag stellen: Wat is de hoofdstad van Nederland? De vraag lokt een reactie van de student uit, die in een aantal gevallen voorspelbaar is. Maar die reactie kan ook niet-voorspelbaar zijn. Een serie tests moet nu in staat zijn deze communicatie te herkennen.

Onze strategie van de tests is deze: ten eerste testen we op het goede antwoord. Vervolgens geven we een test (c.q. serie tests) die voorspelbare alternatieven herkent. Hier is het bepaald zinvol om voorspelbare FOUTE antwoorden in de serie evaluatietesten op te nemen.

Binnen onze programma's maken we steeds de test op antwoorden als "Geen idee" via patroonherkenning (vgl. noot 2). In die situatie zou de computer dan bijvoorbeeld een hint kunnen geven, of een multiple-choice-menu kunnen toevoegen aan de vraag.

Als laatste test is opgenomen de test ARBITRAIR. We denken dan aan antwoorden van de student die niet voorzien waren. In dat geval zou een communicatiestoornis dreigen en daarom laten we de computer de vraag herformuleren. De werkwijze is vergelijkbaar met die bij de test met het patroon WEETNIET.

Op deze wijze ontstaat per vraag een piramide van tests. De top is het volledig correcte antwoord, het middengedeelte bestaat uit voorspelbare antwoorden, en de bodem wordt gevormd door de tests WEETNIET en ARBITRAIR.

2.5 Strategische tests vereisen bezinning over het type vraag dat gebruikt gaat worden

Toch zou deze hele strategie weinig functioneel zijn als we te lichtzinnig omspringen met de vraag zoals de computer die in de begin-situatie op het scherm zet.

De vragen die een computer kan stellen zijn oneindig. De antwoorden die een student kan geven zijn dat ook. Tussen die werelden van oneindigheid moet een c.o.o.programma noodzakelijk duidelijk lijnen trekken. Nogmaals wil ik hier wijzen op de pragmatische aanpak die c.o.o. vereist. Want binnen deze twee uitersten moet binnen afzienbare tijd een werkbaar programma gerealiseerd zijn. Dat betekent echter geenszins dat er geen bezinning hoeft plaats te vinden over de te verwachten interactie. Daarom willen we in deze paragraaf nader ingaan op het type vraag dat de computer stelt.

Een op het eerste gezicht eenvoudig type vraag is de JA/NEEVraag. Stel dat onze beginvraag een JA/NEEVraag is: "Is Amsterdam de hoofdstad van Nederland?" Op deze vraag zijn twee typen antwoord voorspelbaar. De eerste is het type JA/NEE. Maar er kunnen ook antwoorden komen in de trant van: "Inderdaad, dat is sinds de grondwetsherziening van 1983 het geval" of "Nee, totaan de grondwetsherziening van 1983 was dat geen uitgemaakte zaak".

Met het eerste type vraag (JA/NEE) koos de student voor een antwoord binnen een onderwijskundig model. We beschouwen de varianten op JA en NEE dan ook als onderdelen van de niet-domeingeboden interactie, omdat ze los staan van de juridische relevante informatie.

Daarentegen is het tweede type een antwoord uit het domein. Bij dat antwoord behoort bekeken te worden of het "Nee, totaan de grondwetsherziening van 1983 was dat geen uitgemaakte zaak" als zinvol antwoord geaccepteerd mag worden of niet.

Ook de onderwijscomponent uit het A.I.paradigma blijkt in de praktijk niet als een eenduidig af te bakenen eenheid binnen het programma op te treden. Zelfs bij een eenvoudig voorbeeld wordt hij al vertroebeld door onderdelen van de domeincomponent. Immers het tweede antwoord bevat vakinhoudelijke kennis en behoort als zodanig herkend te worden.

Dat stelt de ontwerper van c.o.o. die meende een eenvoudige JA/NEEVraag te hebben opgesteld voor onverwachte complicaties. De evaluatie door het programma van het antwoord moet immers voor de paradox van het NEE (onderwijscomponent) en het NEE uit "Nee, tot de grondwetsherziening van 1983 was dat geen uitgemaakte zaak" (domeincomponent) een oplossing bieden. Dit betekent dat een 'intelligent' programma over een sterke interactiecomponent moet kunnen beschikken. Er zal dan een test moeten bestaan die uit het antwoord van de student niet alleen het NEE maar ook relevante informatie als GRONDWETSHERZIENING en 1983 op de juiste manier moet kunnen interpreteren. Dat vergt veel tijd omdat niet snel is te overzien wat voorspelbare antwoorden zijn. Bedenk: het gaat hier alleen nog maar om voorspelbare GOEDE antwoorden. Want een c.o.o.programma moet een voorspelbaar onjuist antwoord als: "Nee, volgens mij is dat Den Haag, omdat regering en vorstin daar gevestigd zijn." kunnen onderscheiden van een niet-voorspelbaar antwoord als "STOKKUM".

Belangrijkste conclusie is, dat de te verwachten interactie een veel adequater afbakening van theoretische én praktische problemen geeft, dan de op A.I. gebaseerde terminologie.

Thans wil ik nader ingaan op de interactie aan de hand van onze beginvraag: "Is Amsterdam de hoofdstad van Nederland?"

In het hierboven gestelde perspectief is de beste oplossing: kies een ander type vraag. Een van de mogelijkheden is om er een OPEN VRAAG van te maken, bijvoorbeeld: Wat is de hoofdstad van Nederland?

Afhankelijk van de gegeven antwoorden kan nu veel effectiever relevante feed-back worden gegeven. Naast het goede antwoord is nl. het aantal voorspelbare foute antwoorden systematisch te overzien: het antwoord behoort in ieder geval de naam van een hoofdstad te hebben.

Daarmee hebben we een criterium waarmee we de tests kunnen samenstellen. Die tests hebben het meeste rendement wanneer ze piramidaal zijn opgebouwd. Niet alleen werkt het programma sneller omdat de computer erop ingesteld is, eerst het goede antwoord te testen (de top van de piramide), zodat hij onmiddellijk naar een volgende vraag kan. Maar ook kunnen de tests er zo zorg voor dragen dat andere uitleg over de kennis wordt gegeven bij het antwoord ZWOLLE als bij het antwoord BERLIJN. De feed-back kan bij deze soort antwoorden gerelateerd worden aan het antwoord zelf. Bij Zwolle zou bijvoorbeeld gewezen kunnen worden dat het weliswaar een hoofdstad is, maar slechts van een provincie, niet van een land. Mutatis mutandis (programmeertechnisch gezien, niet politiek) kunnen we andere feed-back bij het antwoord BERLIJN geven. Toch is daarmee niet ieder element van de dialoog ondervangen. Immers, een antwoord als: STOKKUM zal als niet-voorspelbaar antwoord worden aangemerkt. Daarmee bereiken we de basis van de piramide: de ARBtest moet dan in werking treden. In het concrete geval zou dat kunnen betekenen dat de computer de vraag herformuleert en de OPEN VRAAGsituatie loslaat en vervangt door een meer rigide type vraag, bijvoorbeeld een MULTIPLE-CHOICEvraag. In dat geval mag van de student nl. verwacht worden dat het antwoord zal zijn: A, B, C, D of E, zoals uit dit voorbeeld blijkt:

Wat is de hoofdstad van Nederland?

Is dat:

- A. Den Haag
- B. Berlijn
- C. Amsterdam
- D. Didam
- E. Zwolle?

Op basis van het piramidale model is het mogelijk de aanvankelijke open vraag "Wat is de hoofdstad van Nederland?" bij een onvoorspelbaar antwoord als STOKKUM te reduceren tot een multiple-choicevraag, waar nog maar een beperkt aantal alternatieven relevant is.

In de nu geldende tests moet naar voren komen dat de computer "begrijpt" welk gebrek aan inzicht er bij de student is, als het antwoord keuze A is. De feed-back zal bij de vier onjuiste antwoorden steeds anders moeten zijn. Enkel feed-back als: "U maakte een onjuiste keuze. Nog eens a.u.b." is niet acceptabel: de feed-back behoort relevante kennis te geven. Bij keuze A is het bijvoorbeeld zinvol om in te gaan op de misvatting dat als hoofdstad die stad wordt aangemerkt, waar de regering zetelt c.q. waar de koningin woont.

Een ander type vraag is het PONEREN VAN STELLINGEN: de student wordt geconfronteerd met twee stellingen. Verwacht wordt dat hij aangeeft of de stellingen goed dan

wel fout zijn. Met behulp van die stellingen kan worden nagegaan in hoeverre de student iets heeft begrepen van de tot op dat moment behandelde stof. Bij een volstrekt goed antwoord zal het programma verder gaan, bij een foutief antwoord zal kenbaar gemaakt worden welke onderdeel niet goed wordt beheerst en dat de computer voornemens is op dat specifieke onderdeel de repetitie te herhalen. Afhankelijk van de leerstof zal dat bijvoorbeeld een meer uitgebreide repetitie kunnen zijn. Tot nu toe hebben we aandacht besteed aan deze typen vraag: de JA/NEE vraag, de OPEN vraag, de MULTIPLE-CHOICE-vragen en de vraag om te reageren op twee STELLINGEN. Voorts maken we gebruik van een test voor het geval de student het goede antwoord niet weet: daarvoor gebruiken we het patroon WEETNIET en als laatste test wordt steeds het patroon ARB voor ARBITRAIR gebruikt: het vangnet dat ALTIJD slaagt, zodat het programma doorgestuurd kan worden naar een passend vervolg. Via deze zes typen vraag zijn onze applicaties ontwikkeld die lineair zijn opgebouwd en waarbinnen de student naar eigen keuze een overhoring kan uitkiezen. Ten slotte vermeld ik nog de optie HINTS. Afhankelijk van het object dat gerepeteerd wordt, kunnen er hints gegeven worden, die met name in onze praktijk gerelateerd worden aan de tests op het antwoord WEETNIET en ARB. Het betreft hier dan een of meer aspecten van kennis die gegeven worden voordat de computer de vraag herformuleert in een multiple-choice-vraag, dan wel het volledig goede antwoord zelf geeft. Afhankelijk van de gekozen driver kan de overhoring door de combinatie testen, feed-back en vervolgvragen een bijzonder 'slimme' indruk maken. Er kan dan met recht gesproken worden van intelligente onderwijssystemen.

3 Interactie-model of leerlingcomponent als uitgangspunt voor theorievorming?

De uitwerking van de theoretische construct 'leerlingcomponent' binnen beide AMOR-applicaties leidt opnieuw tot kritische vragen ten aanzien van het gebruik van het A.I.paradigma. Kern van deze paragraaf is, dat ook de leerlingcomponent geen eenduidig onderdeel van een c.o.o.programma is. Ook deze component wordt in bijna alle gevallen vermengd met elementen uit het domein. In het voorafgaande zagen we dat de piramidale opbouw van tests zeer specifieke feed-back gegeven kan worden. De combinatie VRAAG, ANTWOORD en EVALUATIE zijn onderdeel van een eenvoudig communicatiemodel. Precies dat eenvoudige model gaf heel nauwkeurig aan waar welke onderdelen van het te overhoren object zijn ondergebracht. Als we nu kijken naar wat men theoretisch aanduidt met 'leerlingcomponent' dan blijkt andermaal dat een ander uitgangspunt voor theorievorming dan het paradigma van de A.I. een eenvoudiger beschrijvingsmodel oplevert. Uit de

beide AMORapplicaties is gebleken dat de kern van de 'leerlingcomponent' is gelegen in de opbouw van de tests. Immers, de onder 3-a genoemde tests van het gegeven antwoord kunnen dank zijn patroonherkenning (vgl. noot 2) resulteren in het onderscheid maken tussen bijvoorbeeld:

1. het perfecte antwoord.
2. het als goed te rekenen antwoord.
3. het voorspelbare foute antwoord, dat als FOUT antwoord niet al te 'ernstig' is.
4. het voorspelbare foute antwoord, dat als FOUT antwoord ernstiger is dan onder 3 gegeven.
5. Volstrekt foutieve, voorspelbare antwoorden.
6. de reactie van de student dat kennis niet aanwezig is.
7. niet-voorspelbare antwoorden.

Een voorbeeld: ter inleiding op de overhoring over de codificatie (Tak 1987:51 e.v.) is de nuvolgende vraag gemaakt. (Men neme er nota van dat deze vraag in eerste instantie als een open vraag was gepresenteerd.) In een aantal gevallen (bijvoorbeeld wanneer de test WEEETNIET dan wel ARB slagen) is het gewenst dat de computer de vraag herhaalt, maar nu voorzien van een multiple-choice-gedeelte, waarmee de computer het verdere verloop van de overhoring naar zich toe trekt. In tweede instantie zal dan het nuvolgende op het scherm komen:

Door de diversiteit van het recht aan het eind van de 18de eeuw bestond er een grote mate van willekeur bij de gerechten. Dit bracht voor de burgers *rechts*.... met zich mee.

Bedoeld zijn:

1. rechtsgelijkheid en rechtszekerheid
 2. rechtsgelijkheid en rechtsonzekerheid
 3. rechtsongelijkheid en rechtszekerheid
 4. rechtsongelijkheid en rechtsonzekerheid
- welk van de vier mogelijkheden is juist?

Een groot aantal antwoorden is in dit geval voorspelbaar. Goed is: '4. rechtsongelijkheid en rechtsonzekerheid'.

De student kan volstaan met een antwoord volgens het onderwijskundige model. Dit betekent dat allerlei variaties op de tekst: 'optie 4 is goed' in de onderwijscomponent aanwezig moeten zijn. Ook enkel het antwoord '4' moet daar als 'goed' herkend kunnen worden.

Maar ook zijn allerlei variaties op de woorden 'rechtsgelijkheid' en 'rechtsonzekerheid' als antwoord te verwachten. De termen kunnen bijvoorbeeld in omgekeerde volgorde worden gegeven en men kan volstaan met enkel 'onzekerheid' en 'ongelijkheid' als antwoord te geven, omdat immers in de vraag zelf stond: '*rechts*...'

Deze eenvoudige situatie stelt de A.I.achtige theorie met betrekking tot c.o.o. al tot onontwarbare problemen, omdat onderwijskundige aspecten samenvloeien met elementen uit het domein en de 'leerlingcomponent'. Letten we echter op de interactie, dan kan de situatie veel eenduidiger omschreven worden:

Er is een open vraag gesteld, waarop een antwoord kwam, waarop de computer niet was voorbereid. Om aan die communicatiestoornis het hoofd te bieden, laten we de computer de vraag opnieuw stellen, nu vergezeld van een multiple-choice-systeem. De reactie van de student kan nu volgens het onderwijskundige model gaan, maar kan ook inhoudelijk van aard zijn. Via de patronen (vgl. noot 2) moeten we dan in de piramidaal opgebouwde tests de communicatie ondervangen. Dit betekent:

1. de test op het perfecte antwoord, dat zowel onderwijskundig als inhoudelijk kan zijn. Bijvoorbeeld 4 (onderwijskundig) of *rechtsongelijkheid* en *rechtsonzekerheid* (inhoudelijk).
2. de voorspelbare foute antwoorden.
3. de reactie van de student dat kennis niet aanwezig is.
4. niet-voorspelbare antwoorden.

Op beide typen van antwoord (onderwijskundig c.q. juridisch-inhoudelijk) kan binnen AMOR adequaat worden gereageerd door middel van patroonherkenning.

Door het onderwijskundige systeem van vragen stellen in eerste instantie los te koppelen van het juridisch-inhoudelijke gedeelte van het programma konden via patronen tests ontwikkeld worden die razend snel gebrek in kennis bij de student konstateren. Het onderwijskundige systeem is in hoge mate object-onafhankelijk, maar kan op bijzonder eenvoudige wijze in de evaluaties van de antwoorden gekoppeld worden aan specifieke (juridische) kennis.

Concreet:

```
ALS ANTWOORD((STELD (NULL | ZIJNGOED)) | (POSITIEF
ALS SPAT LID ALS ("RECHTS" | NULL) ("ONGELIJKHEID"
ALS ET ("RECHTS" | NULL)"ONZEKERHEID")) | ("ONZEKERHEID"
ALS ET ("RECHTS" | NULL) "ONGELIJKHEID")) ZIJN SPAT)
```

In het eerste gedeelte is het onderwijskundige gedeelte gegeven, nl. in (STELD (NULL | ZIJNGOED)). In het tweede gedeelte is de inhoudelijke informatie opgenomen. Het voert te ver om in dit bestek de syntaxis van de patronendefinities uit AMOR aan de orde te stellen, maar de lezer kan uit de tweede, derde en vierde regel in ieder geval herkennen dat de test zich o.a. oriënteert op lettercombinaties als 'RECHTS' 'ONGELIJKHEID' en 'RECHTS' 'ONZEKERHEID'.

Het voordeel van het ontwerpen van een interactiemodel dat in eerste instantie geheel zelfstandig functioneert is:

1. studenten kunnen efficiënt antwoord geven: korte antwoorden kunnen volstaan.
2. de programmeur beschikt zo over een bijna universeel te gebruiken onderwijskundig systeem omdat het inhoudelijk los staat van het te repeteren object
3. dank zij de patroonherkenning kan het systeem heel snel inhoudelijk 'gevuld' worden, waardoor de interactie ook op vakinhoudelijk terrein kan plaats vinden.

4 Verdere uitbouw van het communicatiemodel

Tot nu toe hebben we nauwelijks of geen aandacht besteed aan de onderlinge samenhang van vragen binnen een c.o.o.-programma. Een paar aspecten van mogelijke samenhang tussen de afzonderlijke vragen willen we hier nader ter sprake brengen.

4.1 Lineaire opbouw van een repetitie

Een menselijke repetitor zal zijn vragen zo stellen dat een vooropgesteld einddoel: het verwerven van kennis bereikt wordt. Een automatische repetitor zal zijn onderwijskundige opbouw op vergelijkbare wijze moeten bereiken. Binnen onze applicaties bleek het gewenst dat de automatische overhoring een duidelijke draad zichtbaar maakt, zodat de student zo veel mogelijk houvast heeft aan het leerboek dan wel de casus. Ons gaat het om het wegwerken van kennislacunes. Dat betekent dat het domein, alvorens het 'ingevoerd' wordt, zorgvuldig moet zijn geanalyseerd in hoofd- en bijzaken, oorzaak- en gevolgverbanden, opdat straks de repetitor stap voor stap op relevante wijze kennis met de student doorneemt.

4.2 Tussentijdse evaluaties

Evenals in de normale onderwijssituatie kan ook tijdens een oefensessie met de computer tussentijds getest worden of de gerepeteerde kennis voldoende is opgenomen. Voor deze situatie is met name het type vraag geschikt dat de student confronteert met twee stellingen.

4.3 Samenvatting van de gehouden repetitie

Binnen het AMOR-R.I.N.programma is ervoor gekozen om na afloop van iedere samenhangende reeks vragen de mogelijk-

heid aan te bieden een samenvatting van de behandelde stof op het scherm te laten verschijnen. In die samenvatting wordt de lijn duidelijk van de afzonderlijke vragen die in precies die ene overhoringssessie zijn gesteld. Briefopener sluit de drie hoofdonderdelen per casus af met een presentatie van de relevante gegevens die desgewenst geprint kunnen worden, opdat de gebruiker bij de volgende fase die gegevens bij de hand heeft dan wel aan het eind van het programma de hele brief op papier heeft.

5 Afsluiting

In deze congresbijdrage hebben we tot nu toe laten zien dat de theoretische uitgangspunten voor een juridisch c.o.o.programma niet goed uit de voeten kan met eisen zoals die binnen theoretisch onderzoek naar A.I.systemen worden geformuleerd. Veeleer is de praktijk dat de interactie vertrekpunt is voor theorievorming. Vervolgens hebben we laten zien hoe we op basis van een eenvoudig communicatiemodel een serie VRAGEN hebben ontworpen, die ieder voor zich gevolgd worden door een piramide van tests, waardoor de interactie niet stagneert. Daarmee is een aantal 'tools' tot stand gekomen die telkens weer in de programmeerwerkzaamheden gebruikt kunnen worden, omdat ze in eerste instantie domein-onafhankelijk zijn. Briefopener is hiervoor het bewijs: het programma kon probleemloos van de object-onafhankelijke AMOR-R.I.N.modules gebruik maken.

Afhankelijk van de doelstellingen van het te ontwerpen c.o.o.programma kunnen op deze wijze sneller programma's ontwikkeld worden. De verscheidenheid van bouwmogelijkheden zijn echter zó groot, dat het niet mogelijk is, die in kort bestek overzichtelijk te presenteren. De keuze om AMOR te gebruiken als driver voor een tweetal repetitors is een keuze van ons, geen beperking van het systeem.

Noten

1. Schaars, F.A.M. (1988) AMOR-R.I.N. In: J.F. van Beek, M.J.B. Boerma en C.M.M. Hurts (red.) *Computer-Ondersteund Onderwijs in de Juridische Discipline*. Deventer: Kluwer 1988, 123-126 & 175.

Het project startte op 01-08-1988 aan de K.U. Nijmegen als onderwijsstimuleringsproject voor de tijd van twee jaar en beoogde de vervaardiging van een automatische repetitor bij Tak, P.J.P. (1987(2)). *Rechtsvorming in Nederland*. Alphen aan den Rijn: Samsom H.D. Tjeenk Willink / Open Universiteit. Op 1 november 1990 is het project om financiële redenen stopgezet.

2. Gerritsen, J., A. van den Hoorn, drs. B. Salemans, dr. F. Schaars en dr. O. Severijnen (1991) *Briefopener Computercursus voor het schrijven van zakelijke brieven* Leiden/Antwerpen: Martinus Nijhoff Uitgevers begon als een INSP-project en werd in 1990 door tussenkomst van de stichting LOC (Nijmegen) als een AMOR-applicatie vervaardigd. Daarbij kon gebruik gemaakt worden van de voor het AMOR-R.I.N.-project ontworpen object-onafhankelijke interactiemodel. Briefopener presenteert een schrijfmethode die niet alleen aandacht besteedt aan concrete schrijfgeregels vooraf, maar ook aan casusafhankelijke feedback tijdens de oriëntatie- en planfase van het schrijfproces. Deze begeleiding en sturing ontbreekt veelal in de huidige schrijfdidactiek, omdat ze een docent per individuele student zou vergen. Om die reden is besloten tot een computergestuurde cursus. Briefopener bestaat uit een cursusboek en uit een softwarepakket.

3. Zie: Coppen, P.A. & B. Salemans (1989) *AMOR AutoMatische OveRhorning en instructie* Leiden/Antwerpen: Stenfert Kroese. Inz. § 3.2. Patronen zijn een speciale programmeerfaciliteit binnen o.a. AMOR die bijzonder geschikt bleek voor het ontwerpen van het interactiemodel van het AMOR-R.I.N.-project.

Zie tevens: Schaars, F.A.M. (1988) *AMOR-R.I.N.* In: J.F. van Beek, M.J.B. Boerma en C.M.M. Hurts (red.) o.c. Inz. p. 126 (vgl. noot 1).

4. De begrippen 'domeincomponent', 'leerlingcomponent', 'onderwijscomponent' en 'interactiecomponent' als onderdelen van een onderwijssysteem zijn in hun ideale vorm, dus los van concreet onderzoek beschreven in Sandberg, J. & F. Pijls (sic!) (1989) *De computer onderwijst in: Pijls, F & J. Sandberg (red.), De computer als expert en didacticus* Muiderberg: Dick Coutinho, p.8-16.

5. Vgl. bv. het AI-paradigma in: Kempen, G.A.M. (1987) *Natuurlijke taal en kunstmatige intelligentie* Groningen

6. Van den Berg, P. & C.A.F.M. Grütters, O.W.M. Kamstra, T. van Willigenburg (1988) *RI-paradigmata Toekomst van Nederlands onderzoek in de rechtsinformatica* Lelystad: Koninklijke Vermande bv.

7. Vgl. bv. de bijdragen in Van den Berg e.a. (red.) o.c. van de universiteiten van Tilburg, Nijmegen, Groningen, Maastricht en Utrecht.

8. Vgl. in dit opzicht bv. Daelemans, W. (1989) *TDTDT onderwijst spellen* in: F. Pijls en J. Sandberg (red.) o.c. p.16-28, waar in § 2.1. *De domeincomponent* kennis over de Nederlandse morfofonologie is opgenomen die tot een beslissingsboom leidt. Deze boom is afgedrukt op p. 19. De zin

"We kochten het gisteren bekeken huis, toen we wisten dat het geld was gestort" zal door dit kennissysteem voor alle woordvormen die zijn afgeleid van werkwoorden als FOUTIEF worden aangemerkt!

9. Vgl. Fernhout, F.J. & T. van Willigenburg, M.J. Cohen, J. Schöpping, G. Span, W. Temme (1988) *De computer als nuttige slaaf* in: P. van den Berg, e.a. (red.) o.c. p.46-55, waar op p.50 wordt gezegd: "Daarbij geldt: 'anything goes'. 'Rapid prototyping' heeft natuurlijk de voorkeur, en van het moeizame volledig gebruiksklaar maken van een pakket heeft iedereen die het al een paar keer heeft moeten doen snel de schrik te pakken. Naar goede ideeën wordt niet systematisch gezocht."

10. Fernhout e.a. (1988) o.c. p.50-51.