

Drs J. E. M. van Dijk

Mogelijkheden en beperkingen van expertsystemen

1 Inleiding

Expertsystemen worden zo vaak als de meest succesvolle praktische toepassing van kunstmatige intelligentie genoemd (bijvoorbeeld in: Andriole (1985), Blanning (1984), Clement (1985) en Frenkel (1985)), dat men de indruk zou kunnen krijgen dat organisaties hopeloos achterlopen als ze nog niet hun eigen expertsysteem hebben ontwikkeld. Dat men het in het bedrijfsleven nodig vindt om op de hoogte te zijn van de ontwikkelingen op het gebied van kunstmatige intelligentie en speciaal op het gebied van expertsystemen blijkt ook uit het aantal deelnemers aan congressen over deze onderwerpen; het aantal deelnemers uit het bedrijfsleven is namelijk bij die conferenties de laatste jaren meer toegenomen dan het aantal deelnemers dat afkomstig is van universiteiten en onderzoeksinstituten (EDP Analyzer 12-1985).

Het is de bedoeling om in dit artikel de beperkingen en mogelijkheden van expertsystemen te bespreken. We veronderstellen geen voorkennis op dit gebied; daarom zullen we eerst een definitie geven van expertsystemen en uitleg geven over de componenten van een expertsysteem. Daarna gaan we in op het ontwerp van expertsystemen, waarna we het gebruik en de toepassingsgebieden behandelen. In de loop van het artikel zullen de volgende vragen aan de orde komen:

- Wat is het doel van expertsystemen?
- Wat onderscheidt expertsystemen van ‘traditionele’ informatiesystemen?
- Waarom zijn expertsystemen nu opeens populair?
- Wat zijn de voor- en nadelen van expertsystemen?
- Voor welke type problemen is een expertsysteem een geschikte oplossing?

2 Definitie en beschrijving van componenten

In Hayes-Roth e.a. (1983) wordt een expertsysteem gedefinieerd als: ‘een computerprogramma dat de kennis van één of meer expert(s) op een bepaald vakgebied bevat en die kennis kan gebruiken om problemen op dat gebied op te lossen’. Deze definitie schrijft geen bepaalde implementatievorm voor, het is alleen noodzakelijk dat er een menselijke expert bij de

ontwikkeling en/of validatie van het systeem betrokken is geweest. We noemen iemand een expert als hij met behulp van zijn kennis en ervaring problemen oplost die door weinig mensen kunnen worden opgelost. Door de schaarste aan experts ontstaan er onder andere grote problemen binnen een organisatie als een expert ziek wordt, overlijdt of wordt weggekocht door een concurrent.

Omdat de opbouw en daardoor de manier van werken van een expertsysteem in belangrijke mate verschilt van andere informatiesystemen, lijkt het ons aangewezen om eerst de interne structuur van een expertsysteem beknopt te behandelen. Wij onderscheiden in een op kennis gebaseerd expertsysteem de volgende componenten:

- 1 een kennisbank;
- 2 een redeneermechanisme;
- 3 een dialooggedeelte;
- 4 een werkblok en agenda.

2.1 De kennisbank

In een *kennisbank* bevinden zich feiten, omschrijvingen, vuistregels en relaties die nodig zijn om een bepaald *type* probleem op te lossen. Informatie over een *concreet* probleem wordt apart bijgehouden. In de praktijk worden drie structuren gebruikt om kennis op te slaan:

a Rules

Rules zijn net zoals de kolommen van een beslissingstabel opgebouwd uit een 'als' gedeelte en een 'dan' gedeelte. In het 'als' gedeelte staan één of meer voorwaarden; indien aan die voorwaarden is voldaan dan is het 'dan' gedeelte van toepassing.

Voorbeeld:

'Als' de klant meer dan 1000 eenheden bestelt of de afstand tot de klant kleiner is dan 10 kilometer

'Dan' komt de klant in aanmerking voor korting (hoe hoog de korting zal zijn wordt bepaald met behulp van andere regels uit de kennisbank).

Regels in een kennisbank kunnen met elkaar in verband staan doordat (één van) de conclusie(s) in regel A als een/de voorwaarde voorkomt in regel B, maar regel B zal niet vanzelfsprekend worden toegepast als de conclusie in regel A is getrokken. De controle over de volgorde waarin regels worden uitgevoerd valt namelijk onder het redeneergedeelte (zie 2.2).

b Frames

Frames worden op twee niveaus gebruikt, namelijk om een beschrijving te geven van een klasse van entiteiten en om specifieke objecten uit die klasse te beschrijven. Het algemene frame is verbonden met frames van specifieke objecten, bevat gegevens over de klasse als geheel en kan (prototype) gegevens over elementen uit die klasse bevatten. Als een nieuw frame voor

een specifiek object wordt gecreëerd kan de prototype-informatie worden overgeërfd. Er zijn overeenkomsten tussen frames en records van een gegevensbank; records hebben echter niet de mogelijkheid van overerving. Voor informatie over andere verschillen verwijzen wij naar Fikes&Kehler (1985). Een eenvoudig voorbeeld van een klasse-frame en een element-frame:

Huis frame

speciaal geval van: een gebouw
eigenaar: een persoon
doel: bewoning door eigenaar of huurder
type: keuze: (flat, villa, bungalow)

Villa frame

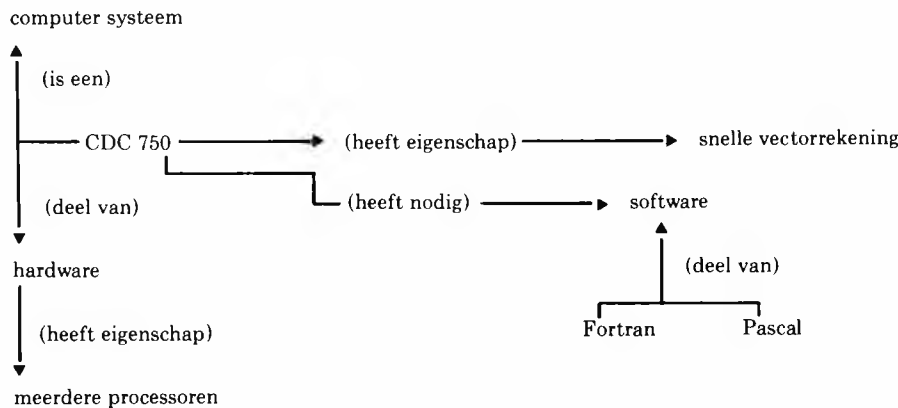
speciaal geval van: huis
eigenaar: Boshuizen
oppervlakte tuin: 10 are
aantal verdiepingen: 4
aantal bewoners: 6

Het villa frame is element van de klasse huizen. Het huis frame behoort tot de klasse gebouwen, en de klasse gebouwen zou op zijn beurt bijvoorbeeld kunnen worden gedefinieerd als een subklasse van de klasse fysieke objecten.

c Semantische netwerken

Semantische netten worden voorgesteld als een verzameling knooppunten die verbonden zijn door een verzameling bogen. Volgens De By (1985) worden de knooppunten gebruikt om entiteiten en situaties aan te duiden en geven de bogen relaties aan tussen de entiteiten in de knooppunten. Semantische netten hebben een aantal dezelfde eigenschappen als frames. Frames zijn echter (meer) gestructureerd.

Een voorbeeld van een semantisch net:



2.2 *Het redeneermechanisme*

Het *redeneergedeelte* is volkomen onafhankelijk van de inhoud van een kennisbank en kan dus voor meerdere kennisbanken worden gebruikt om al redenerend met behulp van de inhoud van zo een kennisbank tot een conclusie te komen. De functie van het redeneergedeelte is enigszins te vergelijken met de functie van een Database Management System met betrekking tot gegevensbanken. Er is echter een belangrijk verschil tussen het gebruik van een gegevensbank en het gebruik van een kennisbank. Een gebruiker van een gegevensbank leest, schrijft en/of muteert gegevens in een gegevensbank. De gebruiker van een expertsysteem heeft veel minder initiatief (Ford, 1985). Zijn belangrijkste taak is: antwoord geven op vragen die door het expertsysteem worden gesteld.

Twee methoden van redeneren zijn 'forward chaining' en 'backward chaining'. Bij achterwaarts redeneren wordt een hypothese gekozen en wordt gezocht naar feiten en voorwaarden die die hypothese ondersteunen; als informatie ontbreekt om een conclusie te trekken dan wordt het verkrijgen van die informatie als subdoel gesteld. Dit recursieve proces gaat door totdat de hypothese wordt aanvaard of verworpen; als tijdens het redeneren wordt aangetoond dat de hypothese onjuist is, dan wordt een andere hypothese gekozen.

Voorwaarts redeneren ('forward chaining') geschiedt vanuit de gegevens. In het geval dat de kennisbank uit regels bestaat, wordt met behulp van het redeneergedeelte gezocht naar regels die voorwaarden bevatten waarvan de waarde bekend is, zodat een conclusie kan worden getrokken. Het trekken van een conclusie zal meestal betekenen dat nieuwe feiten bekend worden. De nieuwe informatie wordt gebruikt om andere regels toe te passen. Indien de voorwaartse redenering vastloopt doordat met behulp van de bekende gegevens geen oplossing kan worden gevonden, kan altijd nog worden overgeschakeld op achterwaarts redeneren om op die manier alsnog een oplossing te vinden.

Als de gebruiker in het begin al over veel informatie beschikt en het expertsysteem zo is ontworpen dat de gebruiker spontaan (zonder dat het systeem er om vraagt) informatie kan invoeren, dan kan met voorwaarts redeneren het zoekproces efficiënt verlopen. In geval van achterwaarts redeneren zouden die bekende gegevens niet direct worden gebruikt en kan het voorkomen dat er allerlei onnodige vragen aan de gebruiker worden gesteld omdat een onjuiste hypothese als doel wordt gekozen. In het geval dat bij achterwaarts redeneren (toevallig) meteen een juist doel wordt gekozen, wordt een minimaal aantal vragen aan de gebruiker gesteld (er worden geen onnodige zijtakken bewandeld).

Aan de hand van het volgende voorbeeld uit Clement (1985) wordt getoond welke stappen er worden genomen bij achterwaarts en bij voorwaarts redeneren:

Een kennisbank bevat de zes regels:

1 als A & B dan F

- 2 als C & D dan G
- 3 als B & G dan J
- 4 als F & H dan X
- 5 als J & K dan X
- 6 als G & E dan K

Eerst redeneren we achterwaarts:

We willen aantonen dat X waar is (we nemen als hypothese aan dat X waar is), en er is bijvoorbeeld vooraf al bekend dat B, C, D en E waar zijn. We zien in de kennisbank dat X waar is als aan de voorwaarden van regel 4 of regel 5 is voldaan.

Laten we beginnen met regel 4. In die regel zien we dat moet worden nagegaan of F & H waar zijn, dat wordt nu het subdoel.

De juistheid van F valt af te leiden uit de juistheid van A en B (regel 1).

Over A is niets bekend¹, daarom staken we deze poging en proberen het via regel 5.

In regel 5 blijkt dat X waar is als J en K waar zijn, dus de nieuwe subdoelen zijn: aantonen dat J en K waar zijn.

Volgens regel 3 is J waar als B en G waar zijn.

G is waar als C en D waar zijn.

C & D zijn al bekend

dus is G waar,

B is ook al bekend

dus is J waar.

Uit regel 6 blijkt dat K waar is

(beide subdoelen zijn bereikt) dus is X waar.

Voorwaarts redeneren gaat in dit voorbeeld als volgt:

B, C, D en E zijn vooraf bekend en stel dat we regel 6 als eerste uitproberen. G is niet bekend dus kunnen we K niet concluderen. We gaan nu verder met regel 1, daar is A niet bekend. In regel 2 kunnen we als conclusie trekken dat G waar is, daarna blijkt in regel 3 dat J waar is. F, H en K zijn nog onbekend dus kunnen we nog niets over X zeggen. Als we weer regel 6 proberen kunnen we nu concluderen dat K waar is omdat inmiddels bekend is geworden dat G waar is (uit regel 2). We blijven de regels proberen die we nog niet gebruikt hebben (regel 1, 4 en 5) en in regel 5 kunnen we tenslotte aantonen dat X waar is.

2.3 *Het dialooggedeelte*

Het dialooggedeelte kan worden onderverdeeld in:

- een vraag-antwoordgedeelte;
- een uitlegggedeelte.

De gebruiker moet op zo natuurlijk mogelijke manier met het systeem om kunnen gaan. Als de gebruiker wordt gevraagd om informatie, dan moet hij/zij een antwoord kiezen uit een aantal voorgeschreven mogelijkheden. Net zo als bij andere informatiesystemen zijn voor expertsystemen nog geen natuurlijke taal interfaces operationeel.

Als de gebruiker uitleg wil hebben moet hij/zij een vraag met een vraag kunnen beantwoorden, en vervolgens een begrijpelijk antwoord van het systeem krijgen. Twee type vragen waar expertsystemen in het algemeen antwoord op kunnen geven zijn 'why' en 'how' vragen. De gebruiker kan vragen waarom een bepaalde vraag wordt gesteld en hij/zij kan vragen hoe een conclusie is getrokken. Het is natuurlijk alleen mogelijk een 'why' vraag te stellen als er achterwaarts wordt geredeneerd, want als een expertsysteem werkt met forward chaining dan wordt er niet naar een bepaald doel toe gewerkt. Als een 'how' vraag wordt gesteld dan is het antwoord beperkt tot het tonen welke kennis is gebruikt. Expertsystemen hebben meestal geen kennis over de gebruikte kennis (metakennis) en kunnen dus niet uitleggen waarom bepaalde kennis is gebruikt, er wordt alleen aangegeven welk pad is gevolgd om tot een conclusie te komen.

2.4 Het werkblok en de agenda

Een expertsysteem heeft een *werkblok* nodig dat wordt gebruikt om door de gebruiker verstrekte gegevens en tussenresultaten te bewaren. De *agenda* dient om allerlei gegevens over het redeneerproces bij te houden. Bijv. in een kennisbank die bestaat uit regels: welke regels zijn toegepast en van welke regels is aan het voorwaarden-gedeelte voldaan maar zijn de conclusies nog niet getrokken. In geavanceerde expertsystemen kan de gebruiker vragen stellen over de gegevens die in het werkblok en de agenda worden bijgehouden.

2.5 Overige mogelijkheden en shells

Een expertsysteem bestaat ten minste uit één versie van de eerder genoemde componenten; een uitgebreider systeem zal ook beschikken over één of meer van de volgende eigenschappen/mogelijkheden:

- 1 Een redeneergedeelte waarin we kunnen kiezen tussen voorwaarts of achterwaarts redeneren.
- 2 De keuze uit meerdere structuren voor kennisrepresentatie.
- 3 De mogelijkheid van 'what if' vragen.
Net als bij spreadsheets moet het bij expertsystemen mogelijk zijn om gevoeligheden te analyseren. Een spreadsheet-model willen we opnieuw kunnen doorrekenen door één van de inputvariabelen een andere waarde te geven; bij een expertsysteem moet het mogelijk zijn het antwoord op een vraag te veranderen en daarna het redeneerproces te laten herstarten.
- 4 Op verzoek ook uitleg waarom een bepaalde conclusie *niet* is getrokken.
- 5 Een dialoog op kunnen slaan om na aanpassing van de kennisbank de dialoog te kunnen herhalen, zonder dat de gebruiker alle vragen opnieuw moet beantwoorden.
- 6 Faciliteiten voor communicatie. In De Swaan Arons&Lucas (1984) worden bijvoorbeeld besproken:
 - grafische uitvoer
 - raadplegen van databanken
 - externe procedures voor onder andere rekenwerk

De eerste expertsystemen zijn voornamelijk in de programmeertaal Lisp geschreven. Tegenwoordig zijn er 'Shells' beschikbaar. Een shell bevat alles wat nodig is om een werkend expertstelsel te krijgen: een vooraf gespecificeerd redeneergedeelte, een vastliggende structuur om kennis in op te slaan en een uitlegfaciliteit. Alleen de kennisbank moet nog worden gevuld door de gebruiker van de shell. Shells maken het mogelijk om een expertstelsel te bouwen (in vrij korte tijd), zonder dat er hoeft te worden geprogrammeerd. Deze prettige eigenschap heeft natuurlijk een keerzijde: het redeneergedeelte is niet flexibel; de manier waarop het redeneergedeelte werkt zou wel eens niet de meest geschikte manier van werken kunnen zijn voor het op te lossen probleem. Dit kan weer opgevangen worden door een veelzijdige en dure shell te kopen of verschillende shells voor verschillende typen problemen te gebruiken.

Achterwaarts redeneren is makkelijker te implementeren dan voorwaarts redeneren; dat is volgens Van Lith (1985) de reden dat het in goedkopere shells meestal niet mogelijk is om voorwaarts te redeneren.

3 Het ontwerp van expertsystemen

Het heeft jaren geduurd voordat het onderzoek op het gebied van de kunstmatige intelligentie praktisch succes opleverde omdat men zich in eerste instantie concentreerde op redeneerstrategieën en werkte aan 'general problem solvers'. Een goed redeneergedeelte is wel noodzakelijk maar niet voldoende. Nadat men tot het inzicht kwam dat de hoeveelheid en kwaliteit van de kennis belangrijker was dan de redenering werd vervolgens de nadruk gelegd op kennisrepresentatie (hoe de kennis kan worden opgeslagen in de computer) en momenteel richt het onderzoek zich steeds meer op het oplossen van de problemen die zich voordoen bij de eerste fasen van het ontwikkelen van een expertstelsel: de kennisverwerving, zie onder andere Wielinga & Breuker (1985).

In de (eerste fasen van de) systeemontwikkeling is de beperkte tijd van de expert een knelpunt. Het is natuurlijk alleen economisch verantwoord om een stelsel te bouwen dat een expert vervangt die veel geraadpleegd wordt, wat betekent dat de expert ook nauwelijks tijd heeft om de ontwerper uit te leggen welke kennis hij/zij gebruikt. Als we in plaats van één expert meerdere experts hun kennis willen laten spuien stuiten we op andere bezwaren:

- soms is het onmogelijk omdat er binnen een organisatie maar één expert aanwezig is;
- het is moeilijk om de vuistregels die verschillende experts gebruiken tot een consistent geheel te maken.

Het bouwen van expertsystemen wordt gezien als een vak apart. Dat heeft tot gevolg dat er weinig wordt geleerd van andere gebieden. Het is bijvoorbeeld de vraag of methoden voor het ontwikkelen van informatiesystemen niet geschikt zouden zijn voor het ontwikkelen van expertsystemen. In tegenstelling tot de aanpak bij het ontwikkelen van conventionele infor-

matisystemen wordt de taak om een expertsysteem te ontwikkelen niet verdeeld over meerdere personen. De 'knowledge engineer' verricht alle taken die anders achtereenvolgens door informatieanalysten, systeemanalysten, -ontwerpers en programmeurs worden verricht (zie onder andere Canters, 1985). Dit kan natuurlijk alleen als bij voorbaat al vast staat dat een te ontwerpen systeem als expertsysteem zal moeten worden geïmplementeerd. Maar wie bepaalt dan dat het nodig is om een te automatiseren systeem als expertsysteem te implementeren?

Er wordt wel heel veel over expertsystemen geschreven maar er is weinig bekend van expertsystemen waarvan de ontwikkeling succesvol is verlopen en die ook daadwerkelijk worden gebruikt. Artikelen over mislukte expertsysteemprojecten zijn nog schaarser. Een aantal voor de hand liggende redenen van mislukking zijn:

- De expert kan of wil geen tijd vrijmaken.
Een expert kan op korte termijn veel geld verdienen of kosten besparen voor een organisatie door adviezen te geven. Meewerken aan de ontwikkeling van een expertsysteem levert pas op langere termijn eventueel succes op.
- De expert is niet bereid om zijn/haar status op te geven, door jarenlang opgebouwde kennis zo maar vrij te geven (Clement, 1985).
- Gebrek aan kennis om kennis te implementeren.

Er is een goede samenwerking nodig tussen de expert en de knowledge engineer; die samenwerking kan pas ontstaan als ze beiden 'iets' van elkaars vakgebied begrijpen. De knowledge engineer moet de juiste vragen kunnen stellen om de kennis uit de expert te krijgen, en de expert moet enigszins aanvoelen wat de beperkingen zijn om kennis in de computer te representeren. Verder moet een knowledge engineer antwoord kunnen geven op vragen als:

Welke kennisrepresentatie is geschikt voor het probleem?

Welke methode van redeneren is geschikt voor het probleem?

Hoeveel kennis moet er in het systeem gestopt worden om een aanvaardbaar niveau van probleemoplossen te bereiken?

Omdat het ontwerpen van een expertsysteem een specialisme is, leent deze taak zich bij uitstek om op zijn beurt ondersteund te worden door een expertsysteem, twee voorbeelden hiervan zijn: TEIRESIAS (Davis & Lenat, 1982) en ROGET (Bennett, 1985).

Het testen van een expertsysteem levert andere problemen op dan het testen van een klassiek programma. Als een expertsysteem niet de gewenste oplossing geeft dan kan met behulp van de uitlegfaciliteit op eenvoudige wijze het gedeelte van de kennisbank gevonden worden dat verantwoordelijk is voor de ongewenste oplossing. Indien de kennisbank bestaat uit regels dan kan de kennis die daar wordt gebruikt worden verbeterd of verfijnd, zonder dat dat gevolgen heeft voor de overige kennis in de kennisbank. Deze prettige eigenschap is helaas niet van toepassing op semantische netten en frames. Maar aantonen dat een kennisbank geen foutieve kennis

bevat is zeer moeilijk want het is niet mogelijk om een testset te genereren zodat alle kennis in de kennisbank getest wordt omdat het voor degene die test niet duidelijk is volgens welk pad een conclusie zal worden getrokken. Bij een procedureel algoritme is het makkelijker om alle takken van het programma op een eventuele onvolledigheid of tegenstrijdigheid te onderzoeken, maar het is moeilijker om zo een fout te herstellen, omdat een verandering ergens in een programma meestal gevolgen heeft op andere plaatsen in het programma.

Er zijn maar weinig artikelen waarvan de titel wijst op een pessimistische kijk op expertsystemen. In de literatuurlijst zijn dat alleen Bell (1985) en Martins (1984). Martins merkt op dat het succes van de eerste expertsystemen onder meer veroorzaakt is doordat:

- 1 ze ontwikkeld zijn door briljante programmeurs;
- 2 ze gemakkelijke problemen oplossen;
- 3 er genoeg tijd en geld beschikbaar werd gesteld;
- 4 er geen commerciële expertsystem tools of shells werden gebruikt.

Geen van deze argumenten lijkt ons een reden om geheel af te zien van het ontwikkelen van expertsystemen. Ten eerste zijn het eerste en het laatste argument complementair: als er een shell wordt gebruikt hoeft er namelijk niet meer geprogrammeerd te worden. En ten tweede is men ook tegenwoordig bereid om hoge kosten te maken voor expertsystemen omdat de besparingen nog groter kunnen zijn.

Als één van de grote voordelen van expertsystemen ten opzichte van traditionele systemen wordt vaak genoemd dat de kennis en het redeneren (de controle over de kennis) niet door elkaar heen lopen zoals in procedurele programmeertalen. Ook spreadsheet-modellen lijden daaronder. In een spreadsheet staan namelijk de gegevens en de relaties tussen de gegevens dwars door elkaar heen. Door die scheiding bij expertsystemen kan niet alleen prototyping worden gebruikt bij het ontwikkelen maar ook het onderhoud wordt hierdoor makkelijker. De kennis die in het prototype is opgenomen kan op eenvoudige wijze worden uitgebreid en verfijnd, zonder dat het redeneergedeelte moet worden aangepast. In procedurele programmeertalen is de volgorde van gegevensverwerking belangrijk; als we een opdracht weglaten of op een verkeerde plaats neerzetten zal het programma niet doen wat de bedoeling is. In een kennisbank is de volgorde van de kennis onbelangrijk, en als er kennis wordt weggelaten zal het systeem toch nog vaak met goede antwoorden kunnen komen. Door deze eigenschap kan al in een vroeg stadium een werkend systeem getoond worden. Het gevaar bestaat dat op deze manier niet wordt toegepast wat we geleerd hebben bij het ontwikkelen van andere informatiesystemen, namelijk dat het beter is om het gebied volledig in kaart te brengen voordat we een implementatievorm kiezen.

Het blijkt echter dat er in de praktijk toch structuur wordt aangebracht in de kennisbank, zodat die niet in geheel willekeurige volgorde wordt door-

lopen. Dit gebeurt dan om de gebruiker niet het gevoel te geven dat het systeem op een 'onlogische' manier redeneert en van de hak op de tak springt. Het blijkt bij grote problemen (grote zoekruimte) ook nodig te zijn om het aantal regels te beperken dat voor evaluatie in aanmerking komt. Een derde reden om structuur aan te brengen is dat het natuurlijk efficiënt is om er voor te zorgen dat een hypothese die vaak juist blijkt te zijn als eerste wordt onderzocht. Bijvoorbeeld bij een systeem voor de bepaling van de indeling in tariefsgroepen voor de loonbelasting is het verstandig om eerst na te gaan of de voorwaarden voor indeling in groep 1 van toepassing zijn want de overige drie groepen zijn uitzonderingsgevallen.

In Fikes&Kehler (1985) wordt er op gewezen dat frames erg geschikt zijn om te worden gebruikt voor het controleren van het redeneren over kennis in de vorm van produktieregels.

4 Gebruik van expertsystemen

Een expertstelsel is een geschikt hulpmiddel bij het oplossen van problemen waarvan de oplossingsmethode moeilijk of helemaal niet als een sequentieel proces is te beschrijven, of waarbij niet alleen feiten en berekeningen een rol spelen maar ook vaagheden, onzekerheden of kwalitatieve factoren. Dergelijke problemen zijn namelijk niet geschikt om op een traditionele manier te worden aangepakt. In Lucas & Van der Gaag (1986) worden twee niveaus van onzekerheid onderscheiden:

- de onzekerheid over een conclusie in een regel
de expert kan in een regel aangeven dat een bepaalde conclusie niet 100 % zeker is maar bijvoorbeeld: 'waarschijnlijk' of '90 % zeker'.
- onzekerheid van de gebruiker
een vraag van het systeem kan door een gebruiker beantwoord worden met bijv.: 'waarde onbekend' of 'voor 20 % zeker'.

De meeste expertsystemen lossen typen problemen op waarvoor van tevoren alle mogelijke oplossingen bekend zijn en waarbij de oplossing van een concreet probleem door middel van een dialoog gezocht wordt. Een expertstelsel voor medische diagnose bijvoorbeeld bevat expliciet alle mogelijke ziekten waardoor dat systeem een diagnose kan stellen. Dat alle mogelijke oplossingen bekend zijn wil nog niet zeggen dat het probleem eenvoudig oplosbaar is. Het oplossen van het probleem blijft complex door de grote hoeveelheid symptomen en relaties tussen symptomen en ziekten. Het doel van expertsystemen is om in situaties waar menselijke experts duur en schaars zijn toch in minimale tijd een oplossing voor een probleem te vinden die van aanvaardbare kwaliteit is. Het is niet de bedoeling dat het systeem mensen helemaal vervangt; meestal handelt een persoon op grond van het advies van het systeem. De eerste generatie expertsystemen is niet in staat om te leren, dus als daar (nieuwe) kennis in gestopt moet worden moet er een menselijke expert bij komen. Een belangrijke eigenschap van de tweede generatie expertsystemen zal zijn dat ze wel kunnen leren (Steels, 1985).

Overzichten van expertsystemen zijn onder andere te vinden in Clement (1985), Frenkel (1985) en Rault (1984); er wordt wel bij opgemerkt dat veel van die systemen nog in een experimenteel stadium zijn. De toepassingen die het meest worden genoemd zijn: 'medische diagnose' en het 'ontwerpen van geïntegreerde circuits'.

Blanning (1984) noemt een aantal management-taken die in aanmerking komen om te worden ondersteund door een expertstelsel. We zullen hier enkele voorbeelden van geven:

Een expertstelsel dat personen verdeelt over taken kan rekening houden met ervaring van de verschillende personen, of bepaalde personen wel of niet kunnen samenwerken, enz.. Aspecten waar geen rekening mee wordt gehouden als alleen aantallen worden toegewezen met behulp van een kwantitatief model/stelsel.

Een tweede voorbeeld is het beoordelen van balansen. Voor deze taak is niet een (eenvoudig) algoritme te geven, maar er wordt wel veel gebruik gemaakt van allerlei regels, voorschriften en kengetallen, die geschikt lijken om in een kennisbank te worden gestopt.

In EDP-analyzer (december 1985) is uitgebreid het expertstelsel 'The Financial Advisor' beschreven. Dit stelsel is behulpzaam bij het nemen van investeringsbeslissingen en als de gebruiker belangrijke punten over het hoofd ziet wordt hij/zij daar door het expertstelsel op gewezen.

Als een expertstelsel wordt ontworpen om taken van managers te ondersteunen dan zijn er principieel twee verschillende mogelijkheden om het te gaan gebruiken:

- 1 De manager kan het zelf gaan gebruiken en daardoor minder afhankelijk worden van zijn/haar stafmedewerkers; bovendien kunnen medewerkers het expertstelsel dan gebruiken om de redenering achter bepaalde beslissingen te begrijpen.
- 2 Medewerkers nemen beslissingen met behulp van het expertstelsel en de manager hoeft de beslissingen alleen nog maar te controleren, om eventueel nieuwe kennis toe te voegen aan de kennisbank.

In beide gevallen kan het expertstelsel worden gebruikt als communicatiemiddel binnen het bedrijf (zie Maes&Van Dijk, 1986); ook Basden (1983) wijst op deze toepassing van expertsystemen.

4.1 Voordelen ten opzichte van conventionele systemen en mensen

Een belangrijk onderscheid met traditionele systemen is de *uitlegfaciliteit*: niet alleen kan uitgelegd worden welke antwoorden er kunnen worden gegeven op een vraag, maar ook kan worden uitgelegd waarom een bepaalde vraag wordt gesteld (wat is het doel). Verder kan worden getoond welke weg is gevolgd om tot een bepaalde conclusie/oplossing te komen. In Clement (1985) wordt opgemerkt dat de uitlegfaciliteit belangrijk is om managers c.q. gebruikers te kunnen overtuigen van de juistheid en betrouwbaarheid van een conclusie. Helemaal in dezelfde lijn wordt in Bouwman (1983) gesteld dat een klacht over beslissingsondersteunende systemen in tegen-

stelling tot expertsystemen juist is dat die systemen een uitkomst leveren zonder de achterliggende argumentatie te tonen op een manier die voor de gebruiker begrijpelijk is.

Kenmerkend voor een expertstelsysteem is verder dat een redeneerproces niet zal stoppen als er een vraag aan de gebruiker wordt gesteld en er geen antwoord of een *onvolledig antwoord* wordt gegeven (Ford, 1985). Andere informatiesystemen verwachten als invoer duidelijke antwoorden en kunnen niet meer met de gewenste uitvoer komen als één van de invoergegevens ontbreekt of als de invoer niet in vooraf vastgestelde volgorde wordt gegeven.

Een expertstelsysteem zal, net zoals een conventioneel systeem, *consequente* uitvoer geven. Indien we tweemaal dezelfde invoer geven krijgen we ook tweemaal dezelfde conclusie/oplossing. Een menselijke expert is in het algemeen niet zo consequent in het oplossen van problemen. De gekozen oplossing kan bij mensen in sterke mate afhangen van vergeetachtigheid, humeur en motivatie van de deskundige. (Zie Feenstra&Van de Poel (1985) voor meer menselijke eigenschappen die de mogelijkheid van het probleem oplossen beperken.) Experts die hebben meegewerkt aan het bouwen van een expertstelsysteem merken, als ze het systeem testen of gebruiken vaak verrast op dat het systeem komt met goede adviezen die de expert op dat moment over het hoofd ziet.

Als het gelukt is om de kennis van een expert in een expertstelsysteem te krijgen kan de expertise makkelijk worden *verspreid*. Zo kan bijvoorbeeld een oliemaatschappij er voor zorgen dat op elk booreiland expertise (in de vorm van een expertstelsysteem) aanwezig is. Zonder zo een expertstelsysteem zouden er bij de schaarse menselijke experts lange wachttijden ontstaan, of er zouden (te)veel dure experts in dienst moeten worden genomen. Een extra moeilijkheid is het tijdsverschil. Als ergens ter wereld problemen ontstaan kan het zijn dat een expert moeilijk of niet bereikbaar is doordat het niet gebeurt tijdens de normale werktijden op het hoofdkantoor. In het geval dat met expertsystemen wordt gewerkt zijn alleen (tijdelijk) dure knowledge engineers nodig. In paragraaf 3 hebben wij er al op gewezen dat er ook expertsystemen bestaan om de taak van een knowledge engineer te ondersteunen.

4.2 Nadelen en beperkingen bij het gebruik

Een van de grote moeilijkheden die zullen ontstaan als voor steeds meer toepassingen binnen bedrijven expertsystemen worden ontwikkeld is de noodzakelijke integratie met andere (bestaande) informatiesystemen. Als een expertstelsysteem informatie nodig heeft die in een gegevensbank gevonden kan worden dan is het natuurlijk inefficiënt als de gebruiker die informatie (opnieuw) zou moeten invoeren. De tot nu toe ontwikkelde succesvolle expertsystemen zijn vaak onafhankelijke toepassingen, (zie bijv. Swaan Arons&Lucas, 1984), die zelfs vaak draaien op aparte machines.

De kennis van een expert neemt langzaam af in de richting van de grenzen van zijn vakgebied; bij een expertsysteem is die grens veel scherper: de kennis is of aanwezig of totaal niet aanwezig. Een expertsysteem kent zijn eigen beperkingen niet en is zich niet 'bewust' van de semantische waarde van de kennis in de kennisbank. Expertsystemen zijn (nog) niet in staat om te leren van ervaringen (Steels, 1985), en problemen waar (te)veel gezond verstand aan te pas komt zijn niet geschikt om met behulp van een expertsysteem te worden opgelost (Kastner&Hong 1984).

Het gebruik van een expertsysteem door personen die niets van het betreffende onderwerp afweten is haast onmogelijk. Het systeem zou te uitgebreid worden als elk begrip uitgelegd zou moeten kunnen worden, en dus intern zou moeten zijn omschreven. Bovendien heeft een leek vaak een andere redeneertrant dan een expert, dus ook al weet een leek iets van het probleem dan nog lost hij/zij het op een andere manier op dan het expertsysteem (zie Riesbeck, 1984 en Basden, 1983). Zo kan een tussenpersoon bijvoorbeeld een expertsysteem gebruiken om voor een klant de meest geschikte verzekering te vinden, maar een klant zelf met het systeem laten werken zou te veel onduidelijkheden opleveren.

Demonstratiemodellen van expertsystemen/kennisbanken zijn dan ook modellen van zeer alledaagse problemen waar iedereen 'verstand' van heeft, dat wil zeggen de gebruikte begrippen hebben voor iedereen min of meer dezelfde betekenis. Enkele voorbeelden zijn een systeem voor diagnose van kinderziekten, het bepalen van de tariefsgroepindeling van de loonbelasting of de oplossing van startproblemen van een auto. De kennis voor dergelijke demonstratiemodellen is zo algemeen dat die verkregen kan worden uit een boek, belastinggids of gebruiksaanwijzing, zonder een expert te raadplegen c.q. te belasten. De ontwikkeling van zo een systeem is dus niet bedoeld om de problemen te demonstreren die zich voordoen bij het vergaren van kennis voor het bouwen van een 'serieus' expertsysteem.

Expertsystemen zijn voorts niet geschikt om problemen die wat betreft ingebouwde logica te eenvoudig of te complex zijn, op te lossen. In Basden (1983) worden als grenzen genoemd een kennisbank met minder dan 10 of meer dan 10.000 regels. Een ander criterium dat vaak wordt genoemd is de tijd die nodig is om het probleem op te lossen. Als vuistregel wordt dan gesteld dat een probleem waarvoor een menselijke expert minder dan een paar minuten of meer dan een paar uur nodig heeft om het op te lossen, niet geschikt is om met behulp van een expertsysteem te worden aangepakt.

5 Vergelijking met beslissingstabellen

Tot nu toe zijn diverse aspecten van expertsystemen aan de orde gekomen. Als laatste willen wij (regels in) expertsystemen op een aantal punten vergelijken met de wellicht meer vertrouwde beslissingstabellen:

- In beslissingstabellen bestaat een sterke scheiding tussen condities en acties. Dat wil zeggen: eerst wordt er naar de voorwaarden gekeken,

daaruit volgt één regel die van toepassing is. Als die regel gevonden is wordt de actie in die regel uitgevoerd. In een kennisbank kan elke regel meerdere keren worden geëvalueerd.

- Tussen de regels van een beslissingstabel bestaat een exclusieve 'of' relatie: regel 1 is van toepassing 'of' regel 2 is van toepassing. In een regel van een beslissingstabel is alleen een 'en' operator toegestaan, bijv.:

regel 1: 'als' A = waar 'en' B = waar 'dan' doe actie 5 'of'

regel 2: 'als' A = niet waar 'en' B = waar 'dan' doe actie 3

Regels in een kennisbank mogen ook andere operatoren dan de 'en' operator bevatten. Zie bijv.: paragraaf 2.1.

- Beslissingssituaties waarin veel condities een rol spelen en waarin er weinig samenhang is tussen de condities kunnen beter niet met behulp van beslissingstabellen worden weergegeven. Dat zou namelijk een groot aantal tabellen opleveren. Als een beslissingssituatie in meerdere tabellen is weergegeven bestaan er expliciete relaties tussen de (regels van) verschillende tabellen, bijv. als actie: ga 'verder met (regel y in) tabel x', dus in tegenstelling tot expertsystemen is bij beslissingstabellen nooit de kennis gescheiden van de redenering over de kennis.
- In een beslissingstabel kunnen in het actie-gedeelte geen onzekerheden voorkomen; een actie wordt zeker uitgevoerd of zeker niet uitgevoerd.
- Een beslissingstabel kan goed getest worden op consistentie en volledigheid, bij kennisbanken is dit niet mogelijk, zoals reeds opgemerkt in paragraaf 3.
- Er bestaan verschillende algoritmen om de regels van een beslissingstabel samen te trekken tot een compact of optimaal aantal. De optimaliteit kan daarbij op verschillende manieren worden gedefinieerd, bijv.: het aantal te testen voorwaarden of het aantal regels. Voor zover ons bekend zijn er nog geen methoden om het gebruik van een kennisbank zo efficiënt mogelijk te laten verlopen.

6 Samenvatting en conclusie

Expertsystemen zullen het mogelijk maken om beslissingsprocessen te ondersteunen op punten waar traditionele (beslissingsondersteunende) systemen te kort schieten. Dankzij traditionele beslissingsondersteunende systemen is het mogelijk geworden om tijdens een beslissingsproces een antwoord te krijgen op 'what if' vragen. Met behulp van een expertstelsysteem kunnen ook 'how' vragen worden beantwoord, waardoor de resultaten van een expertstelsysteem gecontroleerd en begrepen kunnen worden. Dat maakt het voor de gebruiker makkelijker om te beslissen of een uitkomst acceptabel is of niet. Als een gebruiker niet geïnteresseerd is om te weten hoe een uitkomst tot stand is gekomen dan is deze faciliteit toch nuttig tijdens het testen van een expertstelsysteem.

Ondanks het feit dat er slechts enkele succesvolle operationele expertsystemen zijn, zijn de verwachtingen hoog gespannen. Er wordt nu veel geld besteed aan kunstmatige intelligentie en expert-systemen. Grote bedrij-

ven richten er speciale afdelingen voor op en overheden en internationale instellingen zoals de EG financieren grootschalige onderzoekprogramma's. Verder zijn er bedrijven die, na een flinke investering, geld beginnen te verdienen aan kunstmatige intelligentie en expertsystemen, door er over te adviseren of produkten te verkopen die 'knowledge engineering' ondersteunen. Dergelijke produkten zijn echter vaak nog beperkt en ze staan nog in hun kinderschoenen.

Ontwikkeling van expertsystemen is (nog) gecompliceerder dan ontwikkeling van andere informatiesystemen; daarom moeten alleen expertsystemen worden ontwikkeld als andere oplossingen niet uitvoerbaar zijn. Er zijn zeker toepassingen waarvoor expertsystemen nodig zijn, maar de indruk bestaat dat veel organisaties alleen belangstelling tonen voor expertsystemen omdat ze bang (gemaakt) zijn dat ze de boot zullen missen als ze er nu niet mee beginnen.

Literatuur

- 1 Andriole S.J., The promise of artificial intelligence, *Journal of Systems Management*, Volume 36, No. 7, July 1985, pp. 8-18.
- 2 Basden A., On the application of expert systems, *International Journal of Man-Machine Studies*, Volume 19, 1983, pp. 461-477
- 3 Bell M.Z., Why expert systems fail, *Journal of the Operational Research Society*, Volume 36, number 7, July 1985, pp. 613-621.
- 4 Bennett J.S., ROGET: A Knowledge-based System for Acquiring the Conceptual Structure of a Diagnostic Expert System, *Journal of Automated Reasoning*, Volume 1, No 1, 1985, pp. 49-75.
- 5 Blanning R.W., Management Applications of Expert Systems, *Information & Management*, 7, 1984, pp. 311-316.
- 6 Bouwman M.J., Human Diagnostic Reasoning by Computer: An Illustration from financial analyses, *Management Science*, Volume 29, No. 6, June 1983, pp. 653-672.
- 7 By de R.A., Semantische netwerken 1, *Informatie*, jaargang 27, nr. 9 september 1985, pp. 782-794.
- 8 Canters J.P.W., De kunst van knowledge engineering, *Supplement Computable*, 10 mei 1985, pp. 36-40.
- 9 Clement I., Expert Systemen, *Beleidsinformatica Tijdschrift*, Volume 11, nummer 1, 1985, pp. 1-61.
- 10 Davis R. & D.B. Lenat, *Knowledge-based Systems in Artificial Intelligence*, McGraw-Hill, 1982, 490 pp.
- 11 EDP Analyzer, *What is happening with expert systems*, december 1985, vol. 23, no. 12, 16 pp.
- 12 Feenstra D. W. & J. H. R. van de Poel, Bedrijfseconomie, Methodologie en Besluitvorming (deel 3), *Maandblad voor Accountancy en Bedrijfshuishoudkunde*, maart 1985, pp. 65-72.
- 13 Fikes R. & T. Kehler, The Role of Frame-Based Representation in Reasoning, *Communications of the ACM*, Volume 28, No. 9, September 1985, pp. 904-92.
- 14 Ford, F. N., Decision Support Systems and Expert Systems: A Comparison, *Information & Management*, 8, 1985, pp. 21-26.
- 15 Frenkel K. A., Toward automating the Software-Development Cycle, *Communications of the ACM*, Volume 28, No. 6, June 1985, pp. 578-589.
- 16 Hayes-Roth F., D. A. Waterman & D. B. Lenat (eds.), *Building Expert Systems*, Addison-Wesley, 1983, 444 pp.
- 17 Kastner, J. K. & S. J. Hong, A review of Expert Systems, *European Journal of Operational Research*, Volume 18, 1984, pp. 285-292.
- 18 Lith P. van, Start bescheiden met kunstmatige intelligentie, *De Automatisering Gids*, 21 augustus 1985, pp. 7-9.
- 19 Lucas, P. J. F. en L. C. van der Gaag, Prolog en Expertsystemen, *Informatie*, jaargang 28, nr. 1, januari 1986, pp. 16-25.

- 20 Maes R. & J. E. M. van Dijk, A user-friendly propositional formalism for a DSS-generator, in: *Artificial intelligence in economics and management* (ed.: L. F. Pau), North Holland Publishing Co., 1986, pp. 65-76.
- 21 Martins G.R., The overselling of Expert systems, *Datamation*, November 1984, pp. 76-80.
- 22 Rault J., Les systemes experts: perspectives industrielles, *bulletin de liaison de la recherche en informatique et automatique*, no. 97, Novembre 1984, pp. 9-23.
- 23 Riesbeck C.K., Knowledge reorganization and reasoning style, *International Journal of Man-Machine Studies*, Volume 20, Number 1, January 1984, pp. 45-62.
- 24 Steels L., Tweede generatie expert systemen, *Informatie*, jaargang 27, nr. 11, november 1985, pp. 966-971.
- 25 Swaan Arons H., de & P. J. F. Lucas, Expert systemen in een toepassingsgerichte omgeving, *Informatie*, jaargang 26, nr. 8, juli/augustus 1984, pp. 631-637.
- 26 Wielinga B. J. and J. A. Breuker, Interpretation of verbal data for knowledge acquisition, in: *Advances in Artificial Intelligence*, T. O'Shea (ed.), North Holland Publ. Comp., 1985, pp. 41-50.

Noten

I Er zou hierover een vraag kunnen worden gesteld aan de gebruiker maar we veronderstellen in dit geval dat die gebruiker het antwoord ook niet weet.