

Kennissystemen in de ruimtelijke planning

Citation for published version (APA):

Veld, op 't, A. G. G., Bijlsma, E., Starmans, J. G. M., & Timmermans, H. J. P. (1987). Kennissystemen in de ruimtelijke planning. *Planning : Methodiek en Toepassing*, (29), 2-10.

Document status and date:

Gepubliceerd: 01/01/1987

Document Version:

Uitgevers PDF, ook bekend als Version of Record

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Kennissystemen in de ruimtelijke planning

*Dree Op 't Veld°, Emma Bijlsma°, Jan Starmans° en Harry Timmermans**

1. Inleiding

Bij ruimtelijke planning gaat het om de voorbereiding van beslissingen over het ruimtegebruik. De uiteindelijke beslissing is het resultaat van een afwegingsproces waarin de voors en tegens van verschillende mogelijkheden tegen elkaar afgezet worden. Zowel de onderscheiden mogelijkheden als de afweging zijn daarbij zoveel mogelijk het resultaat van een systematisch redeneerproces. Wanneer het gaat om rekenwerk of de bewerking van grote hoeveelheden gegevens kan de computer daarbij behulpzaam zijn. Wat het eigenlijke redeneren betreft liet de computer het tot voor kort afweten. Met het beschikbaar komen van de eerste algemeen bruikbare resultaten van „Artificial Intelligence” in de vorm van expert- of beter kennissystemen lijkt daarin verandering te komen. In dit artikel wordt ingegaan op de belangrijkste kenmerken van deze nieuwe generatie software en een eerste indruk gegeven van toepassingsmogelijkheden in de ruimtelijke planning.

2. „Artificial Intelligence”

De tak van informatica die zich voornamelijk bezighoudt met het nabootsen van het menselijke denkproces in computers is „Artificial Intelligence”. „Artificial Intelligence” is als aparte richting in de 50-er jaren ontstaan als gevolg van initiatieven van een aantal toonaangevende informatica pioniers als Shannon, Minsky en Simon in de Verenigde Staten. Tegenwoordig omvat „Artificial Intelligence” een hele reeks van werkerreinen: o.a. kennisrepresenta-

tie, probleemoplossingsmethoden, kennissystemen, interfaces via natuurlijke spraak, lerende systemen, cognitieve modellering en robotica.

Kennis heeft betrekking op objecten, relaties en procedures binnen een bepaald bereik. Kennisrepresentatie neemt een centrale positie in binnen „Artificial Intelligence”. Binnen dit deelterrein van „Artificial Intelligence” gaat het om de vraag hoe symbolische kennis van de mens en de manier waarop deze daarmee omgaat in een computermodel afgebeeld kunnen worden. De mens is sterk in symboolgericht redeneren en zwak in numerieke operaties, het zuivere rekenwerk. Voor de computer geldt tot nu toe het tegenovergestelde.

Kennissystemen vormen een nieuwe categorie van computerprogramma's die kunnen adviseren, analyseren, categoriseren, consulteren en diagnosticiseren op een niveau dat vergelijkbaar is met dat van een menselijke deskundige (Van Lith, 1986).

Ze lijken derhalve goede mogelijkheden te bieden voor toepassingen in de ruimtelijke planning. Met name de ervaringskennis van een expert of regelgeving op een deelgebied van woningbouw en volkshuisvesting, verkeer en vervoer.

3. Kennissystemen: een eerste indruk

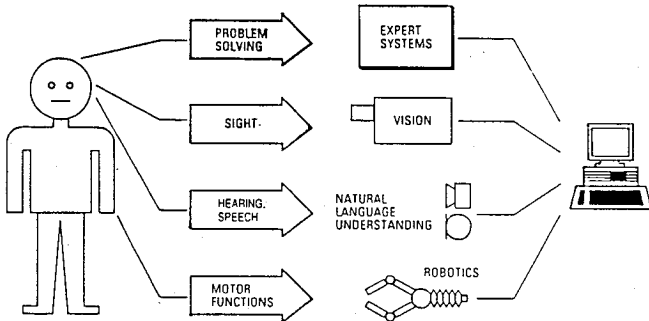
Het aantal definities van kennis- en expertsystemen is groot. Ofschoon ze op onderdelen verschillen wordt de kern wel aangegeven met de definitie van Feigenbaum en McCorduck (1983):

een intelligent computerprogramma dat kennis en inferentieprocedures gebruikt om problemen op te lossen die zo moeilijk zijn dat daarvoor menselijke deskundigheid nodig is.

° Planologisch Studiecentrum TNO

* Technische Universiteit Eindhoven

Figuur 2.1. „Artificial Intelligence” werkkerreinen.*



* Bron: The technical journal of ITT. Nr. 2, 1986

Ze sluit goed aan op de in paragraaf 2 gegeven omschrijving die ontleend is aan Van Lith (1986). Feitelijk is er een onderscheid tussen kennissystemen en expertsystemen. De hiervoor gegeven definitie heeft eigenlijk alleen betrekking op expertsystemen. Kennissystemen is een algemeen begrip. Een systeem kan immers ook betrekking hebben op kennis die geen expertkennis is. Een voorbeeld daarvan is een systeem waarin de juridische regels op een bepaald gebied worden afgebeeld.

In een kennissysteem wordt gebruik gemaakt van op ervaring gebaseerde regels, ook wel heuristische genoemd. De regels die gebruikt worden voor de oplossing van problemen in een bepaald domein zijn georganiseerd in een „knowledge base”. Problemen worden aan het systeem gepresenteerd als een reeks feiten over een bepaalde situatie. Het kennissysteem probeert vervolgens een conclusie te verbinden aan de feiten door de kennis uit de „knowledge base” te benutten.

Heuristieken of heuristische regels zijn de op ervaring gebaseerde beoordelingsregels die gebruikt worden om op grond van feiten een beslissing te nemen. Een van de eerste kennissystemen was een medische diagnosesysteem voor bepaalde soorten ziekten. In de „knowledge base” bevinden zich dan heuristische regels die symptomen verbinden aan een bepaalde ziekte. Wanneer het systeem geconfronteerd wordt met een patiënt en diens ziektebeeld in termen van een aantal symptomen kan de informatie in de „knowledge base” doorzocht worden op de bijpassende ziekte.

Dit zoekproces wordt gestuurd door het zgn. *infe-*

rentiemechanisme (inference engine).

De meeste kennissystemen kunnen gebruikt worden door een vraag- en antwoordspel. De antwoorden worden steeds gebruikt om de feiten vast te stellen op grond waarvan de „knowledge base” doorzocht kan worden.

Townsend en Feucht (1986) noemen een aantal *kenmerken* karakteristiek voor kennissystemen:

- ze beperken zich steeds tot een betrekkelijk klein specifiek kennisdomein. Een medisch diagnosesysteem is niet geschikt voor andere toepassingen,
- de „knowledge base” en het inferentiemechanisme zijn afzonderlijke onderdelen. Het inferentiemechanisme kan vaak goed in combinatie met een ander toepassingsgebied benut worden,
- ze zijn het meest geschikt voor deductieve redeneerprocessen volgens een ALS..., DAN... structuur,
- ze kunnen de redenering die geleid heeft tot een bepaalde conclusie uitleggen in voor de gebruiker direct begrijpelijke termen,
- de verkregen uitspraken zijn kwalitatief,
- een kennissysteem heeft een modulaire structuur, en groeit met de „knowledge base” mee.

Kennissystemen worden inmiddels steeds meer gebruikt voor allerlei *toepassingen*. Van de systemen voor medische diagnose is MYCIN het oudste (midden 70-er jaren) en meest bekende. Digital gebruikt XCON om de configuratie van VAX-systemen af te stemmen op de gebruikerswensen. PROSPECTOR is voor Shell ontwikkeld en wordt gebruikt om via heuristische regels de kans op de aanwezigheid van olie te bepalen.

Deze lijst kan bijna willekeurig uitgebreid worden. Een indruk van de diversiteit van toepassingen geeft het door Schönig (1986) aan Waterman (1981) ontleende overzicht in tabel 3.1.

Kennissystemen bieden goede *mogelijkheden* wanneer:

- de beschikbare data en kennis betrouwbaar en niet aan verandering onderhevig zijn,
- het domein betrekkelijk klein en goed afgebakend is,
- een aanvaardbare probleemoplossing door redeneren bereikt kan worden.

Tabel 3.1. Enkele kennissystemen.*

naam	toepassingsgebied	taal/ontwikkelomgeving
ACE	Bewaken van telefoonnetwerken	PROLOG
ACES	Landkaarten van tekst voorzien	LOOPS
COMPASS	Foutdiagnose in telefoonschakelingen	KEE
CRYSLIS	Analyse van proteïene structuren	LISP
DELTA	Reparatie handleiding voor lokomotieven	LISP, FORTH
DRILLING ADVISOR	Foutdiagnose bij olieboringen	S.1
DRUG INTERACTION CRITIC	Interactie van medicijnen	PROLOG
FALCON	Controle op chemische processen	LISP
FOLIO	Investeringsplanning	MRS
ISA	Planning van orders	OPS 5
ISIS	Productieplanning	SRL
POMME	Controle op appelboomgaarden	PROLOG
RABBIT	Intelligent zoeken in een databank	SMALLTALK, KL-ONE

* Bron: „Waterman: A Guide to Expert Systems”

Voor wat betreft expertsystemen moet daar nog aan toegevoegd worden dat:

- er tenminste één werkelijke expert beschikbaar moet zijn om de kennis aan te ontlenen.

Het gebruik van kennissystemen is aantrekkelijk omdat ze systematisch, ongehinderd door vooroordelen en invloeden van buiten (stemming van de dag) tot controleerbare conclusies komen. Daarmee kunnen ze een aanzienlijke steun voor de menselijke deskundige betekenen.

4. De structuur van een kennissysteem

Het merendeel van de ontwikkelde kennissystemen zijn zogenaamde „rule-based systems” of „production systems”, die ook wel aangeduid worden als „pattern matching inference systems”. Figuur 4.1 geeft een beeld van de componenten van een dergelijk systeem.

Een „production system” bestaat uit drie sleutelementen: de „rulebase”, het gegevensbestand en het inferentiemechanisme. Om de gebruiker een gemakkelijke toegang te garanderen omvat het systeem meestal nog drie andere onderdelen: een module voor kennisvergaring, een in omgangstaal werkende gebruikersingang en een verklaringscomponent.

De *rulebase* bestaat uit een aantal situatie-actie regels van een ALS... , DAN... , vorm. Ze worden „production rules” genoemd.

Bijvoorbeeld:

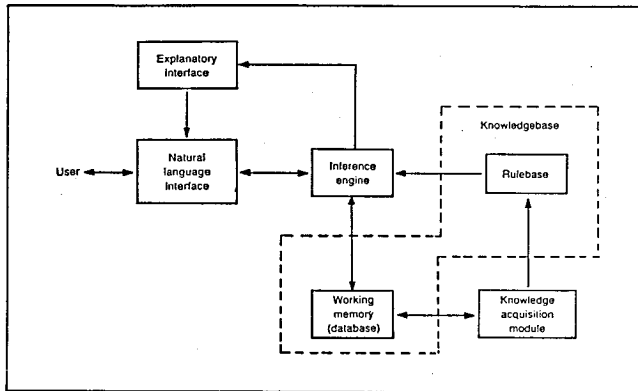
ALS	de motor niet start)	Antecedent of Premisse
EN	de startmotor het niet doet)	
DAN	is er iets mis met het elektrisch systeem)	Consequent of Conclusie

Het eerste deel van een regel, als ALS-deel wordt het antecedent of de premisse genoemd en bestaat uit een of meerdere uitspraken verbonden door EN en OF. Het tweede deel, het DAN-deel, wordt aangeduid als consequent of conclusie en bestaat uit een of meer uitspraken die een conclusie of een actie betreffen. Het antecedent staat voor een „pattern” en de regel wordt aangesproken, „triggered” in het jargon, wanneer de feiten in het werkgeheugen kloppen met het antecedent. Wanneer de actie uitgevoerd wordt, wordt de conclusie aan het werkgeheugen toegevoegd. Dit wordt aangeduid als de regel „has fired”.

De rulebase van een „production system” vertoont daarmee een sterke gelijkenis met het in de psychologie ontwikkelde stimulus-response model.

Het *werkgeheugen* of de gegevensbank (database) omvat een stel feiten die de aangetroffen situatie be-

Figuur 4.1. De componenten van een kennissysteem.*



* Bron: Designing and Programming Personal Expert Systems, Carl Townsend and Dennis Feucht

schrijven en het bevat ook de conclusies die tijdens de consultatieronde door het systeem getrokken zijn. Met het aantal aangesproken regels neemt de kennis van deze tweede soort toe.

In een monotoon of statisch systeem blijven de feiten in het werkgeheugen tijdens de probleemoplossing onveranderd. In een niet-monotoon of dynamisch systeem kunnen de feiten veranderd of zelfs uit het geheugen verwijderd worden tijdens de consultatie. Dat betekent dat binnen het systeem een module voorhanden moet zijn die conclusies die niet met de veranderde feiten stroken snel en goed uit het werkgeheugen verwijdert.

Rulebase en werkgeheugen vormen samen de „knowledge base”.

Het *inferentiemechanisme* bestaat uit twee onderdelen met een eigen taak. De inferentiecomponent onderzoekt de bestaande regels op overeenkomsten met de in het werkgeheugen ingevoerde feiten en voegt eventuele conclusies aan het werkgeheugen toe. De controlecomponent bepaalt de volgorde waarin geselecteerde regels aangesproken worden.

De inferentiecomponent werkt volgens het principe van modus-ponens. Dit houdt in dat wanneer er een regels bestaat ALS A, DAN B; en A is waar (een feit), dan is ook B waar. Wanneer een premisse waar is, is ook de bijbehorende conclusie waar.

De controlecomponent bepaalt de volgorde waarin

de regels doorzocht worden en of wanneer tijdens de consultatie feiten veranderd kunnen worden.

Feitelijk betekent dit dat de gegevens zelf via het „matchen” de route bepalen die op weg naar de oplossing door de rulebase wordt afgelegd. Het lijkt alsof een onzichtbare procedure de weg door de probleemruimte bepaalt. Deze procedures worden wel eens als „demons” aangeduid.

Voor het ontwerp van de controlecomponent van een kennissysteem is de keuze van de zoekprocedure beslissend. De *zoekprocedure* bepaalt de volgorde waarin de rulebase doorzocht wordt en de regels worden aangesproken. Er zijn momenteel 3 alternatieven hiervoor:

1. „backward chaining”
2. „forward chaining”
3. een combinatie van „backward chaining” en „forward chaining”.

Bij backward chaining wordt een conclusie verondersteld, waarna het inferentiemechanisme terug werkt om te zien of feiten voorhanden zijn die de conclusie ondersteunen.

Wanneer van een conclusie op grond van de feiten de geldigheid onderzocht wordt, wordt een sub-doel voor die conclusie gezocht en geprobeerd de geldigheid daarvan aan te tonen. Dit type zoekprocedure wordt vaak aangeduid als „goal-driven” of „consequence-driven” zoeken. Backward chaining is effectief wanneer de gezochte conclusies of doelen bekend en klein in aantal zijn.

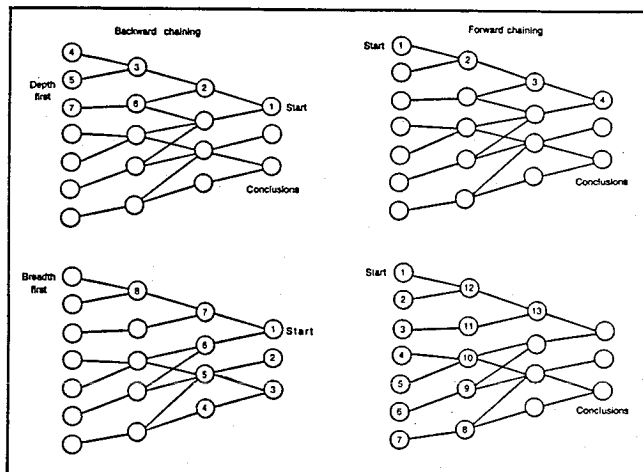
Bij forward chaining wordt begonnen vanuit de feiten en vooruit gewerkt om een conclusie te vinden die door de feiten ondersteund wordt. Dit type procedure wordt aangeduid als „data-driven” of „antecedent-driven” zoeken. Sommige kennissystemen en „tools” werken met een combinatie van backward en forward chaining.

De aard van het probleem bepaalt welke van de drie alternatieven (het meest) geschikt is.

Naarmate de „rulebase” groter is wordt de efficiëntie van het zoekproces belangrijker. Er bestaan verschillende methoden om de efficiëntie te vergroten. Zo kan afhankelijk van het probleem gekozen worden voor „depth-first” of „breadth-first” in een beslisboom (zie figuur 4.2).

Een andere mogelijkheid om de zoekefficiëntie wezenlijk te vergroten biedt het gebruik van „metare-

Figuur 4.2. Zoekprocedures.*



* Bron: Designing and Programming Personal Expert Systems, Carl Townsend and Dennis Feucht

gels". Dit zijn regels die aan de controle-component toegevoegd worden en op grond van ervaringskennis (heuristieken) een aanwijzing geven over de richting waarin de oplossing het snelst gevonden kan worden. Ze grijpen daarmee op een heel ander punt in het proces in dan gewone regels.

In het algemeen vergroot een duidelijke hiërarchische organisatie van de rulebase overeenkomstig de oplossingsstructuur de zoek efficiëntie aanzienlijk. Veel van de heuristische of ervaringsregels waardoor mensen zich bij hun redeneren laten leiden hebben niet het karakter van zekerheden. Een bepaald symptoom wijst op een bepaalde ziekte maar sluit een andere aandoening niet uit. In ruimtelijke planning zijn veel van de verbanden tussen de onafhankelijke en afhankelijke variabelen in een scenario niet met zekerheid bekend. Sommige kennissystemen en „tools” komen hieraan tegemoet door de ALS... , DAN... uitspraken te voorzien van een waarschijnlijkheid. De wijze waarop bij de combinatie van meerdere regels in een redeneerproces met deze waarschijnlijkheden gerekend wordt om tot een samengestelde waarschijnlijkheid te komen verdient dan wel de nodige aandacht. Elk van de daarvoor tot nu toe ontwikkelde en toegepaste procedures heeft zijn eigen aantrekkelijke en onaantrekkelijke kanten. De bekendste methoden om met waarschijn-

lijkheden om te gaan zijn „certainty factor” benadering en de Bayesiaanse statistiek.

5. Vormen van kennisrepresentatie

In een „rulebased system” of „production system” wordt de kennis voor een belangrijk deel vastgelegd in de vorm van ALS... , DAN... regels, maar er zijn ook andere vormen van kennisrepresentatie. De weergave in de vorm van regels en daarmee het „production system” heeft voordelen:

1. de organisatie is modulair, en
2. uniform,
3. de afbeelding van het redeneerproces sluit goed aan op de wijze waarop mensen redeneren,
4. de hiërarchie is flexibel, ze wordt bepaald door de relaties tussen de regels te wijzigen snel veranderd worden.

Daar staan de volgende nadelen tegenover:

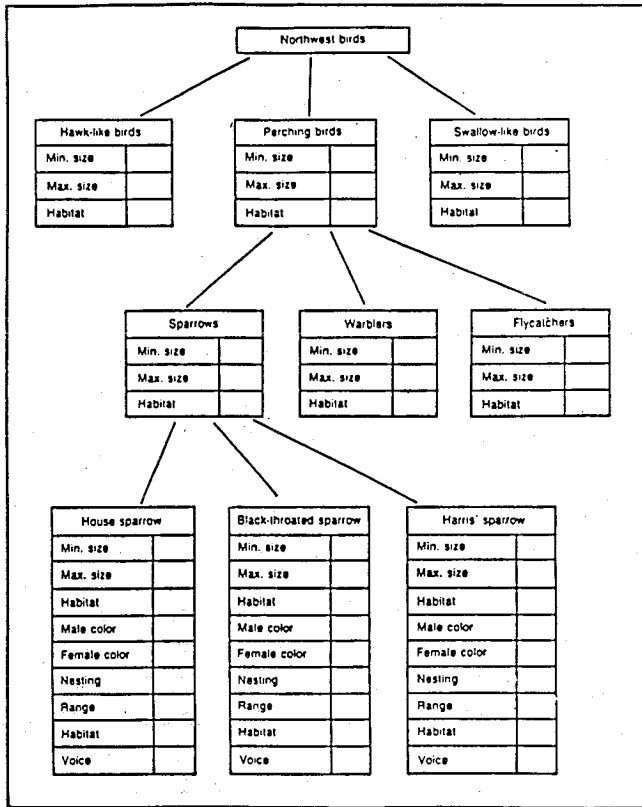
1. ze werken weinig efficiënt. Veel tijd gaat verloren met overbodige zoekcycli,
2. de wijze waarop naar de oplossing gezocht is, is vaak moeilijk te volgen,
3. de hiërarchische structuur is vaak moeilijk te visualiseren.

Een produktiesysteem omvat in de praktijk al snel circa 400 regels. Het wordt dan moeilijk zo niet onmogelijk de „rulebase” en de relaties tussen de regels nog te overzien. Toegevoegde regels kunnen strijdig zijn met al aanwezige regels of deze dupliceren.

„Frame-based” representatie vormt een alternatief voor een „production system”. Frames zijn kaders die naar een aantal vaste rubrieken of „slots” essentiële informatie over een begrip of object samenvatten. Ze lijken daarmee op de „records” in een gangbare databank. Een voorbeeld van een frame-organisatie van onderling samenhangende objecten geeft figuur 5.1.

Elk „slot” of elke rubriek kan een waarde voor het kenmerk, een procedure voor de berekening van die waarde (algoritme) of een of meer produktieregels

Figuur 5.1. Een voorbeeld van een „frame based” kennissysteem.*

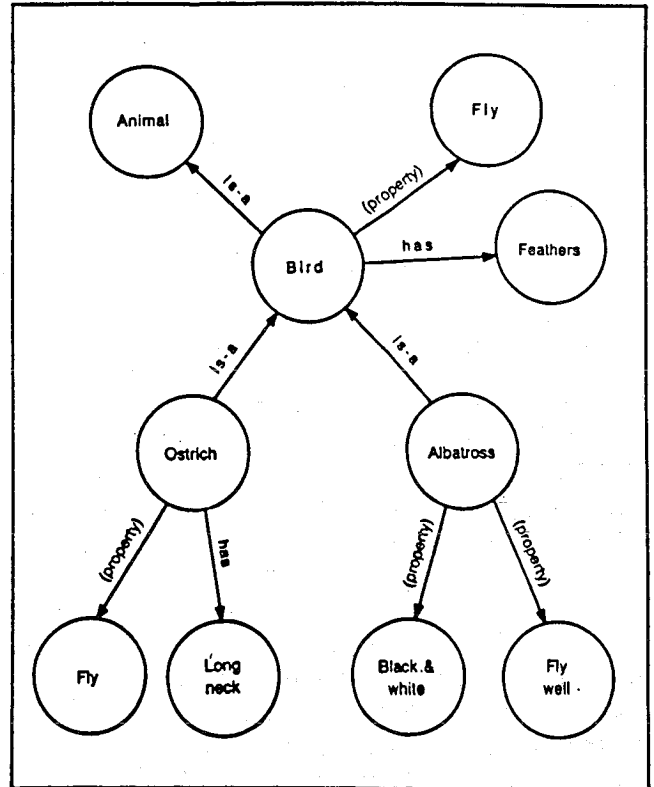


* Bron: Designing and Programming Personal Expert Systems, Carl Townsend and Dennis Feucht

om die waarde te vinden bevatten. Doorgaans zijn de „frames” hiërarchisch georganiseerd. De kenmerken zoals het „frame” voor het „object” boven in de hiërarchie die geeft gelden voor alle lagere orde „objecten”. Elk „object” van lagere orde erft dus de kenmerken van een „object” van hogere orde.

Een derde en misschien de oudste wijze van kennisrepresentatie in AI, naast „rules” en „frames”, zijn de zogenaamde *semantische netwerken*. In een semantisch netwerk wordt de kennis over objecten en begrippen weergegeven door gebruik te maken van kaders die één begrip, object of actie bevatten en onderling verbonden worden door lijnen met een aanduiding van het verband. Een voorbeeld geeft figuur 5.2.

Figuur 5.2. Een voorbeeld van een semantisch netwerk.*



* Bron: Designing and Programming Personal Expert Systems, Carl Townsend and Dennis Feucht

Afhankelijk van het specifieke domein en het doel van het te ontwikkelen kennissysteem verschilt de bruikbaarheid van de geschetste drie vormen van kennisrepresentatie. In de praktijk blijkt een combinatie van „frames” en „rules” het meest gebruikt te worden. Een kennissysteem dat een combinatie van de verschillende vormen van kennisrepresentatie gebruikt wordt een *hybride systeem* genoemd.

Veel auteurs (zie Townsend en Feucht, 1986; Hoppe, 1986) onderscheiden de zogenaamde predicaatlogica als een afzonderlijke vorm van kennisrepresentatie. In predicaatlogica kunnen objecten met kenmerken gedefinieerd worden, kunnen aan objecten en begrippen eigenschappen toegekend worden en kunnen met de logische operatoren AND, OR, NOT en IF ook produktieregels gedefinieerd worden. De predicaatlogica is daarmee eigenlijk van een andere orde als methode voor kennisrepresentatie. De predicaatlogica kan benut worden voor zo-

wel systemen die op regels, op frames of op semantische netwerken gebaseerd zijn als voor hybride systemen.

6. De tools

Programmeertalen als FORTRAN, PASCAL, C zijn sterk in numerieke operaties op basis van een algoritme, een vaste algemene procedure voor de oplossing van een probleem. Ze zijn minder geschikt voor de vormgeving van redeneerprocessen in een computerprogramma. Daarvoor zijn talen nodig die sterk zijn in symboolgerichte bewerkingen waarbij niet uitgegaan wordt van een algoritme maar van een reeks ervaringsregels. „Data-driven control” in plaats van „Procedure-driven control”. Dit heeft er toe geleid dat speciaal voor AI-toepassingen programmeertalen ontwikkeld zijn die aan deze voorwaarden voldoen. De belangrijkste zijn LISP en PROLOG.

LISP (LISt Processing) dateert al van begin 1960 en is daarmee de op één na oudste hogere programmeertaal. Bij de ontwikkeling is voortgebouwd op IPL (lists als data-type) en FORTRAN. LISP is een algemene taal met een grote flexibiliteit. Deze flexibiliteit heeft er samen met de lange tijd waarin LISP aan nieuwe ontwikkelingen blootstaat voor gezorgd dat er een groot aantal dialecten ontstaan zijn. Een dialect bevat steeds verschillende sterke punten van andere vanaf begin 1960 ontwikkelde programmeertalen.

Een aardig overzicht hiervan geeft Schöning (1986).

De veelheid van dialecten (MacLisp, Franz-Lisp, Interlisp, Scheme, Zetalisp, enz.) heeft inmiddels tot een reactie geleid. Er is geprobeerd met COMMON LISP (Winston en Horn, 1984) een krachtige standaard te formuleren. In hoeverre dat een succes is moet blijken.

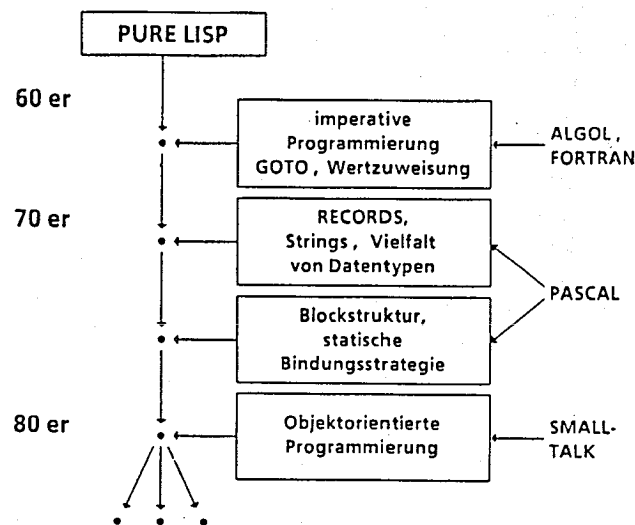
Schöning (1986) noemt een aantal redenen die kennis van LISP voor toepassingen op het gebied van AI en kennissystemen noodzakelijk maken:

- LISP is de meest gebruikte taal in AI, in de standaardwerken wordt doorgaans van LISP gebruik gemaakt;
- De grondregels van LISP zijn eenvoudig en duidelijk en er zijn handboeken voorhanden;

- LISP is gemakkelijk uit te breiden, nieuwe concepten vergen geen andere taal maar kunnen in LISP geformuleerd worden;
- LISP is ruim beschikbaar voor zowel mainframes, micro's als PC's;
- LISP maakt het mogelijk ontwikkelingscycli kort te houden;
- LISP is geschikt voor de ontwikkeling van alle mogelijke combinaties van regel- en objectgeoriënteerde kennissystemen.

PROLOG (PROgrammering in LOGic) is in het begin 70-er jaren ontwikkeld en in tegenstelling tot LISP een specifieke taal. PROLOG maakt gebruik van formuleringen uit de predicaten-logica om een kennisbasis van objecten en regels vast te leggen. Het is vervolgens mogelijk aan die kennisbasis vragen te stellen. Om deze te beantwoorden wordt in PROLOG gebruik gemaakt van een „backward chaining” zoekprocedure. „Forward chaining” alleen of in combinatie met „backward chaining” is in PROLOG niet eenvoudig te verwezenlijken. PROLOG is daarmee beperkt in zijn toepassingsmogelijkheden.

Figuur 6.1. Invloeden van andere programmeertalen op LISP*



Heute: Vielzahl von LISP-Dialekten MACLISP, INTERLISP, SCHEME, COMMON LISP ...

* Bron: Expertise, Gesellschaft für Artificial Intelligence Software und Expertensystems GmbH

lijkheden en mist de flexibiliteit van LISP. Het toevoegen van nieuwe programmeerconcepten is daardoor moeilijk. Daar staat tegenover dat de standaard (Edinburgh-standaard) bij PROLOG niet ter discussie staat. PROLOG wordt minder geplaagd door een veelheid aan dialecten. Voorzover die er zijn gaat het meestal om machinegebonden varianten. PROLOG wordt vooral in Europa – de taal is ontwikkeld in Marseille – veel gebruikt, is goed te leren en is net als LISP ruim beschikbaar op de verschillende soorten computers.

Voorals LISP maar ook PROLOG is in de 70-er jaren gebruikt voor de ontwikkeling van expertsystemen voor verschillende toepassingsgebieden. Het programmeren van een expertstelsel op deze manier kost veel tijd. Dit heeft geleid tot het gebruik van zogenaamde „shells”, lege expertsystemen. Een shell is een expertstelsel zonder domeinkennis. De knowledge base is niet gevuld. Een shell beschikt echter wel over het inferentiemechanisme en bijvoorbeeld de verklaringcomponent van het expertstelsel waaruit de shell ontstaat door het leegmaken van de knowledge base. Een van de eerste shells was EMYCIN, ontleend aan MYCIN, het medisch diagnosestelsel. Later is hiervan weer „Personal Consultant” afgeleid. Door de domeingebonden aard van de expertsystemen waren de eerste shells in hun gebruik beperkt tot problemen van een vergelijkbare structuur. Met name het gekozen inferentiemechanisme werkte vaak beperkend. Gaandeweg worden de shells verbeterd en steeds meer geschikt voor het ontwerpen van allerlei mogelijke kennissystemen. Afhankelijk van zijn ontwikkelingsgeschiedenis heeft elke shell zijn eigen beperkingen en mogelijkheden. Het gebruik van een shell voor de ontwikkeling van een kennissysteem is te verkiezen boven het van de grond af zelf in LISP of PROLOG programmeren van het volledige systeem. Het is dan wel belangrijk een voor het betreffende probleem goed geschikte shell te kiezen. Schöning (1986) geeft hiervoor de volgende algemene criteria:

- de kracht van het inferentiemechanisme;
- de probleemgeschiktheid van het inferentiemechanisme;
- de kwaliteit van de gebruikersinterface voor ontwikkelaar en gebruiker;
- „performance”;

- prijs;
- verkrijgbaarheid en „support”;
- links naar programmeertalen en data-banken.

Biedt voor gebruik op een PC een combinatie van een programmataal en een shell al goede mogelijkheden, op een grotere micro zijn zogenaamde „environments” of *ontwikkelomgevingen* verkrijgbaar die uitstekende faciliteiten geven voor het snel ontwikkelen van verschillende soorten kennissystemen (zie tabel 6.1).

Het toenemend aantal toepassingen van kennissystemen gaat gepaard met het verder aanpassen van basistalen als LISP of het ontwikkelen van nieuwe talen die steeds beter geschikt zijn voor het snel ontwerpen van een kennissysteem. Het onderscheid tussen programmeertalen, ontwikkelomgevingen en shells lijkt daardoor geleidelijk aan te vervagen.

Op het niveau van mainframes is wat de hardware betreft het onderscheid belangrijk tussen „general” en „special purpose” machines. De laatste groep van machines is speciaal ontwikkeld voor toepassingen op het gebied van AI. Het gaat in alle gevallen om zogenaamde LISP-machines.

De basisprincipes van LISP zijn via een LISP-chip in de hardware verwerkt. De combinatie van een ontwikkelomgeving als KEE of ART en een LISP-machine als LMI, Explorer of Symbolics biedt de beste mogelijkheden voor de ontwikkeling van een kennissysteem. De aanschafkosten zijn echter zeer hoog.

7. Toepassing in de ruimtelijke planning

Het aantal toepassingen van kennisysteemtechnieken is langzamerhand niet meer te overzien. Toepassingen in de ruimtelijke planning ontbreken voorlopig nog. In hoofdlijnen dienen zich 3 mogelijkheden aan:

1. de afbeelding van regelgeving,
2. het afbeelden van expertise op een bepaald terrein,
3. het afbeelden van het beslisproces van individuele actoren (huishoudens, bedrijven).

Bij de afbeelding van regelgeving gaat het om het meer toegankelijk maken van een systeem van regels

en voorschriften. Zo zou bijvoorbeeld de puntenwaarderingmethodiek die veel gemeenten hanteren voor de urgentiebepaling bij de toedeling van woningen in een kennissysteem ondergebracht kunnen worden.

In een dialoog met een dergelijk systeem kan door een gemeente-ambtenaar en eventueel de woningzoekende zelf de waardering bepaald worden. Verder is het mogelijk voorgenomen wijzigingen in het waarderingssysteem snel op hun gevolgen voor verschillende categorieën woningzoekenden te onderzoeken. Problemen worden daarmee in een eerder stadium onderkend.

Een andere mogelijke toepassing is het afbeelden van de regels die gehanteerd worden bij de toekenning van de beschikbare middelen voor stadsvernieuwing.

Beslissingen omtrent de vestiging of uitbreiding van winkels en winkelcentra vereisen een speciale deskundigheid. Normaliter vormt een deskundige van een gemeente, het winkelbedrijf of een adviesbureau zich aan de hand van beschikbare feiten en eventueel model-analyses een beeld van de alternatieven, vergelijkt deze met elkaar en kiest. De in dit proces gehanteerde ervaringsregels zouden in een expertsysteem ondergebracht kunnen worden. De deskundige zelf kan dit vervolgens in de voorkomende gevallen benutten om sneller tot een goed onderbouwd advies te komen.

Veel beslissingen in de ruimtelijke planning vragen inzicht in het te verwachten gedrag van individuele actoren. Voor de planning van woning- en andere voorzieningen is het nodig zicht te hebben op de te verwachten migratiebewegingen.

Om hierover uitspraken te kunnen doen wordt in de praktijk gebruik gemaakt van eenvoudige trendexploitatie of regressie- of simulatiemodellen die de statistische samenhang tussen migratie en een of meerdere migratie-bepalende factoren weergeven. Deze modellen hebben hun beperkingen. Het eigenlijke oorzakelijke verband ligt immers op het niveau van het individuele huishouden. Kennissysteemtechnieken kunnen benut worden om het individuele beslisproces rondom migratie op de computer weer te geven.

Uit onderzoek van het Planologisch Studiecentrum TNO blijkt dat het individuele beslisproces rond verhuizen en woningkeuze goed met een zogenaamd „decision plan net” kan worden weergegeven (Op 't Veld, Bijlsma en Starmans, 1986). Om de kennissysteemmethodieken succesvol op dit terrein te kunnen toepassen zal een bevredigende oplossing gezocht moeten worden voor de stap van een reeks individuele beslisprocessen naar uitkomsten op het niveau van migratiestromen.

Kennissysteemtechnieken bieden nieuwe mogelijkheden. Het is in principe mogelijk geworden menselijke redeneringen in hun basisvorm in een computermodel af te beelden. Toepassingen in de ruimtelijke planning zijn er nog niet, ze zijn wel denkbaar. De praktijk zal uitwijzen of deze exponent van „Artificiële Intelligentie” op ons vakgebied net zo'n vlucht te zien zal geven als elders al het geval is.

Literatuur

- Feigenbaum, E.A. and P. McCorduck (1983), *The Fifth Generation: Artificial Intelligence and Japan's Computer Challenge*. Reading, Mass.: Addison-Wesley.
- Hayes-Roth, F., D.A. Waterman and D.B. Lenat (eds) (1983), *Building Expert Systems*. Reading Mass.: Addison Wesley.
- Hoppe, H.U. (1986), *Einführung in die Künstliche Intelligenz unter Anwendungsgesichtspunkten*. Hamburg: On-line Seminar.
- Van Lith, P. (1985), *Experiences with Knowledge Engineering*. Mol: SCK-CEN, Seminar on Expert Systems and Available Hard- and Software.
- Schöning, W. (1986), *Expert System Shells auf dem PC: PC-Tools zur Erstellung von Expertensysteme*. Hamburg: On-line Seminar.
- Townsend, C. and Feucht, D. (1980), *Designing and Programming Personal Expert Systems*. Blue Ridge Summit: Tab Books.
- Op 't Veld, E., E. Bijlsma en J. Starmans (1986), *Individuele gedragsmodellen, 'n verkenning van kennissystemen en decision plan nets*. Delft: Planologisch Studiecentrum TNO.
- Whan Park, C., R.W. Hughes, V. Thukral and R. Friedmann (1981), *Consumer Decision Plans and Subsequent Choice Behaviour*. *Journal of Marketing*, 45, pp. 33-47.
- Winston, P. and B. Horn (1984), *LISP*. Reading Mass.: Addison Wesley.