PLANNING BIJ BRIDGE

EEN TAAKANALYSE VAN HET MAKEN VAN EEN SPEELPLAN IN EEN BRIDGEKONTRAKT

Huub van der Houden

PLANNING BIJ BRIDGE

Een taakanalyse van het maken van een speelplan in een bridgekontrakt

H.G. van der Wouden

Afstudeer onderzoek bij de
vakgroep psychonomie,
fakulteit sociale wetenschappen,
Rijksuniversiteit Utrecht

Begeleiders: dr.H. Kunst
dr.E. Bol

INHOUD

Voorwoord	3
Hoofdstuk 1: bridge in de literatuur	4
1.1 bridge en computers	4
1.2 denksporten en psychologisch onderzoek	7
Hoofdstuk 2: kennissystemen	14
2.1 wat zijn kennissystemen	14
2.2 de construktie van kennissystemen	15
Hoofdstuk 3: een kennis beschrijvings model	18
3.1 doelstellingen	18
3.2 het 4 lagen model van expertise	19
Hoofdstuk 4: het gebruik van het model	25
Hoofdstuk 5: experts aan het werk	31
5.1 taakomschrijving	31
5.2 een gedetailleerd protokol	33
Hoofdstuk 6: herkenning nader beschouwd	61
Hoofdstuk 7: resultaten	71
Hoofdstuk 8: toetsing van het model	89
Hoofdstuk 9: conclusies en generalisaties	98
9.1 generalisaties	98
9.2 het denken van bridgers	103
9.3 een kennis-systeem voor bridge-planning?	105
9.4 evaluatie van het gebruikte model	105
Literatuur	108

VOORWOORD

Tot op heden is er op het gebied van computerbridge nog maar weinig onderzoek gedaan. Een belangrijke reden hiervoor is de grote complexiteit van de taak. Onderzoek op gebied van schaken en Go geeft aan, dat een vruchtbare benadering voor het construeren van programma's die zo'n complexe taak moeten uitvoeren allereerst inhoudt, dat een grondige analyse gemaakt moet worden van het probleem oplossend gedrag van mensen met betrekking tot die taak. Dit onderzoek probeert een eerste aanzet te geven tot een taak-analyse van het maken van een speelplan bij bridge, een eerste vereiste voor sukses in deze denksport. Bij de analyse heb ik gebruik gemaakt van het 4 lagen model van expertise, zoals dat in recent werk van Breuker (1987) uiteen is gezet.

In de hoofdstukken 1-4 zal ik een overzicht geven van zaken, die bij deze analyse van belang zullen zijn: onderzoek op het gebied van computerbridge, de psychologie van denksporters, en kennis (expert)-systemen, en een beschrijving van het 4 lagen-model. In de hoofdstukken 5-8 volgt dan de eigenlijke taak-analyse.

Met veel dank aan mijn ouders, vrienden, vriendinnen, studie- en vakgenoten, proefpersonen, begeleiders en alle anderen die door middel van geestelijke, emotionele, en financiele bijdragen aan de afronding van dit onderzoek hebben bijgedragen. In het bijzonder wil ik nog Heleen Rippen bedanken, die me altijd, na dagen staren naar de terminal, weer tot het normale leven wist terug te brengen.

Utrecht, augustus 1987

HOOFDSTUK 1: BRIDGE IN DE LITERATUUR

1.1 bridge en computers

Hoewel denksporten alle jaren een centraal aandachtspunt van het AI-onderzoek zijn, is er over bridge in de literatuur nog maar weinig verschenen. In tegenstelling tot schaken en de laatste jaren ook Go, is er met betrekking tot bridge geen gecoordineerd onderzoek, er lopen geen grote projekten van universiteiten of bedrijven, en er is geen sprake van enige gedachtenuitwisseling middels b.v. tijdschriften. Het gepubliceerde onderzoek is verspreid over vele jaren en heeft een zeer gevarieerd karakter.

Bieden

Het meest bestudeerde aspekt van bridge is het bieden, omdat het bieden als taak enkele aantrekkelijke eigenschappen heeft. Ten eerste is het een moeilijke, niet triviale taak, waarbij tal van onzekere faktoren een rol spelen. Ten tweede is de kennis die nodig is om de taak op te lossen al in bijzonder gestruktureerde vorm aanwezig in het biedsysteem, zodat snel geimplementeerd kan worden.

Het bieden laat zich als volgt beschrijven. De speler en zijn partner krijgen allebei 13 kaarten; elk ziet slechts zijn eigen 13 kaarten. De 26 kaarten samen hebben een bepaald slagpotentieel: een schatting van het aantal slagen wat er met die kaarten in de later volgende speelfase gehaald kan worden. De spelers moeten samen dit slagpotentieel bepalen, maar er zijn twee problemen:

- Iedere speler ziet slechts 13 kaarten; om tot een goede beslissing te komen moet aan de partner informatie gegeven worden over de eigen hand. Dit informatie-uitwisselings-proces is de essentie van de biedtaak.
- De hoeveelheid informatie die uitgewisseld wordt verhoogt het in de speelfase te behalen aantal slagen. Efficientie is dus van het grootste belang.

Om de taak goed op te lossen moeten de spelers de volgende stappen zetten. Ten eerste moet de eigen kaart 'gewaardeerd' worden: de speler moet schatten in welke mate zijn kaart bijdraagt aan het later te halen aantal slagen. Vervolgens moet hij 'een bod doen': partner informatie overbrengen over zijn kaart. Welk bod hij moet doen staat beschreven in zijn biedsysteem. Dan doet partner een bod; de speler moet dit bod 'interpreteren': hij moet op basis van dit bod zichzelf een beeld creeren van partners kaart. Hij moet dan een schatting maken van het waarschijnlijk te halen aantal slagen, gegeven de informatie op dat moment; dit bepaald of hij nog een bod mag doen. Mag dit, dan moet hij zijn kaart 'herwaarderen', nagaan in welke mate de door partner overgebrachte informatie zijn oorspronkelijke waardering aantast. Daarna moet hij een bod kiezen, dat nog niet eerder vertelde informatie overbrengt. Hierna mag partner weer bieden, en dit gaat

door tot een van beide past. Het laatste bod bepaalt het later te halen aantal slagen, en de speelsoort.

Er zijn verschillende programma's geschreven die deze taak tamelijk bevredigend hebben opgelost (zie v.d.Wouden 1987 voor een overzicht). De programma's hadden de volgende eigenschappen:

- De kennis van het programma bevindt zich voornamelijk in het zogenaamde 'biedsysteem'. In het biedsysteem staat beschreven, wat de relatie is tussen een bod, en de betekenis die het bod gegeven een bepaalde context heeft. Dit laatste heeft de vorm van een beschrijving van de kaart van diegene, die het bod doet, in termen van distributie, honneurs, speelkracht. enz.
- Een bodgeneratie module zorgt voor het genereren van het eigen bod. Hiertoe wordt eerst de hand beschreven met behulp van de in het biedsysteem gebruikte termen; vervolgens wordt in het biedsysteem gezocht naar het juiste bod, met 'context' en de beschrijving van de hand als cues.
- Een bodinterpretatie module doet het omgekeerde als partner biedt; het zoekt bij het bod de passende beschrijving. Soms worden ook negatieve konklusies getrokken uit de biedingen die partner niet heeft gedaan.
- Een kaart-herwaarderings module ten slotte zorgt ervoor, dat de beschrijving van de eigen hand wordt aangepast in het licht van de zelf en door partner gedane biedingen.

Zoals bijvoorbeeld Lindelöf (1983) liet zien, is bovengenoemde opzet in kombinatie met een gevoelig kaartwaarderingsmechanisme en een goed gedefinieerd biedsysteem, voldoende voor een zeer goed presterend biedprogramma.

Spelen

Het andere aspekt van bridge, het spelen, is veel minder onderzocht. Waarom dit zo is, is niet zo moeilijk te raden: het is afgrijselijk ingewikkeld. Het 'spelen' is ook niet een taak op zich; het valt uiteen in allerlei deelvaardigheden. Een goede speler moet een speelplan kunnen maken; hij moet dit plan kunnen uitvoeren, en het plan zonodig aanpassen; hij moet zijn onzekerheid over de onbekende kaarten kunnen verkleinen, en hij moet de kennis die hij tijdens het spel verwerft kunnen gebruiken bij het maken en veranderen van het speelplan. Over het merendeel van deze deeltaken bestaat nauwelijks theorie, en het maken van een goed en volledig speelprogramma is dan ook een projekt waar nog geen enkele onderzoeker zich aan gewaagd heeft. Wel hebben enkelen zich geworpen op een van de deeltaken. Quinlan (1979) concentreerde zich op het 'kaartlezen': de manier waarop experts conclusies kunnen trekken over de plaats van onbekende honneurs. Napjus (1971) schreef een programma dat kleurcombinaties kon leren aanspelen; patroonherkenning vormde hiervoor de basis. Het maken van een speelplan, in feite het belangrijkste, is een taak die nog

door niemand onderzocht is.

Gerelateerd onderzoek in andere denksporten

Belangrijkere ideeen over hoe deze laatst genoemde taak door een programma opgelost zou kunnen worden, kunnen gehaald worden uit sommig schaak- en Go-onderzoek.

In tegenstelling tot het grootste gedeelte van de computerschaak gemeenschap probeerde David Wilkins (1983) in zijn programma PARADISE wel het menselijk schaakdenken te simuleren. In PARADISE (PAttern Recognition Applied to DIrecting SEarch) wordt geprobeerd om speelplannen te construeren op basis van uitsluitend een statische analyse, dus zonder van enige zoektechniek gebruik te maken. PARADISE is in staat een grote hoeveelheid patronen te herkennen in een spelsituatie, en elk van deze patronen (b.v. een overbelast stuk, een verzwakte koningsvleugel, een achtergebleven pion) suggereert een bepaald plan om van zo'n verzwakking profijt te trekken. Zo'n gesuggereerd plan is echter nog vrij grof van karakter (b.v. 'val overbelast stuk aan'), maar kan verder worden uitgewerkt omdat PARADISE ook kennis heeft over het aanvallen van overbelaste stukken, achtergebleven pionnen, enz. Door zo'n deelplan weer verder uit te werken, wordt het oorspronkelijke plan in zo primitief mogelijke onderdelen ontleed, en het primitiefste plan is uiteraard een zet.

De in de statische-analyse-fase aldus ontstane plannen vormen de basis voor gerichte zoekaktiviteit in de latere zoekfase. In de standaard 'brute-force' programma's worden alle zetten onderzocht; PARADISE onderzoekt echter alleen maar die zetten die door een van de geformuleerde plannen zijn gesuggereerd. Zoeken in de PARADISE zin van het woord is in feite slechts een taktische verifikatie van zaken die eerder al op basis van stellingskenmerken zijn ontdekt. Dit in tegenstelling tot het brute-force zoeken, dat de hoofdaktiviteit van het programma genoemd mag worden.

Reitman en Wilcox hadden op Go gebied enkele jaren eerder al een dergelijke benadering gebruikt bij hun grootschalige Go-programma projekt (zie Reitman en Wilcox 1978 voor een introduktie). Hun belangrijkste uitgangspunt was de simulatie van menselijk Go-denken, en zij concentreerden zich in eerste instantie op het perceptuele aspekt van de taak: de manier waarop de Go-meester het Go-bord waarneemt, welke relaties en strukturen hij ontdekt, en hoe die relaties de in aanmerking komende zetten bepalen. Ook zij gingen er dus van uit, dat zoeken een 'tweede plan' aktiviteit is: het dient slechts om lokale, taktische effekten van een zet na te gaan. De strategische concepten en plannen, die van veel groter belang zijn bij het vinden van goede zetten, moeten worden geformuleerd in een statische, perceptuele fase, die aan het zoeken voorafgaat.

Voor de construktie van hun programma baseerden Reitman en Wilcox zich op protokollen van partijen van Jim Kerwin, destijds een van de sterkste spelers van Amerika. Het enorme bestand van patronen

waarover hij bleek te beschikken werd gerepresenteerd in de vorm van 'lenses': modules die bepaalde configuraties van stenen op het Go-bord konden herkennen, kennis hadden van hogere orde relaties van deze configuraties met andere configuraties, kennis hadden van te ondernemen akties en te realiseren doelen met betrekking tot die configuraties, enz.

Met behulp van deze modules genereerde het programma een complexe beschrijving van de situatie op het bord, op meerdere nivo's. In de op de perceptiefase volgende zetgeneratiefase werd op basis van deze beschrijving een zet gekozen. Het was de bedoeling van Reitman en Wilcox dat ook deze fase het menselijk denken zou reflekteren, maar aan een goede uitwerking hiervan kwamen zij nooit toe. In de eerdere versies van het programma zorgde een strikte hierarchie van doelen voor de zetgeneratie: de steen-configuraties suggereerden akties; de akties werden geordend op prioriteit, en de aktie met de hoogste prioriteit werd gekozen. Deze struktuur had in latere versies vervangen moeten worden door een flexibelere planner, maar helaas moest het onderzoek wegens geldgebrek worden stop gezet.

De relevantie van de in de boven beschreven programma's gekozen benadering wordt ondersteund door de resultaten uit psychologisch onderzoek. Uit zowel onderzoek naar schaak- bridge- en Go-denken blijkt, dat het voornaamste verschil tussen experts en minder goede spelers is gelegen in de manier waarop zij <u>waarnemen</u>, en niet in geheugencapaciteit of zoekvaardigheid.

1.2 Denksporten en psychologisch onderzoek

De Groot onderzocht in de jaren 30 en 40 het denken van <u>schakers</u> in een reeks beroemde experimenten (de Groot 1946). Zijn resultaten laten zich als volgt samenvatten.

In een eerste experiment liet hij verscheidene grootmeesters en mindere spelers een schaakpositie zien, waarin zij onder hardop denken de beste zet moesten vinden. Het bleek, dat de grootmeesters vaker de beste zet wisten te vinden (niet verbazing-wekkend), maar dat zij hierbij niet dieper zochten (ca. 7 ply) dan de mindere spelers! Dat verhalen over het gigantische geheugen van grootmeesters wat overdreven waren bleek ook uit het feit dat zij slechts een gemiddelde van 35 zetten per partij bekeken. Op een of andere manier is het zoeken van de grootmeester dus efficienter: zij 'zien' meer, zodat zij geen tijd en moeite hoeven te verspillen aan 'niet-relevante' zetten.

De rol die deze perceptie speelt in het denken van schakers onderzocht de Groot met behulp van het volgende experiment. De schakers kregen allen gedurende enkele sekonden een onbekende schaakstelling te zien. Enige tijd later moesten zij proberen die stelling uit het geheugen te reconstrueren. De grootmeesters en meesters wisten zich bijna alle stukken te herinneren (93%); goede

spelers ongeveer 72%, en de mindere spelers uit de groep 51%. Werd het experiment echter herhaald met een random stelling - een bord waar at random stukken op geplaatst waren, zonder onderlinge samenhang - dan skoorden alle spelers gelijk: slechts ca. 15%. Blijkbaar was het niet de capaciteit, maar de organisatie van het grootmeestergeheugen die het superieur maakte aan dat van mindere spelers. De grootmeester is door ervaring in staat om de samenhang tussen individuele stukken te ontdekken, en kan bij het herinneren van de stelling zo'n stukkenconfiguratie als één geheel onthouden. De beginner daarentegen is niet bekend met die patronen, en moet de stelling stuk voor stuk onthouden. Zo'n als een geheel tellend brok informatie wordt ook wel 'chunk' genoemd. Omdat het 'korte termijn geheugen' (het geheugensysteem dat bij de herinner-taak gebruikt wordt) een beperkte inhoud heeft van ca. 7 chunks (Miller 1956), is het duidelijk dat hoe groter de chunks zijn, hoe meer informatie er kan worden onthouden.

Herbert Simon en zijn medewerkers (Chase en Simon 1973, Simon en Gilmartin 1973) onderzochten de grootte en de inhoud van grootmeester-chunks vergeleken met die van beginners. Hiertoe lieten zij hun proefpersonen twee taken uitvoeren, een perceptie-taak en een geheugen-taak. In de perceptie-taak moesten de proefpersonen een stelling van het ene schaakbord naar het andere verplaatsen. Terwijl ze dit deden mochten ze zo vaak als zij wilden naar de twee borden kijken; hun kijkgedrag werd op video vastgelegd. Het bleek, dat de tijd die lag tussen twee opvolgende blikken op het eerste bord langer was naarmate de direkt na elkaar op het tweede bord geplaatste stukken minder relaties met elkaar hadden. Stukken met een sterke onderlinge samenhang werden snel achterelkaar geplaatst, terwijl de tijd die lag tussen twee niet gerelateerde stukken veel groter was. Uit de relatie kijktijd-stuksamenhang destilleerden Chase en Simon, dat stukken die binnen 2 sekonden na elkaar geplaatst werden binnen één chunk lagen. Met behulp van deze grens konden ze vervolgens de preciese inhoud van een chunk nagaan.

Iets dergelijks deden zij bij de geheugen-taak, waar de proefpersonen na 5 sekonden kijken uit het geheugen een stelling moesten reproduceren (de Groot's taak). Dit reproduceren had meestal plaats in korte 'uitbarstingen', waarin een reeks stukken achterelkaar genoemd werden. Chase en Simon gebruikten de tijd tussen het noemen van twee stukken om de hele reproduktie op te delen in chunks. Hun resultaten lieten zien, dat hoewel er geen wezenlijk kwalitatief verschil bestond tussen de chunks van spelers van verschillend nivo, de chunks van grootmeesters aanzienlijk groter waren dan die van mindere spelers; bovendien konden de grootmeesters zich meer chunks herinneren. Net als de Groot vonden zij echter, dat er geen verschil tussen de spelers bestond als de taken werden uitgevoerd met random stellingen. Chase en Simon concludeerden hieruit, dat bij een grootmeester de chunks hierarchisch georganiseerd moesten zijn, zodat het onthouden van een chunk gepaard zou gaan met het onthouden van de daaronderliggende chunks.

Gelijksoortige experimenten werden voor Go en bridge uitgevoerd door resp. Reitman (1976) en Charness (1979). Charness liet in de eerste van een serie experimenten zijn proefpersonen, bridgers varierend van beginner tot expert, onder hardop denken enkele problemen oplossen (zie fig.1.1). Bovendien moesten ze, na enige tijd onderbreking, zich de handen uit de problemen proberen te herinneren.

♠ V B 8 5 2
♥ A V 4
♦ A 10 6
♣ 7 5

♠ A H 9 6 3
♥ H 8 2
♦ H B 4
♣ A V

Kontrakt: 6♠; uitkomst ♥B

---

figuur 1.1: een typisch probleem
(uit Charness 1979)

Het bleek, dat experts niet alleen betere oplossingen vonden, maar dat ook nog veel sneller deden. In het voorbeeld van fig.1.1 kost het de expert slechts 6 sekonden om de oplossing te vinden: het is het resultaat van pure herkenning; een bepaalde kaartconfiguratie suggereert al direkt een bepaalde speelwijze. Experts denken beter en effektiever omdat zij meer, en uitgebreidere patronen hebben dan minder goede spelers, die op hun beurt weer meer, en uitgebreidere speelwijzen suggereren.

Ondanks het feit dat de experts het spel veel korter onder ogen hadden dan de minder goede spelers (door hun snellere oplossingen), wisten zij zich meer kaarten te herinneren. Dit kwam mede omdat zij ook veel meer andere zaken over het spel hadden onthouden, b.v. het kontrakt en het probleem-thema, en de gekozen speelwijze. Uit deze laatsten kon vaak zonder veel moeite de distributie van de handen en de plaatsing van de honneurs worden afgeleid.

Hoezeer de efficientie in het coderen van de kaarten (het herkennen van patronen) verantwoordelijk was voor de snelheid en kwaliteit van de oplossing werd nog verder onderzocht in een tweede experiment. Hierin kregen de proefpersonen dia's te zien van 13 kaarten, gesorteerd op kleur en volgorde. Zij moesten zo snel mogelijk een openingsbod verzinnen voor de getoonde hand.

Het bleek dat zowel de snelheid als de nauwkeurigheid waarmee het openingsbod gekozen werd gecorreleerd waren aan de sterkte van de speler. Hoewel allerlei faktoren het beeld enigszins vertroebelen (de taak vereist enkele vaardigheden die niets met herkenning te maken hebben, zoals puntentellen; minder goede spelers hebben een minder nauwkeurige 'openingsbod-bibliotheek'), concludeerde Charness, dat het toch voornamelijk de efficientie van de kaart-representatie lijkt te zijn, die verantwoordelijk is voor het kwaliteitsverschil.

In een derde experiment werd de geheugentaak van de Groot en Chase en Simon voor bridge aangepast. De spelers werd gevraagd een hand van 13 kaarten te onthouden, nadat ze deze gedurende 5 sekonden hadden kunnen bekijken. In het ene geval was dit een op kleur en volgorde gesorteerde hand, in het andere geval waren de kaarten niet gesorteerd (random over de hand verdeeld). Bij de random verdeelde hand zou de chunk-grootte beperkt zijn tot precies 1 kaart, zodat geen verschil verwacht kon worden tussen de prestaties van goede en van slechte spelers. Bij de gesorteerde hand zou de grotere patroonkennis van de expert grotere chunks tot gevolg moeten hebben, met als gevolg een beter onthouden van de hand. Dit was precies wat Charness vond.

Helaas waren Charness' experimenten er slechts op gericht om te verifieren of de bij Go en schaak gevonden fenomenen ook op bridge van toepassing waren. Hij ging verder niet in op de denkprocessen die plaats vonden nadat de patronen speelwijzen gesuggereerd hadden. Voor schaken werd dit wel gedaan door de Groot (1946), Newell en Simon (1972), en Wagner en Scurrah (1971).

De Groot leidde uit de door hem onderzochte protokollen af dat het denkproces van de schaker de volgende 4 fasen doorloopt:

1. de orientatiefase; hierin wordt de speler zich bewust van de mogelijkheden die de stelling herbergt; de relaties tussen de stukken, enz.
2. de verkenningsfase, waarin enkele kandidaat-zetten worden onderzocht op hun oppervlakkige eigenschappen.
3. de onderzoeksfase, waarin de overgebleven zetten uit fase 2 verder worden onderzocht.
4. de bewijsfase, waarin de in fase 3 geselekteerde beste zet nog eens onderzocht wordt, en gekeken wordt of deze zet inderdaad de beste is.

In de orientatiefase al vindt een groot gedeelte van het selektie-proces plaats, waarin beslist wordt welke zetten in aanmerking komen voor later onderzoek. Deze selektie is enorm: van het totaal van zo'n 30 a 40 mogelijke legale zetten werden gemiddeld slechts ca. 4 zetten onderzocht door grootmeesters. De in deze fase gevonden patronen zijn blijkbaar gekoppeld aan plausibele zet-generatoren; deze zorgen ervoor dat slechts zoektijd wordt besteed aan a priori in aanmerking komende zetten. Newell en Simon (1972) goten dit model in de vorm van een produktie-systeem.

De zoekfasen worden gekarakteriseerd door een proces wat de Groot 'progressive deepening' (successieve verdieping) noemde. Spelers kiezen een bepaalde zet uit voor onderzoek en zoeken die tot een zekere diepte door; vervolgens keren ze terug naar de basis van dezelfde zet om hem nog eens te doorzoeken, zich alleen concentrerend op een andere zijtak. De zoekboom gegenereerd door dit proces lijkt volgens Newell en Simon (1972) nog het meest op 'a skinny christmas tree'.

Ook is het mogelijk dat dezelfde zet later in de analyse nog een keer wordt doorzocht. Dit komt omdat het zoeken altijd doelgericht is: de doorzochte zet wordt geevalueerd in het licht van het doel dat op dat moment wordt nagestreefd. Als op een gegeven moment een nieuw kenmerk in de stelling ontdekt wordt, dat tot het formuleren van nieuwe doelen leidt, dan kan het zijn dat een afgewezen zet opnieuw onderzocht moet worden, omdat de evaluatie met betrekking tot het nieuwe doel heel anders kan uitvallen. Met betrekking tot dit zoekproces bestaat er tussen grootmeesters en andere spelers geen verschil.

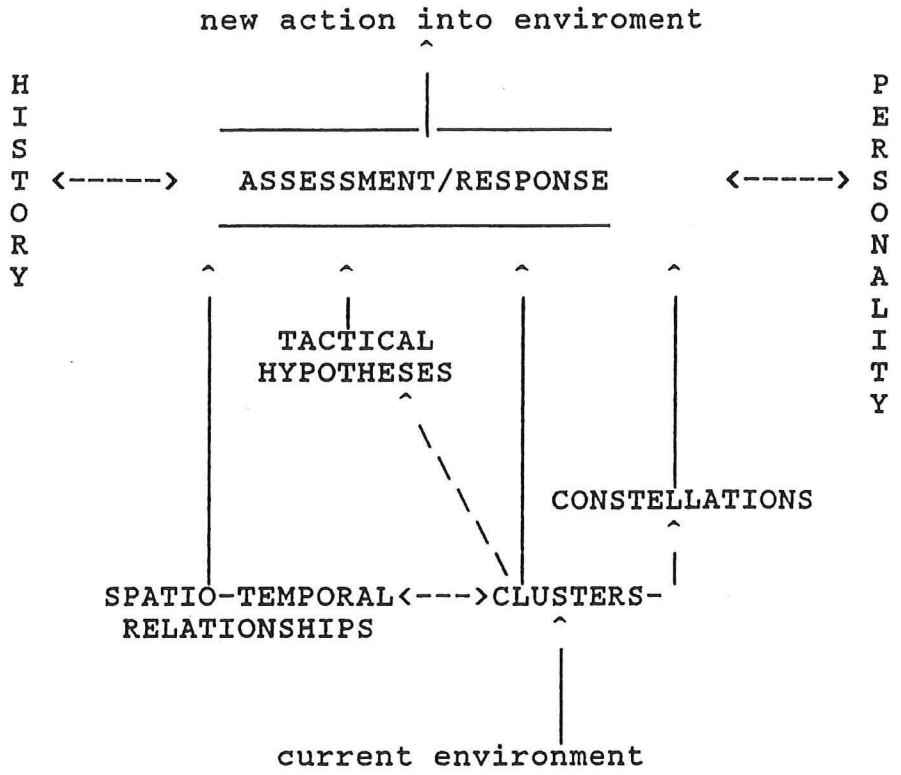
Uitgaande van de hierboven vermelde onderzoeken kunnen we over het denken van denksporters, en bridgers in het bijzonder, de volgende conclusies trekken:

- Het verschil tussen goede en slechte spelers is meer gelegen in perceptuele en geheugen faktoren dan in algemene probleem-oplossende vermogens en logisch denken, nml. in informatie-codering en geheugen-organisatie.
- Experts beschikken over een enorme hoeveelheid patronen, waardoor zij snel problemen kunnen classificeren. Is de categorie waartoe het probleem behoort bekend, dan worden hiermee mogelijke oplossingen geassocieerd. Dit maakt het zoeken door de ruimte van mogelijke oplossingen een efficiente en doelgerichte aktiviteit.
- Bij het schaken wordt de 'orientatie-fase' gevolgd door een reeks zoek-fasen; hoe bij bridge uit de 'gesuggereerde oplossingen' een definitieve oplossing wordt gedestilleerd is niet bekend.

De organisatie van een bridge-programma

Wat voor conclusies kunnen we trekken over de organisatie van een menselijk bridge-denken simulerend programma? Uit het computerbridge-onderzoek zelf weinig. Meer kan afgeleid worden uit de schaak- en Go-programma's op dit gebied. De twee bekendste, de programma's van Wilkins (1980) en Reitman en Wilcox (1978) volgen beide de hierboven beschreven 'denksport-expert' karakterisering. Bruce Wilcox, die met het programma NEMESIS (1985) verder ging waar het Reitman-Wilcox projekt moest ophouden, geeft de

architektuur van zijn programma als volgt schematisch weer (zie fig.1.2):



figuur 1.2: systeem architektuur van situation assessment systems (uit Wilcox 1985)

Deze architektuur is kenmerkend voor 'situation-assessment systems', computersystemen die een complexe situatie moeten kunnen waarnemen, bepalen wat in die omgeving de belangrijkste faktoren zijn die verandering behoeven, en hierop adekwaat kunnen reageren. Voorbeeld is het programma dat de zelfstandig rijdende maanwagentjes moet besturen. Denksportprogramma's zijn volgens Wilcox echter lid van de zelfde familie.

De omgeving wordt eerst geanalyseerd in 'clusters' en 'constellaties', patroon-achtige hogere orde eenheden, die in bepaalde relaties (ruimtelijk en temporeel) met elkaar staan. Deze clusters worden geassocieerd met 'tactical hypotheses', mogelijke akties die in de door de cluster gekenmerkte situatie ondernomen kunnen worden. Er moet echter nog wel voorspeld worden of die akties in de huidige omgeving inderdaad wel het gewenste effekt hebben, vandaar de naam 'taktische hypothese'. De kennis verzameld in de vier categorieen samen vormt de 'perceptie' van de omgeving. Op basis van deze perceptie komt het systeem vervolgens via het

'assessment/response' proces (wat Wilcox verder niet gedetailleerd beschrijft, maar waarin de hypotheses getoetst worden m.b.v. kennis over interaktie van verschillende faktoren, kennis over effekten van bepaalde akties, enz.) tot een passende handeling. Denksporten zijn overigens een uitstekend gebied om dit soort systemen voor te ontwikkelen: de omgeving verandert slechts langzaam en op gezette tijden, en het systeem heeft (bijna) alle tijd om zijn antwoord te produceren. In het dagelijks leven gaat het natuurlijk allemaal veel sneller.

Een ding is in ieder geval duidelijk: het bridgeprogramma in spé zal een flinke hoeveelheid bridge-specifieke kennis moeten hebben. Deze kennis kan, Wilcox' architektuur volgend, natuurlijk vele vormen aannemen: kennis om te omgeving te organiseren, kennis om het effekt van akties te voorspellen, kennis over hoe bepaalde doelen gerealiseerd moeten worden, enz. Het is ook duidelijk, dat het maken van een dergelijk programma geen fluitje van een cent is, en dat de organisatie van zo'n programma knap ingewikkeld zal worden.

Met de huidige ontwikkelingen op het gebied van zog. *'kennissystemen'* (ook wel expertsystemen genoemd) is gelukkig al een stuk meer bekend over het construeren van programma's die met behulp van grote hoeveelheden kennis en een uitgebreid skala van probleem-oplos methoden de meest complexe taken moeten uitvoeren. In het nu volgende hoofdstuk zal ik een korte beschrijving geven wat deze programma's doen, hoe ze eruit zien, en hoe men te werk moet gaan om ze in elkaar te zetten.

HOOFDSTUK 2: KENNISSYSTEMEN

2.1 Wat zijn kennissystemen

Hayes-Roth et al. (1983) definieren kennissystemen als computer-programma's die ontworpen zijn 'to represent and apply factual knowledge of specific areas of expertise to solve problems'. Uitgangspunt bij het maken van deze programma's was het besef, dat het expert zijn op een bepaald gebied te maken heeft met een grote hoeveelheid *domein-specifieke kennis* , en niet met een soort algemene 'probleem-oplos intelligentie'. Oudere AI-programma's gingen juist van het laatste uit, en maakten gebruik van de zog. 'zwakke probleem-oplos methoden' (b.v. Nilsson 1980), zoals generate and test, verschillende zoektechnieken, of means-end analysis. Om programma's óók expert-prestaties te laten leveren, is het dus niet voldoende om het programma algemene kennis mee te geven: op een of andere manier moet de kennis van de expert op een bepaald terrein in het programma belichaamd worden. Domein-specifieke kennis is nodig omdat problemen door de complexiteit en de onvolledigheid van de omgeving vaak niet op algorithmische wijze zijn op te lossen; het is de ervaring van de expert (gegoten in de vorm van *heuristieken*), die hem of haar in staat stelt snel de belangrijke aspekten van het probleem eruit te halen, en een veelbelovende mogelijke oplossing snel te ontdekken.

Aan kennis alleen heb je natuurlijk niets; deze moet immers ook nog toegepast kunnen worden. In de *inference engine* is de methode (of methoden) gespecificeerd volgens welke dit gebeurt. Welke methode gebruikt kan worden is natuurlijk sterk afhankelijk van het op te lossen probleem. Bij eenvoudige problemen (die b.v. gekenmerkt worden door weinig potentiele oplossingen, betrouwbare data uit de omgeving, en betrouwbare, zekere kennis) is een methode als backward-chaining zoeken (zoals gebruikt in het programma MYCIN) voldoende. Aan de andere kant van het spektrum hebben we een programma als HEARSAY-II, een systeem voor spraak-herkenning. Spraakherkenning is een taak die gekenmerkt wordt door veel van de problemen die een ingewikkelde architektuur van het systeem noodzakelijk maken. Enkele van deze problemen zijn: onbetrouwbare en veranderende data; de aanwezigheid van een grote hoeveelheid mogelijke oplossingen; het ontbreken van een vaste manier om het probleem op te delen in sub-problemen; het ontbreken van manieren om te evalueren of een sub-probleem goed is opgelost; het bestaan van veelvuldige interakties tussen sub-oplossingen; afwezigheid van een goed model dat het redeneren stuurt; de noodzaak tot het combineren van verschillende soorten kennis in het zelfde domein; en de noodzaak van het gebruiken van verschillende representaties ter verhoging van de efficientie van het systeem. Om deze problemen aan te kunnen is HEARSAY-II georganiseerd als een verzameling onafhankelijke 'experts' (georganiseerde hoeveeelheden kennis, elk op een deelgebiedje van het probleem), die via een 'blackboard' (een centraal werkgeheugen) met elkaar kunnen communiceren.

Zo'n eenmaal ontwikkelde inference engine is natuurlijk voor meerdere taken geschikt dan alleen spraakherkenning. Omdat de domein-specifieke kennis in een kennissysteem gescheiden blijft van de kennis die nodig is om deze domeinspecifieke kennis toe te passen, is het mogelijk om een eenmaal ontwikkelde inference engine te 'vullen' met verschillende soorten domeinspecifieke kennis. In zo'n geval wordt de inference engine 'tool' of 'shell' genoemd. Een shell wordt dan meestal ook voorzien van middelen die het mogelijk maken de kennis snel en in een gemakkelijke vorm in te brengen, een natuurlijke taal-interface, etc. Of een inference engine bruikbaar is hangt af van het soort taak waar deze oorspronkelijk voor ontwikkeld is. ROSIE, EMYCIN, KAS en nog diverse andere tools hebben allen hun sterke en zwakke kanten, en of zij geschikt zijn voor een bepaalde taak kan alleen beoordeeld worden als de eigenschappen van de taak in kwestie bekend zijn.

Dit is overigens geen onbelangrijk punt. Ondanks de optimistische (bijna euforische) geluiden over de ontwikkelingen in en mogelijkheden van kennissystemen is het bepaald niet zo, dat er voor elk type taak al een inference engine voor handen is. Het snel kopen van een shell, en van daar uit maar eens verder werken, is dan ook nauwelijks een aan te bevelen strategie bij het ontwerpen van een kennis-systeem. Van eerste belang bij de construktie van een kennis-systeem is een grondige analyse van de taak die het systeem moet uitvoeren. In de volgende paragraaf wordt dit nader toegelicht.

2.2 De construktie van kennis-systemen

Voordat er voor een taak een werkend systeem gemaakt is moet er heel wat gebeuren. Allereerst zijn er twee mensen nodig: een expert op het domein in kwestie, wiens kennis overgedragen moet worden naar het programma, en een 'knowledge engineer' (kennis-ingenieur?) wiens taak het is om de kennis van de expert in een zodanige vorm te gieten dat het samen met het gekozen inferentie-mechanisme combineert tot een programma dat doet wat het moet doen.

De eerste stap in het ontwerpproces wordt wel 'kennis-acquisitie' genoemd. Kennis-acquisitie bestaat uit het eliciteren en interpreteren van data betreffende expert-kennis op een bepaald domein. Via gesprekken met de expert, het observeren van de expert als deze prototypische problemen oplost, en het gebruik van diverse interview- en andere technieken moet de knowledge engineer zich een idee vormen van het op te lossen probleem, de soorten kennis die de expert hiertoe gebruikt, en op wat voor manier hij hierbij te werk gaat.

De informatie die de bovenstaande fase heeft opgeleverd moet de knowledge engineer in staat stellen om een passende tool te kiezen. De tool moet geschikt zijn om het oplossingsproces, zoals hij dat bij de expert heeft waargenomen, gestalte te geven. Vervolgens moet hij de aan de expert onttrokken kennis

formaliseren : hij moet de kennis gieten in de vorm van de in de tool gebruikte representatie. Dit kunnen b.v. simpele 'als-dan regels' zijn om heuristische kennis mee vorm te geven, frame-achtige strukturen om concepten te representeren, enz.

Een volgende stap moet een poging tot implementatie zijn. In deze fase wordt ook de controlestruktuur van het systeem belangrijk: hoe en waar de verschillende soorten kennis interakteren bij het construeren van een oplossing voor het probleem. Ook moet een gedeelte van de kennis geimplementeerd worden, nml. zoveel als nodig is om de werking van het systeem goed te kunnen evalueren. Het resultaat is een eenvoudig prototype van het systeem.

Het testen van het prototype sluit de cyclus af. Als het gedrag van het programma echter niet voldoet (en dat is uiteraard in het merendeel van de gevallen het geval), dan zullen in de in eerdere fasen genomen beslissingen aanpassingen moeten worden aangebracht. Dat kunnen eenvoudige aanpassingen zijn, zoals het veranderen van regels omdat deze de kennis niet goed reflekteren, maar ook hele ingrijpende, zoals het veranderen van de controle-struktuur, of zelfs het geheel en al weggooien van de tot dan toe gebruikte tool.

Het bovengeschetste proces wordt door Hayes-Roth et al. (1983) in de volgende vijf stadia samengevat:

Identifikatie:	het bepalen van de eigenschappen van het probleem;
Conceptualisatie:	het vinden van concepten om de aan de expert onttrokken kennis te representeren;
Formalisatie:	het ontwerpen van strukturen om de kennis in te organiseren;
Implementatie:	het formuleren van regels die de kennis gestalte moeten geven; en
Testen:	het nagaan of de in de eerdere stadia genomen beslissingen tot het gewenste gedrag van het systeem geleid hebben.

Bij deze methodologie zijn verschillende kanttekeningen te plaatsen. Allereerst vormt zij niet meer dan een neerslag van de ervaringen die in de korte tijd dat het vakgebied bestaat zijn opgedaan, zoals de schrijvers op verschillende plaatsen aangeven. De bovengenoemde vijf stadia vormen dus meer een groffe karakterschets van het kennisacquisitie proces, dan een strikt te volgen sukses recept.

Ook zijn de aktiviteiten binnen de verschillende stadia vrij slecht gedefinieerd. In slechts weinig gevallen is er sprake van een echt raamwerk, waarbinnen de aktiviteiten van de knowledge engineer gestruktureerd kunnen worden. De schrijvers stellen: "Although work is progressing on automating the expert system

development process (..), at present knowledge engineers must rely on their own skill and insight to guide the knowledge acquisition activity" (blz.23).

Het ontwikkelingsproces van kennissystemen zoals Hayes-Roth et al. dat schetsen lijkt een vrij ongestruktureerde en inefficiente aktiviteit, gekenmerkt door een grote hoeveelheid backtracken (het terugnemen van eerder genomen beslissingen) en het weggooien van prototypes. De invulling van de in de verschillende stadia geformuleerde taken bestaat in veel gevallen uit beschrijvingen van eerder ontwikkelde systemen, en praktische tips die bij het ontwikkelen hiervan zijn opgedaan. Op het moment dat het boek werd samengesteld kon het misschien ook niet beter, maar een alternatief kon niet uitblijven.

Dit alternatief wordt geboden in het werk van Breuker en Wielinga en hun medewerkers (b.v. Breuker en Wielinga 1985). In hun KADS-methodologie (Knowledge Acquisition and Documentation Structuring) wordt de nadruk binnen het kennissysteem ontwikkelings proces verschoven naar de vroege fasen van de kennis acquisitie. Niet eerder mag men zich met implementatie problemen bezig houden, dan na het formuleren van "a detailed conceptualisation of the functions the prospective expert system has to perform in an operational environment, and of the basic architecture of the system" (Breuker en Wielinga 1985, blz.3).

Wat nodig is, is het exakt in kaart brengen van de kennis van de expert, en wel in dusdanige termen dat aan deze beschrijving direkte konklusies verbonden kunnen worden aangaande de inrichting van het systeem in spé. Hoe dit volgens hen gerealiseerd kan worden wordt beschreven in het volgende hoofdstuk.

HOOFDSTUK 3: EEN KENNIS BESCHRIJVINGS MODEL

3.1 Doelstellingen

Op basis van het werk van Clancey (b.v. 1983, 1985) proberen Breuker en Wielinga te komen tot een struktuur, waarbinnen expert-kennis "in its full glory" beschreven kan worden. Het model moet het volgende mogelijk maken:

- binnen het model moet een volledige beschrijving mogelijk zijn van de kennis van de expert, met nadruk op de flexibiliteit van deze;
- het model moet gebruikt kunnen worden als uitgangspunt voor het ontwerpen van flexibelere systemen dan die er nu bestaan; en
- het moet ondersteuning bieden voor de analyse van expert-kennis in de kennisacquisitie-fase van de ontwikkeling van kennissystemen.

Waar het om gaat is het struktureren van data-interpretatie. Tot nog toe ontbrak het de knowledge engineer aan hulpmiddelen hiervoor. Een belangrijk punt is, dat het model een analyse van de gegevens op epistemologisch nivo voorstaat. Dit wil zeggen dat binnen het model slechts gerefereerd wordt aan strukturele eigenschappen van de kennis, en nog niet aan eventueel mogelijke implementatie-mechanismen. Dit punt is in de AI nog al eens genegeerd; meer dan eens is er bij het construeren van programma's uitgegaan van voor handen zijnde formalismen, in plaats van een grondige studie van de gebruikte kennis als uitgangspunt te nemen.

Het resultaat van zo'n data-analyse op epistemologisch nivo is de 'functionele specificatie' van een kennissysteem. Binnen die specificatie wordt beschreven welke soorten kennis de expert bij het oplossen van de taak gebruikt, hoe die kennis georganiseerd is, hoe ze interakteert, en welke verschillende processen de kennis manipuleren. Het zal duidelijk zijn dat het organiseren van de kennis in dergelijke epistemologische categorieen een aanzienlijke betere basis vormt voor het nemen van latere implementatie-beslissingen, dan deze direkt te laten afhangen van verbale gegevens.

3.2 Het 4 lagen model van expertise

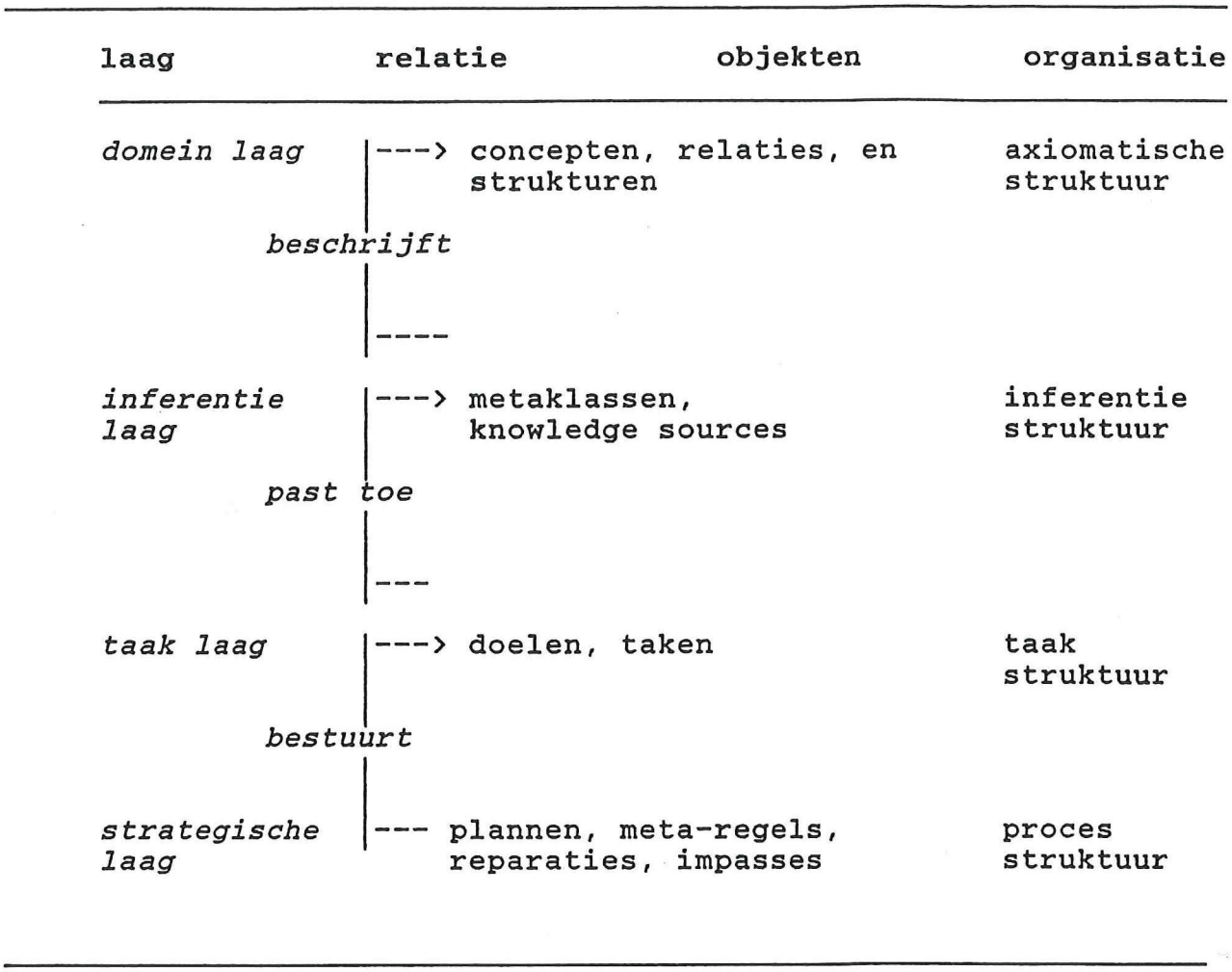
Kennis kan vele vormen aannemen. Op verschillende plaatsen (Davis 1980, Sternberg 1982) wordt beargumenteerd, dat het handig is een onderscheid te maken tussen bijvoorbeeld kennis over domein-elementen, en kennis over hoe er met die elementen geredeneerd moet worden. In het 4 lagen model, dat zijn oorsprong heeft in het onderzoek dat Clancey verrichte voor het systeem MYCIN, wordt dit onderscheid nog wat verder doorgevoerd.

De kennis die een expert gebruikt bij het oplossen van een probleem kan op 4 verschillende nivo's beschreven worden. Op het eerste nivo bevindt zich de *statische domeinkennis*. Hieronder wordt verstaan alle feitelijke kennis over het domein van expertise: domeinconcepten, relaties tussen de concepten, en eventueel complexere strukturen zoals modellen van b.v. processen.

In de tweede laag, de inferentielaag, bevindt zich kennis over de *inferenties* die gemaakt kunnen worden op basis van de kennis op domein-nivo. Deze kennis moet gebruikt worden, en in de inferentie-laag wordt aangegeven hoe nieuwe kennis uit oude verkregen kan worden, welke primitieve bewerkingen hierbij gebruikt worden, en op welke kennis die bewerkingen aangrijpen. Op dit nivo worden 2 soorten eenheden onderscheiden, nml. *meta-klassen* en *'knowledge sources'* . Meta-klassen beschrijven de rollen die domeinconcepten kunnen spelen in een redeneer proces. Een bepaald concept kan bijvoorbeeld fungeren als hypothese, als gegeven, als assumptie, enz. Knowledge sources beschrijven primitieve redeneer processen; deze processen grijpen aan op relaties tussen domeinconcepten. Een voorbeeld van een knowledge source (verder: KS) is 'generalisatie'; deze maakt gebruik van b.v. de 'is-een' relatie. KS-en en metaklassen samen vormen de inferentie-struktuur.

In de derde laag, de *taak-* laag, bevindt zich kennis hoe bepaalde KS-en gebruikt kunnen worden om een doel te realiseren. De basiselementen op dit nivo zijn taken en doelen. In de taak-struktuur wordt aangegeven hoe een taak ontleed kan worden in deeltaken; de primitiefste taak is natuurlijk een KS.

In de vierde laag tenslotte wordt *strategische* kennis beschreven: kennis over hoe plannen gemaakt worden, controle-kennis over de uitvoering van taken, en kennis over wat er moet gebeuren als er iets mis gaat tijdens het oplossen. Het model is schematisch weergegeven in figuur 3.1.



figuur 3.1: Nivo's in de beschrijving van expert-kennis (uit Wielinga en Breuker 1986)

De domein-laag

De kennis in de domein-laag is neutraal ten aanzien van de te verrichten taak, d.w.z. dat er op dit nivo geen kennis aanwezig is die bepaalde probleem-oplos processen sturen of ondersteunen. De kennis in de domein-laag kunnen we beschouwen als competentie-model van de expertise.

De manier waarop de domeinkennis georganiseerd is wordt wel de axiomatische struktuur genoemd. Deze is in principe alleen bepaald door het domein zelf, maar is natuurlijk ook afhankelijk van de bekend zijnde representaties.

De inferentie-laag

Breuker en Wielinga (1986) geven het volgende voorbeeld van

verschillende metaklassen die er bestaan in een medisch domein (zie figuur 3.2):

META-KLASSE	VOORBEELD	ROL
symptoom	klachten koorts bleekheid hoofdpijn	in het aktiveren van hypotheses
manifestatie	overspannenheid koorts	in verifiering van hypotheses
faktor	alcoholgebruik medikatie leeftijd	als ondersteunend bewijs
hypothese	pathologische toestand causaal objekt	in het stellen van een diagnose
klinische diagnose	syndroom	oplossing
causale diagnose	kwaal	

fig.3.2: meta-klassen in een medisch domein (uit Wielinga en Breuker 1986).

Een concept kan tot meer dan één meta-klasse behoren, zoals b.v. koorts, dat zowel een symptoom kan zijn als een manifestatie van een bepaalde ziekte. De metaklassen zelf kunnen bovendien ook georganiseerd zijn in een sub/super-klasse hierarchie, zodat een rol die een bepaalde metaklasse kan spelen ook voor haar subklassen geldt. Deze rol tenslotte ligt vast; het concept 'leeftijd' bijvoorbeeld kan alleen gebruikt worden als ondersteunend bewijs voor een bepaalde hypothese. Bovendien hebben concepten behorende tot de metaklasse 'faktor' ook nog andere eigenschappen: er kan bijvoorbeeld direkt naar gevraagd worden tijdens een consultatie, iets dat niet geldt voor b.v. concepten in de meta-klasse 'hypothese'.

Een KS is een funktionele beschrijving van een primitief inferentie-proces. Een KS werkt op bepaalde meta-klassen, en heeft weer andere meta-klassen als output, en gebruikt bepaalde domein-relaties. Een KS kan derhalve beschouwd worden als een abstrakte operatie die het ene abstrakte concept omzet in een ander.

In een inventarisatie van verschillende KS-en onderscheidt Breuker (1987) KS-en die 1. concepten veranderen; 2. een nieuw concept genereren uit een ander concept; 3. uit de relatie tussen twee concepten een nieuw concept genereren; en 4. strukturen manipuleren. Een (wellicht niet volledige) typologie van KS-en ziet er als volgt uit (zie fig.3.3):

```
knowledge-source
|   change-concept
|   |   assign value
|   |   compute
|   generate concept
|   |   instantiate
|   |   classify (identify)
|   |   generalise
|   |   abstract
|   |   specify (refine)
|   differentiate between concepts
|   |   compare
|   |   match
|   structure manipulation
|   |   assemble
|   |   sort
|   |   decompose
|   |   transform
|   |   |    parse
```

fig.3.3 Typologie van knowledge sources (uit Breuker 1987).

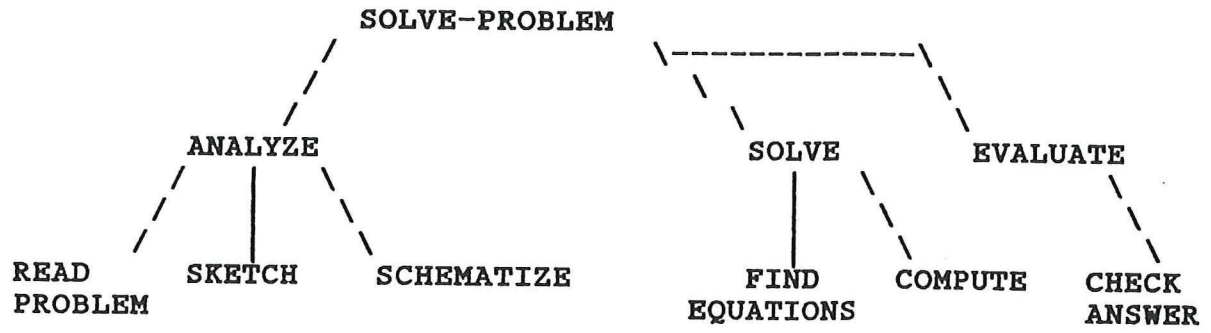
Een andere eigenschap van een KS is de inferentie-methode die een KS gebruikt om uit een domein-relatie nieuwe concepten af te leiden. Een specifikatie-KS kan b.v. een IS-A taxonomie van domein-concepten gebruiken door middel van de 'inheritance'-methode. Breuker geeft geen uitgebreide opsomming van deze methoden, maar voorbeelden kunnen zijn: value propagation, inheritance, resolution, bepaalde match-algoritmes en dergelijke uit de AI-literatuur bekende methoden.

In de inferentie-struktuur worden KS-en en meta-klassen gecombineerd om aan te geven welk type inferentie wel en welk type niet gemaakt kan worden. Belangrijk is dat de inferentie-struktuur een statisch karakter heeft: er wordt niet in aangegeven in welke volgorde de inferenties gemaakt worden. Dit gebeurt namelijk op taak-nivo. De volgende inferentie-struktuur karakteriseert het systeem NEOMYCIN:

data --(abstraction)-> evidence --(association)-> hypothesis --(refinement)-> diagnosis

De taak-laag

Op dit nivo wordt beschreven hoe een taak uiteenvalt in primitievere taken, en in welke volgorde deze laatsten moeten worden uitgevoerd om de 'hogere' taken tot een goed einde te brengen. De taakstruktuur wordt gewoonlijk gerepresenteerd als een 'goal-tree'. Een voorbeeld is gegeven in figuur 3.4:



figuur 3.4: statische taakstruktuur voor het oplossen van natuurkunde-problemen (uit Wielinga en Breuker 1986)

Een dergelijke taakstruktuur heeft een statisch karakter; er staat niet in aangegeven wat er moet gebeuren als het uitvoeren van een subtaak mislukt, of mogelijke alternatieve manieren om een bepaald doel te bereiken. Dit soort zaken worden bepaald op strategisch nivo.

In de taakstruktuur wordt gerefereerd aan doelen en aan 'controle' (de volgorde waarin de doelen moeten worden gerealiseerd). Een doel wordt beschreven als een aktie op een objekt, b.v. 'generate hypothesis'. Uiteindelijk, op het laagste nivo van de goal-tree, moeten de akties KS-en zijn, en de objekten meta-klassen. Een controle-beschrijving kan twee vormen aannemen: if <state> then <goal statement>, en while <state> then <goal statement>.

De strategische laag

Hoewel een taakstruktuur zoals hierboven beschreven goed kan funktioneren bij het oplossen van problemen, is het duidelijk dat hiermee niet de flexibiliteit van de expert wordt verkregen. Experts zijn in staat om hun strategie af te stemmen op het

probleem: voor het ene probleem wordt een andere aanpak gekozen (wordt een andere taakstruktuur gegenereerd) dan voor het andere. Ook moet tijdens het uitvoeren van de taken zoals gespecificeerd in de taakstruktuur in de gaten gehouden worden of het oplosproces wel gaat zoals gewenst, wat er moet gebeuren als het misgaat, of er snellere methoden voor handen zijn in bepaalde gevallen, enz.

Wat nodig is, is een component die in staat is om voor een te bereiken doel een taakstruktuur te genereren. In tegenstelling tot een 'vaste strategie', waarbij dezelfde taakstruktuur gevolgd wordt, onafhankelijk van het karakter van het probleem, wordt in een flexibele strategie eerst het probleem geanalyseerd, en op basis van de resultaten van deze analyse wordt een passende taakstruktuur gekozen. Vooralsnog is dit aspekt van het model van Breuker en Wielinga nog oningevuld.

HOOFDSTUK 4: HET GEBRUIK VAN HET MODEL

Eén van de doelstellingen die genoemd werden in relatie tot het 4 lagen model van expertise, was dat het model ondersteuning moest bieden aan de analyse van expert-kennis in de kennisacquisitie-fase van de ontwikkeling van kennis-systemen. Dit punt behoeft enige toelichting.

Het resultaat van de analyse van de kennis die een expert gebruikt bij het oplossen van een bepaalde taak wordt het conceptueel model van de expertise genoemd. Een conceptueel model vormt een beschrijving van deze kennis op alle vier de nivo's van het 4 lagen-model van expertise, en wordt gebruikt voor het bepalen (is een funktionele specifikatie) van de architektuur van het te construeren kennis-systeem.

Ondanks het feit dat expertise tot uiting komt op vele manieren en in het oplossen van vele verschillende taken, lijkt het toch mogelijk om enige struktuur in die expertise aan te brengen. Het uitgangspunt hierbij is dat voor taken die tot hetzelfde type behoren, hetzelfde soort kennis gebruikt wordt om ze op te lossen. Hoewel per type taak de kennis op domein-nivo natuurlijk verschilt, is het wel mogelijk om overeenkomsten aan te duiden op inferentie- en taak-nivo. Dit houdt in, dat conceptuele modellen van expertise op verschillende domeinen, maar op gelijksoortige taken, dezelfde struktuur kunnen vertonen. Het is, met andere woorden, mogelijk om een conceptueel model van expertise op een bepaalde taak los te maken van het specifieke domein van toepassing. Zo'n abstrakt model - taakgebonden maar domein onafhankelijk - wordt door Breuker (1987) een interpretatie model genoemd.

Een interpretatie model is een beschrijving van de kennis die gebruikt wordt om een bepaalde taak op te lossen, op het inferentie-, taak-, en strategische nivo. Het is het interpretatie model nu, dat kan dienen als hulpmiddel bij de interpretatie van de aan de expert onttrokken verbale data voor het construeren van een conceptueel model. Omdat de taak al (voorlopig) geclassificeerd is, weet de knowledge engineer waar hij op moet letten bij de analyse.

Een interpretatie model heeft betrekking op een klasse taken. Breuker onderscheidt in dit verband 'real life tasks' en 'generic tasks'. Een generic task is een elementair soort taak, een taak die, in de woorden van Breuker, 'can stand on its own' (blz.45). Een generic task is primitief, maar op een vrij abstrakt nivo. Een real life task is samengesteld uit generic tasks. Een real life task is dus niet primitief; een real life task is echter nog steeds domein-onspecifiek.

Het classificeren van taken, en het ontwikkelen van een bibliotheek van generic tasks, is een belangrijk doel bij het struktureren van menselijk probleemoplossend gedrag. Anderzijds

kan het bestaan van zo'n bibliotheek de analyse van expert-kennis aanzienlijk versnellen. Hoe verder uitgewerkt een generic model is (een generic model is een interpretatie-model van een generic task), hoe gemakkelijker het wordt om kennis-systemen te construeren. Op verschillende plaatsen in de kennis-systeem literatuur heeft men geprobeerd om dit classificeren vorm te geven.

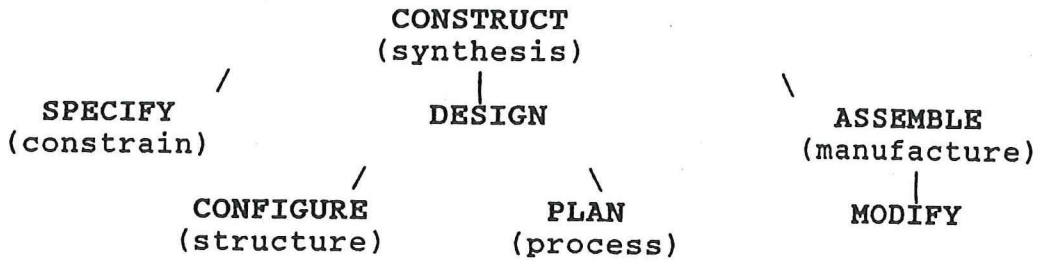
Hayes-Roth et al. (1983) onderscheiden de volgende generic tasks (zie fig.4.1):

INTERPRETATION	Inferring situation descriptions from sensor data
PREDICTION	Inferring likely consequences of given situation
DIAGNOSIS	Inferring system malfunctions from observables
DESIGN	Configuring objects under constraints
PLANNING	Designing actions
MONITORING	Comparing observations to plan vulnerabilities
DEBUGGING	Prescribing remedies for malfunctions
REPAIR	Executing a plan to administer a prescribed remedy
INSTRUCTION	Diagnosis, debugging, and reparing student behavior
CONTROL	Interpreting, predicting, repairing, and monitoring system behaviors

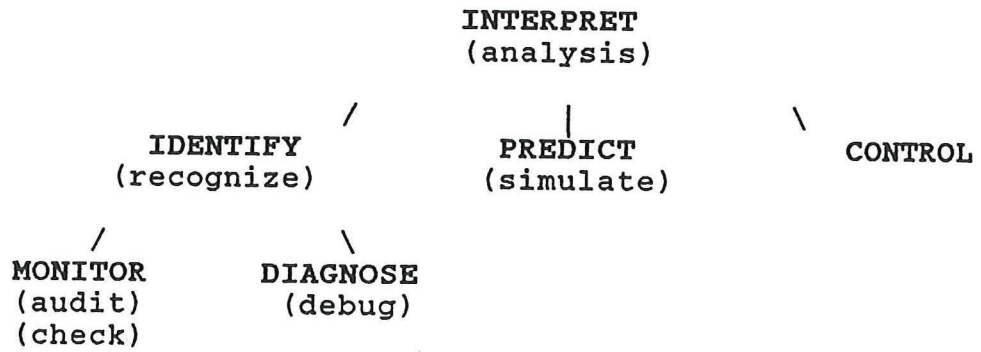
figuur 4.1: probleem categorieen
(uit Hayes-Roth et al. 1983)

Elk van deze taken wordt gekenmerkt door eigen problemen, en suggereert een andere architektuur van het kennis-systeem.

Clancey (1985) bekritiseert deze indeling op verschillende gronden, en formuleert een andere indeling (zie fig.4.2/4.3):



figuur 4.2: generische operaties voor de synthese van een systeem (uit Clancey 1985)



figuur 4.3: generische operaties voor de analyse van een systeem (uit Clancey 1985)

Clancey onderscheidt twee belangrijke groepen: de operaties die een systeem ontleden (analyseren), en operaties die een systeem in elkaar zetten (synthetiseren). In tegenstelling tot de taken in de indeling van Hayes-Roth is elk van de taken die Clancey beschrijft primitief (bestaat niet uit een combinatie van andere taken); bovendien laat zijn indeling de intrinsieke relatie zien die er bestaat tussen taken. Zo is **PLANNING** een vorm van **DESIGN**, iets dat niet duidelijk wordt uit Hayes-Roth's indeling.

Taken kunnen ook gekarakteriseerd worden door combinaties van bovenstaande primitieve taken. Clancey noemt als twee belangrijkste: de *construktie-cyclus* , die bestaat uit **SPECIFY** + **DESIGN** (+ **ASSEMBLE**), en de *onderhouds-cyclus* , die bestaat uit (**MONITOR** + **PREDICT** +) **DIAGNOSE** + **MODIFY**. De laatste cyclus is

typisch voor medische programma's als b.v. MYCIN.

Breuker c.s. breiden Clancey's onderscheidingen nog wat verder uit, en komen tot drie hoofdgroepen: *system analysis, system modification,* en *system synthesis.*

Het classificeren van taken is erg belangrijk voor de toekomst van het kennis-systeem onderzoek. Veel van de bestaande inference engines grijpen nog terug op de methoden van MYCIN, en die zijn veel te beperkt om complexere taken behoorlijk mee uit te voeren. Als het kennis-systeem onderzoek de gewekte verwachtingen waar zal willen maken, dan zullen er nieuwe formalismen ontwikkeld moeten worden. Hiervoor is het van het grootste belang, dat er enige struktuur wordt aangebracht in de lopende onderzoeken naar de aard van allerlei taken, zodat de resultaten van die onderzoeken direkt met elkaar in verband gebracht kunnen worden. Doel is dan, om formalismen te ontwikkelen voor elk van de generische taken, zoals b.v. Clancey die onderscheidt, zodat toekomstige kennis-systemen op welk domein dan ook snel geconstrueerd kunnen worden.

De KADS-methodologie en het interpretation models idee vormen een eerste aanzet hiertoe. Helaas is het niet zo, dat bovengeschetste ontwikkeling - van het classificeren van taken naar het ontwikkelen van nieuwe formalismen - zich al op volle kracht heeft doorgezet. Uit het vrij recente werk van de KADS-groep wordt duidelijk, dat het voornamelijk de <u>analytische</u> taken zijn, die op het moment enigszins in kaart zijn gebracht: identifikatie-, classifikatie-, en diagnose-taken. Dit is niet verwonderlijk, omdat over deze typen taken het meest bekend is vanuit het MYCIN-onderzoek. Met de synthese-taken is het veel somberder gesteld op dit moment. Breuker c.s. schrijven: "Much less is known about synthetic processes than about analytic processes to date. There are many cognitive models and expert systems for analytic tasks, very few exist for synthetic tasks" (Breuker 1987, blz.91).

Nu is dit nauwelijks een problematisch punt: het is immers aan de onderzoekers om meer over synthese taken te weten te komen, en de eventuele lacunes in te vullen. Echter, het is precies dit laatste waarbij een onderzoeker in de problemen kan komen, als hij probeert de taak te beschrijven in de termen die door het model geleverd worden.

In de beschrijvingen van de inferentiestruktuur van generische taken (de karakterisering van de taak in termen van metaklassen en KS-en, en hun onderlinge relatie) die Breuker c.s. geven valt op, dat het abstraktienivo van die beschrijvingen over taken niet uniform is. Dit heeft vooral betrekking op de termen, die gebruikt worden om de KS-en mee aan te duiden.

Op blz.22 schetste ik een typologie van KS-en, zoals die door Breuker c.s. gegeven wordt. Een dergelijke typologie kan alleen enige betekenis hebben, als 1. deze min of meer compleet is, en de typische KS-en ook inderdaad gebruikt worden ter karakterisering

van taken; en 2. als de lading die door de naam van de KS gedekt wordt min of meer constant blijft. Over beide punten valt wel wat op te merken.

Met name bij het beschrijven van synthese taken wordt het met de oorspronkelijk geformuleerde typologie niet meer zo nauw genomen. In de inferentiestruktuur van verschillende soorten **DESIGN**-taken worden resp. de volgende KS-en genoemd: 'transformation and elaboration', 'select and aggregate', 'transform/expand/refine', 'select', 'decompose', 'specify', 'design', 'aggregate' en 'match' (Breuker 1987, blz.95-99). Van deze 9 zijn er 6 niet in de typologie terug te vinden. Omdat de KS-en geacht worden een aanduiding te zijn van primitieve inferentie-processen, lijkt het niet verstandig om zo gemakkelijk om te gaan met het introduceren van nieuwe namen.

Het tweede punt betreft de betekenis van de termen die gebruikt worden om de KS-en mee aan te duiden. Een KS wordt gekarakteriseerd door haar input, haar output, de methoden die worden gebruikt om uit de eerste de laatste te verkrijgen, en de domein-kennis die daarbij wordt gebruikt. Als het zin wil hebben om te kunnen spreken van een *typologie* van KS-en, dan moet op zijn minst de naam van de KS enige relatie hebben met de voor die KS karakteristieke eigenschappen, met name met de in de KS gebruikte methoden. Dit is niet altijd het geval. In de beschrijving van synthese-taken vinden we de volgende KS-en met de hierin gebruikte methoden:

knowledge source: **tranformation and elaboration**
methods: Currently we have no idea how this should be done. Probably this part of the design is highly domain dependent.

knowledge source: **select and aggregate**
methods: Any methods that are known to us are very domain specific.

knowledge source: **transform/expand/refine**
methods: Although various methods are used in this stage of the design process these are either very general, or domain specific.

(drie KS-en in een design-proces, blz.95/96).

Zo gesteld betekent dit, dat de naam van een KS in principe geen enkele formele betekenis heeft. Het zijn eerder de intuities van de gebruiker van het model, die voor de betekenis moeten zorgen. De term 'transformatie' is niet meer dan een funktionele aanduiding van een proces.

Uiteraard heeft het bovenstaande niet betrekking op alle beschrijvingen van KS-en. Zeker wat KS-en in analytische taken betreft is men veel nauwkeuriger en volledediger in het geven van beschrijvingen. Dit komt in niet onbelangrijke mate voort uit het

feit, dat van deze taken veel meer bekend is vanuit het kennis-systeem perspektief. KS-en kunnen in dit geval beschreven worden in termen, die ontleend zijn aan reeds bestaande systemen, en refereren aan bekende methoden en technieken. Taken die worden beschreven vanuit een cognitief perspektief, en waarbij men het voordeel van reeds bestaande systemen niet heeft, moeten het dus noodzakelijkerwijs doen met een wat abstraktere terminologie.

Gevolgen voor het gebruik van het model

Een belangrijke vraag voor de gebruiker van het model is nu, op welk nivo van abstraktie een nieuw te bestuderen taak beschreven moet worden. Moet de beschrijving van de inferentie-struktuur zo gedetaileerd zijn, dat van elke metaklasse en elke KS precies de eigenschappen bekend zijn? Het voordeel hiervan is, dat de resulterende beschrijving al heel dicht bij een eventueel kennis-systeem staat. Of moet in de beschrijving het detail achterwege gelaten worden? Dit kan resulteren in een beschrijving, die weliswaar veel verder van het kennis-systeem af staat, maar die wel het voordeel heeft dat de overeenkomsten met andere taken duidelijker naar voren komen.

Voor het bestuderen van plannen bij bridge heb ik gekozen voor een benadering, die resulteert in een taak-beschrijving van de hierboven genoemde tweede soort. Dit heeft twee redenen. Ten eerste is er bij de huidige 'state of the art' nog zo weinig over gelijksoortige taken bekend, dat het uiterst twijfelachtig is of de mate van detail die bereikt is bij het beschrijven van analytische taken hier wèl gehaald kan worden. Een poging in deze richting zou te gemakkelijk kunnen resulteren in vrij onvolledige beschrijvingen van de soort die ik in het bovenstaande stuk bekritiseerd heb. Ten tweede is het m.i. belangrijker om een beschrijving te geven die een bijdrage kan leveren aan de classifikatie van taken. Dit heeft als gevolg dat 1. uit die beschrijving zoveel mogelijk de overeenkomsten duidelijk moeten worden met andere taken; en 2. geprobeerd moet worden de lacunes die er bestaan in de beschrijvingen van gelijksoortige taken zo veel mogelijk in te vullen. In de nu volgende analyse zullen deze twee doelen waar mogelijk nagestreefd worden.

HOOFDSTUK 5: EXPERTS AAN HET WERK

5.1 taakomschrijving

De taak die in dit onderzoek wordt bestudeerd, betreft een aspekt van het afspel bij bridge. Bridge is een kaartspel voor 4 spelers, die opereren als twee 'paren'; het paar Noord-Zuid (de windrichtingen duiden de plaats van de spelers ten opzichte van elkaar aan tafel aan) speelt tegen het paar Oost-West. Iedere speler krijgt bij aanvang van het spel 13 kaarten gedeeld; hij kan slechts deze 13 kaarten zien. Het spel kent twee fasen.

In de biedfase wordt door een van de twee paren een bindende afspraak aangegaan met betrekking tot het in de latere speelfase te behalen slagen. Een voorbeeld van zo'n afspraak (een 'kontrakt') is: 'wij, Noord-Zuid, verplichten ons om 10 slagen te halen met schoppen als troef'. Als dit lukt (het paar 'haalt het kontrakt') dat krijgen zij hiervoor een zeker aantal punten; lukt het niet (het paar 'gaat down') dan krijgt het andere paar punten.

In de speelfase wordt door het ene paar geprobeerd om het in het kontrakt genoemde aantal slagen ook inderdaad te halen; het andere paar probeert dit te verhinderen. Een 'slag' is een basisbegrip in ieder kaartspel: het is een reeks van (in het geval van bridge vier) kaarten, één van elke speler. Degene die van de vier de hoogste kaart heeft gespeeld, 'wint de slag'. In totaal zijn er dus 13 te verdelen slagen. Willen Noord-Zuid in het genoemde voorbeeld dus hun doel realiseren, dan moeten zij 10 van de 13 slagen halen; Oost-West op hun beurt moeten proberen om er 4 te halen (immers dan halen Noord-Zuid er slechts 13-4=9, en zijn dus down).

Als de speelfase begint voltrekt zich een verandering in het kontrakt te halen paar: een van hen legt zijn 13 kaarten voor iedereen zichtbaar op tafel en doet verder niet meer mee (hij wordt 'dummy'); de ander wordt 'spelleider', en is degene van het paar die de arbeid moet verrichten. De situatie in de speelfase is dus als volgt: twee verdedigers nemen het op tegen de leider; zowel de leider als de verdedigers zien 26 van de 52 kaarten, nml. hun eigen 13 en de 13 van de dummy. Overigens legt de dummy zijn kaarten pas open nadat de speler links van de leider de eerste kaart heeft gespeeld (nadat hij 'is uitgekomen'). De taak van de leider is om ervoor te zorgen dat aan het eind van het spel, als er 13 slagen gespeeld zijn, hij de bezitter is van 10 van die 13 slagen.

Bridge is, in de termen van von Neumann en Morgenstern (1944), een 'onvolledig informatie-spel'. Dit wil zeggen, dat de spelers niet beschikken over alle informatie over het spel. Bij schaken en Go bijvoorbeeld zijn alle stukken voor beide spelers zichtbaar; bij bridge echter is voor iedere speler maar de helft van het aantal kaarten bekend. Dit maakt het spel totaal anders van karakter. Omdat de leider niet weet, wat de kaarten van Oost-West zijn,

wordt zijn taak door twee problemen gekarakteriseerd: 1. hij moet proberen om wat meer van die kaarten te weten zien te komen, en 2. hij moet zorgen dat zijn afspel slaagt bij een zo groot mogelijk aantal distributies van de OW-kaarten.

Voor het eerste probleem zijn er verschillende oplossingsmethoden; de vaardigheid staat bekend als 'kaartlezen'. Dit aspekt van het spel zal in de te bestuderen taak niet ter sprake komen (Quinlan (1979) schreef een kaartlees-programma dat behoorlijk funktioneerde). Het tweede probleem is karakteristiek voor de te bestuderen taak.

Het eerste wat een expert doet als de dummy open gaat, is het maken van een speelplan. Dit maken van een plan is een normale aktiviteit als een bepaald doel gerealiseerd moet worden. Op het gebied van plannen is al veel onderzoek gedaan, maar de taken in dit onderzoek (b.v. Sacerdoti 1977) waren eenvoudig, en werden gemaakt in een omgeving van volledige informatie. Plannen bij bridge is aanmerkelijk ingewikkelder, omdat van de geplande reeks handelingen niet zeker is, of deze het gestelde doel wel zal realiseren. Een plan bij bridge hoeft dus niet met 100% zekerheid tot sukses te leiden; het plan moet in een zo groot aantal mogelijke gevallen tot sukses leiden.

Het maken van een speelplan is vergelijkbaar met het vullen van een koffer in voorbereiding van een lange reis. Stel je voor, dat iemand een jaar op reis gaat rond de wereld. De kans is groot, dat hij in dat jaar de meest uiteenlopende weersomstandigheden zal tegenkomen. Om op elk weertype voorbereid te zijn zou het mooiste een koffer vol kleren zijn, maar omdat het onmogelijk is om op een wereldreis met een enorme koffer rond te sjouwen (en onze held zich geen sjouwers kan veroorloven), zal de beperkte ruimte die er in de koffer is goed benut moeten worden. De meeste ruimte zal gevuld moeten worden met kleren, die dienen voor de weertypes die het vaakst voorkomen. De ruimte die daarna nog over is kan gebruikt worden voor de weertypes die minder vaak voorkomen, maar nog wel vaak genoeg om rekening mee te houden. Een optimale kombinatie van kleerkeuze en ruimte zou bijvoorbeeld kunnen resulteren in een koffer die resulteert in comfort in 90% van de mogelijke gevallen. Met maar een heel klein koffertje zal je misschien genoegen moeten nemen met maar 50%, terwijl een hele grote koffer resulteert in 100% comfort.

In het plannen van een bridgespel stellen de kaarten van de spelleider de koffer voor, de kaarten van de tegenstanders het weer, en het plan de samenstelling van de koffer. Soms zijn de kaarten van de leider zo goed, dat bij een zorgvuldige inrichting van de koffer 100% sukses gegarandeerd is, ongeacht de verdeling van de kaarten van de tegenstanders. Soms ook is de koffer zo klein, dat gehoopt moet worden op een bepaalde distributie van de tegenstanders' kaarten, wil het plan slagen. Hopen op een situatie die in 80% van de gevallen voorkomt wordt natuurlijk vaker beloond dan hopen op een situatie van 50%. Zelfs met een grote koffer kom je voor problemen te staan, als je er alleen maar regenjassen

ingooit...

Een speelplan moet dus voldoen aan de eis, dat het het doel realiseert bij een zo groot mogelijk aantal distributies van de vijandelijke kaarten. De taak van de leider is het maken van een dergelijk plan. Het komt regelmatig voor, dat het beste speelplan al voor de eerste slag gemaakt kan worden. Zo'n speelplan leidt tot sukses bij een bepaalde klasse mogelijke distributies, die samen bijvoorbeeld een frekwentie van voorkomen hebben van 80%. Dit plan kan het afspel leiden tot op het moment dat blijkt, dat de werkelijke distributie niet behoort tot de distributies waarbij het gemaakte plan slaagt. In dat geval moet het plan veranderd worden. Zolang de werkelijke distributie overeenkomt met de geanticipeerde distributie, behoeft van het oorspronkelijke plan niet te worden afgeweken.

5.2 Een gedetaileerd protokol

Om een eerste indruk te krijgen hoe experts te werk gaan bij het maken van een speelplan werd aan 3 spelers het volgende spel voorgelegd:

♠ H 2
♥ A 7 6
♦ 9 7 6 4 3
♣ A 4 2

N

Z

♠ A 5 4
♥ H B 8 4 3
♦ H V
♣ V 5 3

(probleem 1)

De procedure was als volgt. De spelers werd meegedeeld dat zij een hand moesten bieden en daarna afspelen. De wedstrijdvorm die gespeeld werd was 'viertallen', wat wil zeggen dat het doel bij het afspel het halen van het kontrakt was (i.t.t. 'paren'). Er werd hun verzocht om bij zowel bieden als spelen hardop te denken. De spelers kregen eerst de zuid-hand. Deze kregen zij als 13 ongesorteerde kaarten.

De reden dat de spelers de hand ook moesten bieden was de volgende. In de experimenten van de Groot (1946) klaagden enkele proefpersonen over het feit, dat zij zomaar midden in een stelling gegooid werden, zonder dat zij de voorgeschiedenis er van kenden.

Om zoveel mogelijk de normale wedstrijdsituatie te benaderen werd de hand dus eerst geboden; dit maakte de speler alvast bekend met zijn eigen 13 kaarten, en het vermeed eventueel storende vragen over het (dan onbekende) biedverloop. Bij het bieden fungeerde de proefleider als partner van de speler; deze was bekend met het normaal door de spelers gehanteerde biedsysteem. De biedingen van de tegenstanders werden ook door de proefleider uitgesproken.

Als het eindkontrakt bereikt was speelde de proefleider de kaarten van de twee verdedigers. De spelers werd verteld, dat zijn tegenstanders twee goede spelers waren; zij moesten er van uit gaan, dat ondanks het feit dat de proefleider hun kaarten bijspeelde, deze tegenstanders net als in een normale spelsituatie niet 'door de kaarten heen konden kijken'. De linker tegenstander kwam uit, en de dummy werd opengelegd. Vervolgens werd het spel onder hardop denken afgespeeld.

De proefpersonen waren allen goede spelers; twee van hen (M.V. en B.M.) spelen in de één na hoogste competitie klasse (Nederlandse subtop); de derde (E.L.) speelt in het Nederlands team. Zij hadden geen van allen problemen met het hardop denken, zoals hen na afloop werd gevraagd. Aanwijzingen dat dit inderdaad niet het geval was kunnen gehaald worden uit de wijze waarop over de 'tegenstanders' werd gepraat: hoewel deze niet aanwezig waren doen de opmerkingen van de proefpersonen anders vermoeden (b.v. B.M. r.19, 42; E.L. r.20, 29).

In probleem 1 boden alle drie de spelers tot het eindkontrakt van 4♥, via het volgende biedverloop:

Zuid	Noord
1♥	2♦
3SA	4♥

Ik zal hieronder alle drie de protokollen weergeven; uit alle drie is het eerste gedeelte, waarin de proefpersonen aan het bieden waren, weggelaten.

Het protokol van E.L. is het interessantste, en ik zal dit protokol als uitgangspunt van de analyse gebruiken. In de linker kolom hieronder staat het protokol zoals dat werd afgenomen; in de rechter de explicitering van de gemaakte denkstappen etc. De andere twee protokollen volgen daarna, zonder de explicitering.

(♠V)

1 even kijken	telt z'n losers
2 een ruiten loser	
3 anderhalve klaver loser	
4 de hartens ook nog ingewikkeld jezus	
5 even denken hoor	

6	ik kan gewoon een schopje aftroeven, een harten snitje nemen	stelt plan 1 voor: harten-snit
7	ik kan op de ruitens spelen	stelt plan 2 voor: ruiten ontwikkelen
8	even denken	werkt plan 2 uit
9	als ik ♠A neem, ♦H, schoppen voor de H, ruiten voor de V	
10	uhm hoe zou ik het dan moeten spelen	probeert situatie te evalueren (als West de ruiten aftroever over-troeft)
11	als ik overtroevers krijg dan ben ik niet blij	tweede poging
12	ik kan natuurlijk ♠H, ♠A, ♦V voor de aas, iets na	Oost heeft ♦A (is ver-velend, klaver switch).
13	dan hoop ik niet dat rechts...	daarom: stel West heeft
14	nou stel links	♦A; hij speelt schoppen
15	bijvoorbeeld schoppen voor de H, ruiten voor de V	evaluatie-poging
16	uuh	vervolg 1 plan 2: slecht
17	schopje aftroeven is ook klote	vervolg 2: troeftrekken
18	♥H, ♥A, ruitje aftroeven schopje aftroeven ruitje aftroeven	
19	lijkt wel aardig te gaan	goed
20	maar als rechts aan slag komt ben ik niet blij als die door de klavers heen speelt	maar als Oost ♦A heeft.
21	maar dat lijkt niet zo gek	is nog niet gebeurd
22	dan kan ik dus gewoon een harten, een ruiten en eventueel een klaver afgeven	recapituleert resultaat van vervolg 2
23	dan moet ik gewoon hopen dat als er overgetroefd wordt dat het van de driekaart troef is	? (analyse klopt niet)
24	dan gaat het in principe wel goed	
25	maar er kan nog van alles gebeuren	
26	nou ik kan eventueel altijd terug-vallen	als ♣H rechts zit blijft die kans behouden
27	ja ik kan wel gaan rotzooien	
28	nou ik probeer die ruiten (♠2) (♠6) (♠A) (♦V)	
29	je weet maar nooit misschien zakt die eikel (♦2)	
30	dat is weer jammer (♦3) (♦A)	
31	niet gunstig (♠3)	
32	dat valt mee, schoppen terug, dat gaat goed (♠5) (♠9) (♠H)	gevreesde klaver switch vindt niet plaats
33	♠9, hm, kunnen 6-2 zitten	
34	dan ben ik weer niet blij	
35	nou ik begin met een ruitje (♦4) (♦5) (♦V) (♦B)	
36	even kijken	
37	als de hartens nu 3-3 zitten	bedoelt: 3-2

38 ♥H, harten voor de A, kan ik een ruitje troeven, een schopje troeven	nog even eerdere analyse checken
39 ja als dan de driekaart harten aan de rechterkant zit bij de doubleton schoppen	plan faalt dan als ruiten 4-2 zitten: een entree tekort
40 dan ben ik weer de lul	
41 dan moet die ♠H maar goed zitten	de overgebleven kans
42 klopt dat dan dan verlies ik een klaver en een harten	
43 nou ik ga sowieso	
44 misschien moet ik wel ♥H, ♥A eerst spelen	
45 en als die hartens dan 4-1 zitten	evalueert vervolg 2 bij een slechtere distributie: lijkt niet goed, maar
46 nou als die hartens 4-1 er achter zitten ben ik sowieso ongeveer al down	1 ruiten, 2 harten-losers; met de ruiten 3-3 toch gehaald?
47 1 2 3	denkt van niet
48 oh, dan kunnen de ruitens nog 3-3 zitten	
49 nee dat lukt ook al niet	
50 nou dat zie ik wel	
51 ik denk dat ik ♥H speel (♥H) (♥2) (♥6) (♥5) (♥3) (♥9) (♥A) (♥10) (♦6)	voert vervolg 2 uit
52 ik heb trouwens ook niet op die ruiten gelet	
53 nou bekijk het maar (♦8)	
54 ik heb 'm nou toch ongeveer altijd gemaakt (♥4) (♠6) (♠5) (♠B) (♠7) (♠7)	
55 dat valt weer mee	
56 en een ruitje	met sukses

Probleem 1, B.M., 8-2-87

(♠V)

1 ruiten loser
2 2 klaver losers
3 ja, en harten is nog onbekend
4 2 harten losers, 1
5 misschien ook maar 1 klaver loser
6 ik zal in ieder geval iets met die ruiten moeten doen
7 effe kijken hoor
8 die harten kunnen 4-1 zitten
9 dat is wel lastig
10 ik denk dat ik gewoon op de ruiten afga
11 als ik een overtroever krijg is dat misschien niet zo erg
12 misschien van een vaste troefslag

13 moet ik nog iets met die klaver verzinnen, kan ik ook nog een slag extra in maken
14 en die troeven moet ik in ieder geval maar eerst even van afblijven, dat kan ik altijd nog wel zien
15 even controleren
16 schoppen voor mij, ruiten
17 dat is lastig om te combineren
18 nou, ik doe een kleintje
(♠2) (♠6) (♠A)
19 nou ♦V, misschien wordt er wel gezakt
(♦V) (♦2) (♦3) (♦A)
20 lullig dat die ruiten gepakt is
(♠6) (♠4) (♠9) (♠H)
21 nou ik speel een ruiten
(♦4) (♦5) (♦H) (♦B)
22 ♦B dat is lullig
23 nou moet ik even kijken
24 ze lijken 4-2 te zitten aan de verkeerde kant
25 zou ik misschien toch nog iets met die klaver moeten doen
26 schoppen getroefd, dan kan ik eerst een klaver spelen
27 dat had ik misschioen trouwens gelijk wel moeten doen
28 waarom deed ik dat niet gelijk
29 omdat ik een beetje te intuitief speel denk ik
30 nou, schoppen troeven, ruiten troeven
31 als die wordt overgetroefd van de driekaart
32 maar ja, dan moet klaver heer maar goed zitten
33 dat is niet zo erg
34 nee, ik troef gewoon een schoppen
(♠4) (♠10) (♥6) (♠7)
35 moet ik nou eerst een kleine klaver spelen?
36 nee, ik speel toch ruiten, intuitief
(♦6) (♦10)
37 maakt het nou uit met welke ik troef?
38 nee ik denk het niet
(♥3) (♥9)
39 troef 9
40 4 schoppens, 2 ruiten
(♥2)
41 de 2 dat is merkwaardig, zeer merkwaardig
42 als die nou de tien in vieren heeft ben ik nog steeds de lul
43 ja dat is stom
44 ik doe toch maar een kleintje
(♥7) (♥10) (♥B)
45 nou
(♥A) (♥5)
46 hee dat is te gek
47 nou troef ik een ruitje hoog
48 ik geef nog een klaver af
49 het is gunstig gegaan

Probleem 1, M.V., 3-2-87

(♠V)

1 ♠V uitkomst
2 eens even kijken
3 ik heb..
4 hm wat een rothand
5 een potentiele harten verliezer
6 een potentiele.. ♦A is weg
7 ik kan een schopje in dummy introeven
8 en ik heb nog een potentiele twee potentiele klaver verliezers
9 het lijkt me niet in ieder geval niet een handje om een soort safetyplay in harten te nemen, ik heb bovendien ♥9 niet
10 even kijken
11 als ik een schoppen troef
12 moet ik toch van die klavers af
13 ik kan misschien een ruiten ontwikkelen
14 wat gebeurt er als ik ♠A pak dan ruiten speel, komt schoppen terug dan kan ik met ruiten naar m'n hand
15 H H harten naar de aas troef ik een ruitje troef ik een schopje troef ik weer een ruitje
16 en dan ja als de ruitens 4-2 en de hartens 3-2 zitten heb ik het altijd gehaald
17 ik moet ♠A in mijn hand pakken
(♠2) (♠6) (♠A)
(♦H) (♦2) (♦3) (♦A)
18 ♦2 dan zouden ze 3-3 kunnen zitten
(♠3) (♠4) (♠9) (♠H)
19 deblokkeer ik eerst de ruiten
(♦4) (♦5) (♦V) (♦B)
20 ♦B eens kijken of het in het speelplan nu klopt
21 ♥H, ♥A ruiten troef schoppen troef
22 ja okee
(♥H) (♥2) (♥6) (♥5)
(♥3) (♥9) (♥A) (♥10)
23 prima zo
(♦6) (♦8) (♥4) (♣6)
(♠5) (♠10) (♥7) (♠8)
24 de ruitens hoog gemaakt
(♦7) (♦10) (♥8) (♠B)
25 en ik heb nog steeds de harten controle zolang ik niet met harten van slag ga
26 nu moet alleen ♣A niet worden afgetroefd
(♣3) (♣8) (♣A) (♣7)
27 ♦9 gooi ik een klavertje af en ik geef nog 2 slagen af

Het protokol van E.L. zal als uitgangspunt van de analyse dienen.

Het speelplan wordt geformuleerd in de regels 1-28, nog voor er een kaart is gespeeld. Dit plan wordt goed bevonden (r.24), en blijft vervolgens in het verdere spel gehandhaafd. In de regels

29-56 wordt de eerder gemaakte analyse van het spel wel continu gecontroleerd op eventuele fouten, en wordt nagegaan of nieuwe informatie over de handen van de tegenstanders een aanpassing van het plan noodzakelijk maakt. Dit is niet het geval, zodat de gespeelde kaarten geheel in overeenstemming zijn met het plan zoals dat oorspronkelijk gemaakt werd.

Het protokol kan ons op 3 manieren inzicht geven in het proces van oplossen en de kennis die daarbij gebruikt wordt.

- De opmerkingen in het protokol kunnen aanleiding geven tot het veronderstellen van bepaalde aktiviteiten (zie bijvoorbeeld de tweede kolom). Elk van deze aktiviteiten heeft een plaats in de taakstruktuur van het oplossingsproces, en door het groeperen van opmerkingen kan dus in ieder geval een gedeelte van de totale taakstruktuur aan de oppervlakte gebracht worden.
- In de opmerkingen worden verschillende soorten feiten genoemd. Deze verschillende soorten feiten geven een beeld van de organisatie van de domeinkennis: de concepten en de relaties die er daartussen bestaan.
- De volgorde waarin die kennis in de opmerkingen naar voren komt kan ons informatie geven over de inferentiestruktuur van de taak.

Uiteraard is het interpreteren van protokollen niet een bezigheid die alleen maar gebaseerd is op het protokol, vanwege de vele lacunes en vaagheden. Waar nodig zal aanvullende informatie gehaald worden uit gesprekken met de proefpersonen over hun protokol, en uit eigen ervaring (de schrijver dezes is ongeveer een even sterke speler als de proefpersonen M.V. en B.M.).

Aktiviteiten

In het protokol kunnen de volgende aktiviteiten onderscheiden worden:

1-5	losers tellen
6	voorstel plan1: ♠ troeven, ♥ snit
7	voorstel plan2: op de ♦ spelen
8/9	uitwerking plan2: ♠A, ♦H, ♠H, ♦V (situatie1)
-	(vervolg plan2 na situatie1: troef ♦ af in hand)
11	evaluatie plan2-na-situatie1: als overtroevers dan -
12-16	opnieuw uitwerking plan2: ♠A, ♦V; als Oost ♦A heeft kan hij ♠ naspelen: evaluatie - --> assumptie: West heeft ♦A (kan geen ♠ naspelen); ♠H, ♦V (weer situatie1).
17	vervolg plan2-na-situatie1: ♠ aftroeven; evaluatie -
18	vervolg2-plan2-na-situatie1: ♥H, ♥A, ♦troef, ♠troef, ♦ troef.
19	evaluatie +
20	trekt assumptie uit 12-16 in: evaluatie -

21-23 (want dan gaat vervolg2 niet door wegens entree-gebrek); als Oost aan slag komt en hij speelt ♣ na (situatie2): ♠A, ♦V, ♠H, ♦ troef. evaluatie + als de overtroever van de driekaart troef is.
24-28 besluit plan2 uit te voeren

Zoals te zien valt, zijn er in het denkproces drie hoofdaktiviteiten te onderscheiden, nml. 1. losers tellen, 2. voorstellen van mogelijke plannen, en 3. het evalueren, repareren, uitbreiden, enz. van plannen. Deze aktiviteiten komen in alle drie de protokollen voor:

	losers tellen	plan voorstel	evaluatie etc.
E.L.	r.1-5	r.6-7	r.8-23
M.V.	r.1-9	r.10-13	r.14-17
B.M.	r.1-5	r.6,10,13	r.15-17,25-35

Als eerste stap in het analyseproces zal ik van elk van deze drie de aard van de gebeurtenissen beschrijven, die binnen de aktiviteit plaats vinden. Deze beschrijving heeft een vrij informeel karakter, m.i. noodzakelijk omdat zonder een goed globaal beeld van het oplossingsproces en de hierbij noodzakelijke kennis, het toepassen van de in het 4 lagen-model gebezigde termen niet tot de gewenste duidelijkheid zouden leiden.

Fase 1: r.1-4

Over het gebeuren in de eerste fase zijn uit het protokol 2 soorten aanwijzingen af te leiden: 1. de uitingen 1-4 van de proefpersoon; en 2. het feit dat wat er gebeurd is in deze fase voldoende moet zijn om fase 2 te laten beginnen. Het zal blijken dat de processen die ten grondslag liggen aan de uitingen slechts een fraktie vormen van het totaal van de processen in deze fase.

Naar analogie van de door Reitman en Wilcox (1978) gebruikte termen zal ik deze fase de 'perceptie-fase' noemen. In de perceptie-fase inventariseert de expert zijn gezamenlijke bezit, formuleert op basis hiervan een reeks te realiseren doelen, en stelt een lijst op van mogelijke middelen die voor dit laatste zorg dragen.

De 26 kaarten van de leider (z'n eigen 13 en de 13 van de dummy) vormen geen ongestruktureerd geheel. Ten eerste behoort elke kaart tot een van de vier kleuren: klaver, ruiten, harten en schoppen; ten tweede zijn alle kleuren gerangschikt op volgorde. De kaarten van een afzonderlijke kleur (kortweg: een kleur) vormen de omgeving waarin het merendeel van de perceptuele aktiviteit plaats

vindt. Deze omgeving is zelfstandig: binnen een kleur kan een complete analyse plaats vinden, die onafhankelijk is van de analyse van de andere kleuren. Zo'n analyse zal ik een lokale analyse noemen.

Omdat het doel van de speelfase het halen van een zeker aantal slagen is, is de analyse van een kleur dan ook in eerste instantie gesteld in termen, die direkt of indirekt te maken hebben met het halen van slagen. Een voorbeeld:

A 7 6

H B 8 4 3

Dit is de hartenkleur uit het voorbeeldspel. In de lokale analyse van deze kleur wordt gebruik gemaakt van de volgende concepten:

winners:
een winner is een kaart die zeker een slag zal winnen, omdat de tegenstanders geen hogere kaarten bezitten. In het voorbeeld zijn dat er twee: de aas en de heer.

losers:
het aantal losers van een kleur is het aantal slagen dat aan de tegenstanders zal moeten worden afgestaan. Hoeveel dit er zijn is van een aantal zaken afhankelijk, nml.

- Welke partij de kleur aanspeelt
- De speelsoort van het kontrakt
- De verdeling van de kaarten van de tegenpartij, en
- De manier waarop de kleur gespeeld wordt.

1. Als in een troefkontrakt de positie: H x x vs. B x x door de tegenstanders wordt aangespeeld, behoeft het maximum aantal losers (aftroevers negerend) nooit groter dan 2 te zijn. Als de positie zelf wordt aangespeeld is de kans op meer dan 2 losers 25% (zie ook punt 3).

2. Omdat in een SA-kontrakt de controle van de troefkleur niet aanwezig is, kan het maximaal aantal losers in zo'n kontrakt groter zijn dan in een troefkontrakt (de aftroevers die voor extra losers zorgen in een troefkontrakt worden in de lokale analyse niet geteld). Overigens komt het 'losers-tellen' ter bepaling van de strategie slechts in troefkontrakten voor.

3. Deze is uiteraard van grote invloed op het aantal losers in een kleur. Bij het toepassen van de standaardspeelwijze voor het maximaal aantal slagen (zie punt 4) levert deze kleur 5 slagen op als Oost V x x of V x heeft; in andere gevallen slechts 4 of zelfs maar 3. De distributie van de kaarten van de tegenpartij - het 'zitsel' - heeft een bepaalde a priori kans van voorkomen. Deze kansen zijn bij experts bekend.

4. Kaarten op zich zijn maar een statisch geheel. Een slag wordt

pas gehaald als een kaart gespeeld wordt. Een reeks in serie gespeelde kaarten wordt een speelwijze genoemd. Binnen een kleur zijn er diverse speelwijzen mogelijk. Zo kan je in de voorbeeldkleur eerst de aas spelen, en dan de heer; je kan de aas spelen en dan een kleintje naar de boer; je kan uit beide handen een kleintje spelen; enz. Een speelwijze heeft 2 belangrijke kenmerken, nml. het aantal slagen dat de kleur oplevert als die speelwijze wordt toegepast, en de kans op het zitsel dat voor dit aantal slagen nodig is. Welke speelwijze in een kleur het meest suksesvol is hangt natuurlijk af van het zitsel. Als dit bekend is, is het een koud kunstje om het maximale aantal slagen te halen. Vaak komt het voor, dat de leider genoeg informatie heeft om het zitsel af te leiden; hij kiest dan een overeenkomstige speelwijze. In alle andere gevallen moet hij afgaan op de a priori kansen.

In de voorbeeldkleur hebben Oost-West 5 harten, en dus 26-5 =21 andere kaarten. Als je de 26 kaarten verdeeld in 2 hoopjes van 13, dan is de kans dat de harten 3-2 verdeeld zijn over de 2 stapeltjes ca. 68%. De kans op een 4-1 zitsel is 28%, en de kans op een 5-0 zitsel 4%. Deze a priori kansen gelden slechts als er voor de rest niets bekend is van de 2 handen. Als bijvoorbeeld bekend is, dat Oost 6 van de 8 ontbrekende schoppen heeft, dan neemt de kans dat hij 3 harten heeft af volgens Bayes' theorema (Borel en Cheron (1940) geven een uitgebreid overzicht van de mathematische aspekten van bridge).

Op deze manier valt precies uit te rekenen, welke speelwijze gezien de a priori distributiekansen het vaakst sukses heeft, en uiteraard ook in hoeveel procent van de gevallen deze speelwijze leidt tot het verlies van 1 slag, 2 slagen, enz. In de volgende tabel (uit de Encyclopedia of Bridge (Kearse (ed.) 1976): suit combinations) komt de kennis van een expert van zo'n kleurcombinatie goed tot uiting:

dummy declarer	tricks required	play	% chance of success	tricks per deal
A Q 9 8 J x x x	4	Lead the jack. If it is covered, finesse the nine; if it holds, lead small to the nine.	27	
	3	Lead small to the queen. If it loses, cash the ace; if it holds, lead to the nine	97	
	Max	Both the above lines produce		3.16

Variabelen in deze tabel zijn: het gewenste aantal slagen; de bijbehorende speelwijze, met een slagingspercentage; en een 'optimale' speelwijze, die gemiddeld het meeste aantal slagen oplevert, + dit aantal.

Dit type kennis is bijzonder belangrijk voor goed bridge. Het vormt het materiaal waarmee in de latere fasen mee gewerkt gaat worden. De verschillende speelwijzen die mogelijk zijn in een kleurcombinatie zijn in feite de 'tactical hypotheses' in het model van Wilcox (zie blz.12): mogelijk te ondernemen akties, die echter nog wel op hun toepasselijkheid getest moeten worden. Deze akties vormen de bouwstenen van het speelplan.

Het aantal losers is in een kleurkontrakt van direkt belang voor het te volgen speelplan. In een SA-kontrakt gaat het meer om het aantal winners. Hoe dit precies in zijn werk gaat zal nu aan de orde komen, bij de bespreking van het concept 'plan'.

Het plan

Het eerste direkte produkt van de perceptie-fase is een kollektie speelwijzen, samen met hiermee samenhangende concepten. Omdat een speelwijze in feite een lokaal speelplan is, is dit de juiste plaats om wat dieper in te gaan op het concept 'plan'.

Een plan bestaat uit een reeks te ondernemen akties, die in de tijd geordend zijn. De primitiefste aktie is het spelen van een afzonderlijke kaart. Een niet-primitief plan - een plan dat bestaat uit meer dan een primitieve aktie - kan al of niet conditioneel zijn. In een conditioneel plan is het spelen van een kaart afhankelijk van het spel van de tegenstanders. Een voorbeeld van een conditioneel plan is de 'snit'. De snit is een benaming voor een klasse speelwijzen, die gemeen hebben dat extra slagen gemaakt kunnen worden vanwege het feit dat de ene speler eerder moet bijspelen dan de ander. In de positie:

```
        A V
H B           x x
        x x
```

speelt Zuid een van zijn kleine kaarten. Als West de boer speelt, wordt in Noord de vrouw gespeeld; speelt West de heer, dan wordt in Noord de aas gespeeld. Op deze manier worden er altijd twee slagen gehaald.

Het plan 'neem de snit' kan gerepresenteerd worden als een boom met twee takken:

```
          H - A
x <
          B - V
```

Voor een expert is het uitvoeren van zo'n conditioneel plan - het 'vertalen' van het plan in haar primitieve akties - een vanzelf-

sprekende zaak. Er bestaat een grote variatie van dit soort speelwijzen: squeezes, ingooien, coups, enz.

Belangrijk is, dat een concept als 'snit' een zelfstandige rol kan spelen in het redeneerproces. Deze rol wordt ontleend aan de effekten die het uitvoeren van een plan heeft. Een effekt kan gedefinieerd worden als de invloed die de uitvoering van het plan heeft op de bestaande omgevingsvariabelen.

In de omgeving kunnen de volgende variabelen onderscheiden worden:

- de op dat moment nog niet gespeelde kaarten
- degene die aan slag is
- het aantal gemaakte slagen door NZ en OW

Welke effekten de uitvoering van een plan heeft is afhankelijk van het zitsel van de gespeelde kleur, en de plaats van de honneurs in die kleur bij de tegenstanders. In de hierboven genoemde 'voorbeeld-snit' is slechts de plaats van de heer van belang; er zijn twee mogelijkheden.

1. West heeft de heer. Stel dat Zuid aan slag is; het uitvoeren van het plan 'neem de snit' heeft dan drie effekten: 1. na de eerste slag is Noord aan slag; 2. het aantal door NZ gemaakte slagen wordt met 1 verhoogd; en 3. de nieuwe omgeving telt nog slechts 4 kaarten.
2. Oost heeft de heer. De effekten zijn dan: 1. Na de eerste slag is Oost aan slag; 2. het aantal door OW gemaakte slagen wordt met 1 verhoogd; en 3. de nieuwe omgeving telt 4 kaarten.

In het tweede geval worden overigens assumpties gemaakt over het spel van de tegenstanders (nml. dat oost de heer zal nemen als hij die heeft). Hoewel in dit specifieke geval 99.9% van de bridgespelers de heer zal nemen, is dit punt wel van algemeen belang. Meer hierover echter later.

De effekten die een lokaal speelplan heeft zijn van het grootste belang bij het construeren van het overall speelplan, met name het aan slag komen van de tegenstanders. Als dit gebeurt zullen zij namelijk proberen om hun doel te realiseren, en de leider moet proberen te voorspellen welke kaart de hand aan slag zal naspelen, en welke effekten dit weer heeft op de nieuw ontstane situatie. Regelmatig komt het voor, dat de ene tegenstander hele vervelende dingen kan doen, terwijl de ander geheel ongevaarlijk is. Deze wetenschap kan de basis vormen voor het formuleren van eisen, waaraan het overall plan aan moet voldoen. Dit punt behoort echter al niet meer tot de perceptie-fase.

De herkenning van speelwijzen

Het associeren van bepaalde kaartpatronen met mogelijke speelwijzen is een kwestie van herkenning. De kaartpatronen die de

input vormen van het herkenningsproces worden altijd gevormd door de kaarten van de dummy in combinatie met die van de leider. Patronen kunnen bestaan uit kaarten van 1 kleur, maar ook uit kaarten van meerdere kleuren.

Het eenvoudigste, en meest voorkomende patroon is de snit-positie. Een snitpositie wordt gekenmerkt door een 'gat' in het eigen kaarten-bezit; de kaart onder het gat zou met behulp van de speelwijze 'snijden' (= het nemen van de snit) een slag kunnen worden. Dit patroon hoeft dus niet te bestaan uit alle kaarten van een bepaalde kleur; alleen de kaarten onder en/of boven het gat spelen een rol. Voorbeelden: AV-xx, AHB-xx, AV10-xx, HB-xx, Ax-VB.

Een snitpositie <u>suggereert</u> het nemen van een snit; of dit inderdaad een goede speelwijze is, is van meerdere faktoren afhankelijk. Op lokaal nivo (speelplan binnen een kleur) is het totaal aantal kaarten in een kleur van belang, en de 'hoogte' van het gat. Een voorbeeld van het eerste: AHBxxx-xxxx wordt wel degelijk herkend als een snitpositie; vanwege het aantal kaarten erbij (in totaal 10) wordt de speelwijze 'snijden' *normaliter* niet als goed beoordeeld. Een voorbeeld van het tweede: bij AHV10-xxxx wordt de 'snit over de boer' wel herkend, maar deze speelwijze is, a priori, minder kansrijk dan de speelwijze 'speel de aas, dan de heer, dan de vrouw'.

Het zou te ver voeren om alle kennis die experts hebben met betrekking tot snitposities hier op te sommen. Belangrijk is dat zij uit de positie mogelijke speelwijzen binnen die kleur kunnen herkennen. Elk van deze speelwijzen heeft een kans van slagen (nml. de kans van voorkomen van het zitsel waarbij de speelwijze slaagt); wat verder met deze kennis gedaan wordt is in de perceptie-fase verder niet aan de orde.

<u>Andere patronen</u>

Het is niet alleen <u>binnen</u> een kleur, dat speelwijzen herkend kunnen worden. Ook de situaties, waarin complexere speelwijzen als squeezes, ingooien, cross-ruffs, etc. toepasbaar zijn kunnen herkend worden; de patronen zijn echter veel ingewikkelder. Een typisch voorbeeld van het herkennen van de juiste speelwijze kan gevonden in een van de spellen die Charness (1979) gebruikte in zijn onderzoek naar patroonherkenning in bridge:

♠ V B 8 5 2
♥ A V 4
♦ A 10 6
♣ 7 5

♠ A H 9 6 3
♥ H 8 2
♦ H B 4
♣ A V

Kontrakt: 6♠; uitkomst ♥B

Dit probleem werd aan spelers van verschillende speelsterkten voorgelegd. Het volledige protokol van een expert was als volgt (tussen haakjes de verbruikte tijd):

The contract is six spades (3) uh the lead is the jack of hearts. (6) Uh. Well you're a hundred percent for the contract unless the opening heart is ruffed. (10) All you have to do is play low on the heart in dummy, (14) the king of hearts in your hand, draw sufficient rounds of trump, (20) play all the hearts and play ace and queen of clubs and whoever leads it either has to lead a diamond (25) which you can duck to the opposite hand and make sure of picking up the queen of diamonds and giving yourself twelve tricks (29) or they have to give you a ruff and a sluff in which case you sluff a diamond from one hand and ruff in the other and then ruff out the third losing diamond and automatically make twelve tricks (38).

Uit het protokol blijkt, dat de expert al na zes sekonden de juiste oplossing gevonden had. Het lijkt er sterk op dat het speelplan van de expert, uiteengezet in sekonde 10-38, in abstraktie gesuggereerd werd door de omgeving van de kaarten. Voor het toepasbaar zijn van deze speelwijze ('strip and endplay' genaamd) moet de omgeving aan de volgende eisen voldoen:

- er moeten een of meerdere kleuren zijn, waarbij de kans op het vermijden van losers in die kleur groter is als de kleur wordt aangespeeld door een van de tegenstanders;
- er moet een kleur zijn waarmee de tegenstanders aan slag kunnen worden gebracht (mogelijk dezelfde kleur als bovengenoemde);
- met betrekking tot de overige kleuren moet een van de volgende twee zaken het geval zijn:
    - de kaarten van de tegenstanders moeten elimineerbaar zijn, zodat zij die kleuren niet meer kunnen naspelen;
    - de eigen kaarten moeten elimineerbaar zijn EN in beide handen moet nog troef aanwezig zijn.

Deze omschrijving is op de gegeven omgeving van toepassing, zodat de 'strip and endplay' inderdaad als mogelijke speelwijze gesuggereerd wordt. Het enige verschil met het herkennen van snitposities is de complexiteit van het patroon, en de complexiteit van het gesuggereerde plan. Het plan 'strip and endplay' bestaat zelf weer uit een aantal deelplannen, die allemaal moeten worden uitgevoerd. De volgorde van uitvoering van deze deelplannen is belangrijk; welke volgorde gekozen wordt is weer van een aantal faktoren afhankelijk. Dat doet hier echter niet terzake.

Recapitulatie

De eerste stap in het construeren van een speelplan is het inventariseren van het eigen bezit. Hiertoe wordt de omgeving

georganiseerd in bekende kaart-configuraties. Van deze configuraties is veel bekend, waaronder een aantal 'statische' eigenschappen, zoals het aantal winners en losers. het aantal mogelijk te halen slagen, en of een tegenstander 'gevaarlijk' kan zijn met betrekking tot die configuratie. Ook bekend zijn een aantal mogelijk te ondernemen akties, en de effekten van deze akties bij verschillende mogelijke verdelingen van de vijandelijke kaarten.
De in deze fase gegenereerde kennis is op twee punten van belang voor de ontwikkeling van het uiteindelijke speelplan. Ten eerste wordt in de vorm van de mogelijk te ondernemen akties het materiaal geleverd, de bouwstenen waar het speelplan uit opgebouwd moet worden. Ten tweede kunnen aan de hand van deze kennis constraints geformuleerd worden aangaande de vorm van het speelplan.

Fase 2: r.6-7

Alle drie de spelers ontdekken hier de mogelijkheden die de ruiten kleur biedt (E.L.: 'ik kan op de ruitens spelen' (r.7); M.V.: 'ik kan misschien een ruiten ontwikkelen' (r.13); B.M.: 'ik zal in ieder geval iets met die ruiten moeten doen' (r.6).).

Hoe moeten deze opmerkingen geplaatst worden? Nadat in de perceptie-fase de akties ontdekt zijn die in de bestaande omgeving uitgevoerd kunnen worden, moeten nu die akties gekozen worden, die bijdragen aan de realisering van het doel: het maken van 10 van de 13 slagen met harten als troef. Nodig is dus, dat de lokale effekten die een lokaal speelplan kan hebben, overeen komen met de eisen die aan de effekten van het globale speelplan gesteld zijn. De keuze van een lokaal plan verloopt via taak-decompositie.

Het construeren van een speelplan is een ontwerptaak. Het doel van de taak is het ontwerpen van een transformatie-proces, dat het ene systeem (een omgeving van 52 kaarten) omzet in een ander (nml. 13 slagen van 4 kaarten elk). Het proces moet voldoen aan een aantal eisen, de 'ontwerp constraints'. Ten eerste betreffen de constraints natuurlijk de vorm van het plan: het plan mag uit niet meer dan 26 afzonderlijke speel-instrukties bestaan. De inhoud (d.w.z. de volgorde) is in principe niet aan beperkingen gebonden; indirekt wordt de inhoud echter beperkt door de eisen die er gesteld worden aan de output van het transformatie-proces. Het is niet mogelijk om de preciese volgorde van de operatoren direkt af te leiden uit de ontwerp-eisen. Dit 'vertalen' van de ontwerp-eisen naar een precies ontwerp verloopt in stappen.

Allereerst moeten algemene, abstrakte methoden gekozen worden, wier resultaat, als zij uitgevoerd worden, voldoet aan de ontwerp-eisen. Vervolgens moet gekeken worden, of deze methoden in de huidige omgeving wel toepasbaar zijn. Degene, die toepasbaar zijn, moeten tenslotte onderzocht worden op de exakte effekten, die zij hebben als zij in deze omgeving worden toegepast.

Spelers hebben dus kennis nodig over de verschillende manieren, waarop een bepaald doel gerealiseerd kan worden. De realisering van een doel hangt echter ook af van de huidige omgeving (als iemand b.v. naar Utrecht Centraal Station wil, is zijn keuze van transport-methode afhankelijk van de plaats waar hij zich bevindt: is hij op Hoog Catharijne dan zal hij gaan lopen, maar is hij in New York dan is wellicht het vliegtuig een praktischer methode om in de buurt te komen). Voor het kiezen van een speelwijze die een bepaald doel realiseert, is het daarom noodzakelijk dat de huidige omgeving beschreven kan worden in termen, die uitdrukking geven aan de afstand tussen deze omgeving en de doelsituatie.

In bridge wordt de afstand tussen de begin- en de eindsituatie gemeten in slagen. Echter, in de beginsituatie worden niet het aantal reeds gemaakte slagen geteld (dat zijn er namelijk altijd 0), maar het aantal slagen dat men redelijkerwijs verwacht te maken. 'Redelijkerwijs' is natuurlijk een gevaarlijk begrip, en het is afhankelijk van de sterkte van de speler. Een maat voor het aantal verwachte slagen zijn de winners en losers, die al in de vorige paragraaf genoemd zijn.

Strikt genomen is een winner een kaart, die door zijn rang noodzakelijkerwijs een slag moet halen. Een loser is een slag, die aan de tegenpartij afgestaan moet worden. De kleur: A H x - V x x telt b.v. 3 winners en geen losers; de kleur: A H x - x x x telt 2 winners en 1 loser (althans, in een troef kontrakt). Bij goede spelers wordt het begrip 'winner' echter uitgebreid met 'een kaart die 'zeer dikwijls' door het toepassen van een bepaalde speelwijze een slag zal maken'. Een voorbeeld:

De positie: A H V x x - x x x x telt strikt genomen maar drie winners. Door het toepassen van een eenvoudige speelwijze (nml. het spelen van achtereenvolgens de aas, de heer, en de vrouw) worden de kleine kaarten 'hoog': er zijn geen vijandelijke kaarten meer die hoger zijn. Deze positie telt dus 5 winners, behalve in het geval dat de 4 overige kaarten in de kleur 4-0 verdeeld zijn bij de tegenstanders, dan telt hij 4 winners en 1 loser. Deze verdeling komt echter maar zo weinig voor, dat als 'default' 5 winners geteld worden. Nog een voorbeeld:

In het voorbeeldspel komt in schoppen de positie: H x - A x x voor. Deze positie telt strikt genomen 2 winners en 1 loser. Geen van de 3 proefpersonen telt de positie echter als zodanig, omdat m.b.v. een eenvoudige speelwijze - het aftroeven van een schoppen in dummy - de loser geelimineerd kan worden. Beginners zullen zelfs met het vinden van deze speelwijze nog moeite hebben; bij goede spelers is het zo'n automatisme dat hij niet eens meer expliciet genoemd wordt.

Door het tellen van winners en losers is het mogelijk om een eerste schatting van de potentie van het kaartenbezit te geven. Op basis van het verschil van deze schatting en het aantal in het kontrakt gespecificeerde aantal te halen slagen kunnen de eerste constraints geformuleerd worden, waaraan het speelplan moet

voldoen.

Onder 'constraints' worden alle faktoren verstaan, die op een of andere manier invloed uitoefenen op de inhoud van het speelplan. Deze constraints kunnen op verschillende nivo's werkzaam zijn; zo kunnen constraints geformuleerd worden die betrekking hebben op het globale effekt van het speelplan, maar ook die betrekking hebben op de volgorde van de elementen van het plan. Constraints kunnen worden afgeleid uit 1. het te bereiken doel; en 2. de omgeving. Uiteraard worden de constraints tijdens het oplossen steeds meer verfijnd.

Het losers tellen is een eerste stap in het verfijnen van de constraints met betrekking tot het effekt van het plan. In het kontrakt wordt het globale doel gesteld: hoeveel slagen er gehaald moeten worden. Het resultaat van het tellen van winners en losers is het aantal slagen, dat 'zonder moeite' gehaald kan worden, omdat de speelwijzen in het plan niet afhankelijk zijn van de kaarten van de tegenstanders: het reflekteert het aantal slagen, dat 'uit nature' in de hand aanwezig is. Het verschil tussen deze twee aantallen bepaalt in eerste instantie de akties, die in het plan ondernomen moeten worden.

Het 'winner-verschil' en het 'loser-verschil' hebben echter niet dezelfde invloed op de te kiezen akties. Een tekort aan winners heeft altijd als gevolg, dat er in het plan voor gezorgd moet worden, dat er voldoende extra winners gecreeerd worden. Een teveel aan losers kan verschillende gevolgen hebben, afhankelijk van de 'direktheid' ervan, en van het feit of men tijdens het spel 'van slag af' moet (d.w.z. de tegenstander komt aan slag). Deze laatste twee zaken zijn in de perceptie-fase al bekend geworden.

Resumerend: de hand wordt beschreven in termen die de slag-potentie van de hand reflekteren. Op basis van het verschil tussen deze beschrijving en het kontrakt worden akties voorgesteld, die zorgen voor de overbrugging van dit verschil.

Deze akties worden in vrij abstrakte termen beschreven, b.v. 'werk losers weg', of 'creeer x extra winners'. Enkele voorbeelden (met voor de pijl de beschrijving van de omgeving, en achter de pijl de te ondernemen aktie):

te veel losers, te weinig winners	----->	werk losers weg
genoeg winners, hoeft niet van slag,	----->	neem slagen mee
genoeg winners, moet van slag, te veel direkte losers	----->	werk direkte losers weg, ga niet van slag

enzovoorts. Zoals uit het laatste voorbeeld blijkt, kunnen

effekten niet alleen geeist, maar ook verboden worden. Dit wil dus zeggen, dat uit de 'slag-potentie-beschrijving' twee soorten constraints geformuleerd kunnen worden: één betreffende het globale effekt van het plan, en één betreffende het effekt van de in het plan gebruikte elementen.

Deze constraints bepalen vervolgens welke lokale speelwijzen in aanmerking komen om in het speelplan te worden opgenomen. Let wel dat het in dit stadium inderdaad niet meer is dan 'in aanmerking komen'; in de volgende fase worden de hier gedane voorstellen verder uitgewerkt. Tot nog toe lijkt het allemaal vrij veel op wat er bij schaken gebeurt: spelers 'laten de stelling op zich inwerken' (de Groot 1946); zij ontdekken bekende patronen, die bepaalde akties suggereren. Zoals de schaker een verzwakte koningsvleugel 'ziet', en hierbij meteen aan het plan 'koningsaanval' denkt, zo ziet een bridger een mooie kleur, en denkt hierbij meteen aan het ontwikkelen van die kleur.

'Denken aan' is echter nog niet hetzelfde als uitvoeren. Voordat een gesuggereerde aktie goedgekeurd wordt als definitief plan moet er nog heel wat werk verricht worden. Hoe dit gebeurt zal ik in de volgende paragraaf beschrijven.

r.8-28: het uitwerken van het plan

E.L. en M.V. beginnen deze fase met bijna precies dezelfde zin: 'als ik ♠A neem, ♦H, schoppen voor de heer, ruiten voor de vrouw' (E.L. r.8); 'wat gebeurt er als ik ♠A pak, dan ruiten speel, komt schoppen terug, dan kan ik met ruiten naar mijn hand' (M.V. r.14). Wat zij beide doen is het uitvoeren van het plan; zij proberen door simulatie na te gaan of het voorgestelde plan wel het gewenste effekt heeft. Dit simuleren is natuurlijk vergelijkbaar met het zoeken bij schaken, maar er is een belangrijk verschil: een schaker heeft volledige informatie, en een bridger niet. Hoe een bridger dit probleem oplost zal uit de strategie van de spelers duidelijk worden.

Een eerste vereiste voor het kunnen vooruitzien, is dat het abstrakte plan, zoals dat in de vorige fase werd geformuleerd, kan worden 'vertaald' in primitieve akties. In de simulatie moet het plan kaart voor kaart uitgevoerd worden. Deze afzonderlijke kaart-speel-instrukties moeten voortkomen uit het abstrakte plan in kombinatie met de aktuele kaarten.

Wat spelers nodig hebben is dus kennis over hoe een plan uitgevoerd kan (of moet) worden. In dit specifieke geval bestaat daar niet veel variatie in. Het plan 'de ruiten vrijtroeven' bestaat uit 3 stappen: 1. zorgen dat de kaarten uit de korte hand weg zijn (deblokkeren, M.V. r.19); 2. oversteken naar de dummy; en 3. het troeven van een ruiten in de hand. Deze laatste stap moet netzolang herhaald worden tot de overgebleven ruiten vrij zijn. De eerste kaart die beide spelers dan ook spelen is ♦H; het spelen van ♦H is de eerste 'zet' die de eerste stap van het plan

realiseert. Het verklaren van de volgende kaart in de reeks - 'schoppen voor de heer' - vergt iets meer moeite. Daarvoor moeten we iets weten van de effekten die een gespeelde kaart kan hebben. Na het spelen van ♦H kunnen er heel wat dingen gebeuren, afhankelijk van de akties van de tegenstander en van de kaartverdeling:

1. OW duiken de slag; mocht dit gebeuren dan betekent dat dat Zuid nog steeds aan slag is, en mag bepalen wat er in de volgende slag gespeeld gaat worden. Tevens wordt het aantal slagen van NZ met 1 verhoogd.
2. Oost of West neemt de aas; dit heeft als gevolg dat het aantal slagen van OW 1 hoger wordt, maar belangrijker, dat Oost of West aan slag is, en iets terug mag spelen. Wie van de twee de slag wint is onbekend, maar met beide mogelijkheden moet rekening gehouden worden. Wat zij naspelen hangt af van:
   a. welke kaarten hij in zijn handen heeft; dit betekent natuurlijk niet alleen, dat hij niet een kaart na kan spelen die hij niet heeft, maar ook dat de kaarten die hij ziet zijn beslissing over het te voeren tegenspel beinvloeden, net zoals dat bij de leider het geval is.
   b. de sterkte van de tegenstanders; goede spelers spelen doorgaans beter tegen dan slechte. Dit betekent, dat als een bepaald tegenspel vervelend is voor de leider, hij er bij goede tegenstanders rekening mee moet houden dat dat tegenspel ook gevonden wordt. Dit is geen triviale constatering: in tegenstelling tot bij schaken weet de tegenstander aan slag niet precies, wat de hand is van de leider. Vaak is het juiste tegenspel heel moeilijk te vinden, zelfs voor zeer sterke spelers.

Voor het simuleren zijn er dus drie belangrijke variabelen te onderscheiden: hoe de kaartverdeling is bij de tegenstanders, wie er aan slag komt, en welke kaart deze naspeelt. Overigens is de tweede variabele direkt afhankelijk van de eerste, zodat er strikt genomen maar twee variabelen zijn; hun rol is echter verschillend, en het praktisch is om ze gescheiden te houden ('West is aan slag' is een aspekt van een situatie, 'West heeft ♦A' is een aspekt van een omgeving, zie blz.73).

Het is in principe natuurlijk mogelijk om bij het zoeken aan die drie variabelen alle waarden die zij kunnen hebben toe te kennen. Het resultaat zou dan een 'brute-force' zoekproces zijn, waarbij alle varianten onderzocht moeten worden. Bij schaken is dit al een vrij uitgebreid karwei, maar schaken heeft maar één variabele, nml. het tegenspel. Bij twee of bij drie variabelen is dit praktisch onmogelijk. Berliner kwam dit probleem tegen in zijn backgammon onderzoek (1975). Backgammon is een spel met twee variabelen: bij het voorspellen van de zet van de tegenstander moet ook nog rekening gehouden worden met de dobbelstenen. Er zijn in totaal 21 mogelijke worpen; elk van deze worpen kan op ca. 40 manieren gespeeld worden (in partijen ca. 17 manieren). Samen betekent dit een boom met een vertakkingsgraad van meer dan 800

(in partijen 357), nog hoger dan die van Go. Zo'n boom is natuurlijk onmogelijk te doorzoeken. Alternatieven werden genoemd door Levy (1981) en Berliner zelf. Levy stelt voor om in plaats van alle 21 worpen slechts een representatieve subset te onderzoeken, zodat het aantal van 21 wordt terugebracht naar b.v. 3 (nog altijd een niet onaanzienlijke boom overigens). Berliner laat het zoeken helemaal los; hij kiest een op kennis gebaseerde benadering die in een vergelijkbare vorm ook voor het spelen van eindspelen in schaak wordt toegepast. De classifikatie van posities speelt hierin een grote rol.

Bridgers lijken het grote vertakkingsprobleem hanteerbaar te maken door het aannemen van defaults. De bovengenoemde variabelen in het zoeken kunnen verschillende waarden aannemen, maar het is niet zo, dat elk van die waarden een even belangrijke rol speelt in het zoek proces. Omdat een 3-2 zitsel a priori een veel grotere kans van optreden heeft dan b.v. een 5-0 zitsel, is het weinig zinvol om in het zoekproces even veel tijd aan de laatste te besteden als aan de eerste. Het is, met andere woorden, praktisch om uit te gaan van de assumptie van een 3-2 zitsel, omdat dit de situatie is die je het vaakst zal tegenkomen. Zo mogelijk kun je deze assumptie later weer intrekken, om zo ook de situaties die er bestaan bij een 4-1 zitsel te onderzoeken.

Dergelijke defaultwaarden bestaan er ook met betrekking tot het tegenspel. Wat als default wordt aangenomen is echter sterk afhankelijk van de zichtbare dummy, de eventuele informatie uit het biedverloop, en de sterkte van de tegenstanders. 'Standaard' tegenspel is b.v. naar de zwakte toespelen, de kleur van partner terugspelen, troef inspelen bij een singleton in dummy, enz.

In de analyse van het spel neemt E.L. aan, dat degene die met ♦A aan slag zal komen, schoppen zal vervolgen. In dat geval is zijn volgende zet eenvoudig: hij moet ♠H nemen. Vervolgens wordt het speelplan weer aangewend om de volgende zet te genereren: ruiten naar de vrouw realiseert het voltooien van de eerste stap in de uitvoering van het plan.

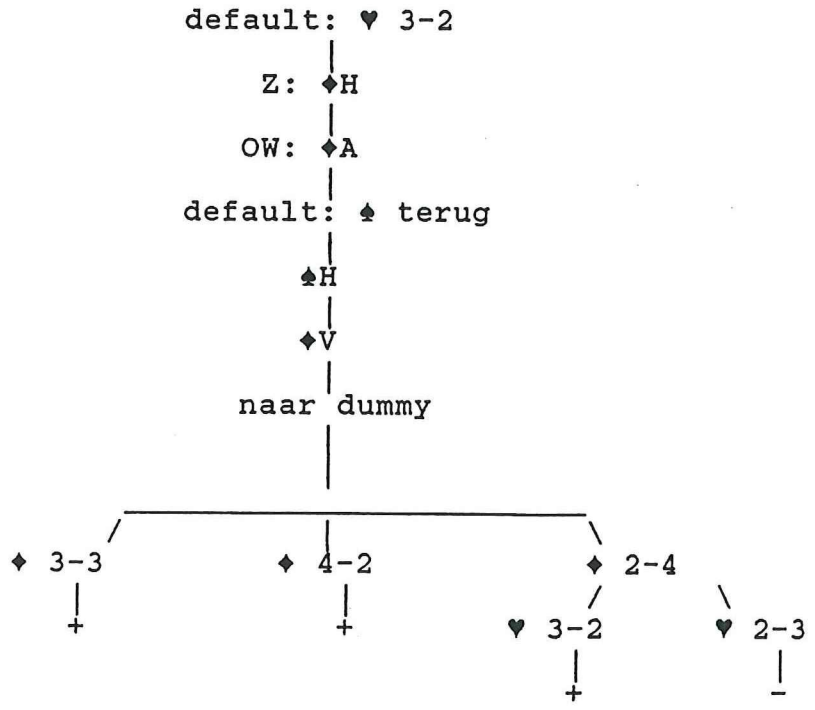
De tweede stap van het plan is het oversteken naar de dummy, en de derde het troeven van een ruiten in de hand. Hoe E.L. de tweede stap denkt uit te voeren wordt uit het protokol niet duidelijk; bij de uitvoering van de derde stap ziet hij problemen opduiken (r.11). E.L. refereert aan de situatie, waarbij west zijn ruiten introever kan overtroeven. Hoe arriveert hij bij deze specifieke situatie?

Bij het uitvoeren van de derde stap wordt de distributie van de ruiten van de tegenstanders van belang. Deze variabele heeft als drie belangrijkste waarden: 3-3, 4-2, en 2-4. De uitvoering van het plan in het eerste geval is geen probleem: als zuid troeft moeten oost en west bekennen; de ruiten zijn dan vrij, en het plan is geslaagd. Het tweede geval is iets lastiger, maar nog steeds niet echt problematisch: oost zal op de ruiten niet meer bekennen, en moet dus een andere kleur bijspelen. Twee belangrijke soorten

tegenspel zijn in zo'n situatie 'hoog voortroeven' en 'de kleur afgooien die in dummy afgetroefd gaat worden', waarvan de eerste de gevaarlijkste is. In dit geval is zuids troefkleur zodanig van samenstelling, dat deze speelwijze geen schade kan berokkenen, dus ook bij deze distributie zal het plan slagen.

Het derde geval is echter wel vervelend. Als west maar twee ruiten heeft, en zuid troeft af, dan kan west overtroeven. In dit geval is van belang hoeveel troeven west heeft. Omdat bij de aanvang van het spel 1 hartenloser werd geteld, betekent dit, dat op zich een overtroever niet erg is, als er daarna maar geen extra troefslagen verloren worden. Dit zal het geval zijn als west van oorsprong een driekaart harten heeft gehad; hij heeft dan ingetroefd 'van een vaste troefslag'. B.M. noemt deze distributie (r.11/12). Als west echter in het bezit van twee troeven, dan bestaat het gevaar dat er nog een troefslag verloren zal gaan worden; het kontrakt is dan down.

De analyse van het spel tot nog toe kan gekarakteriseerd worden door de volgende boom (zie fig.5.1):



figuur 5.1: zoekboom van E.L.

Na deze constatering herhaalt E.L. deze analyse in r.12-17. Hierbij onderkent hij het alternatief voor het default tegenspel van oost (klaver in plaats van schoppen na) mocht deze aan slag komen, maar verschuift dit probleem naar een later stadium. Op dit moment is het in de analyse van belang om te onderzoeken of het plan ook kan slagen bij het probleem-zitsel; het introduceren van nieuwe problemen voor dit is opgelost zou het zoekproces in hoge mate compliceren.

In r.17 noemt hij de entree naar dummy die eerder (r.10/11) in het midden gelaten werd. Het kiezen van de schoppen aftroever als entree zou echter tot dezelfde situatie leiden die in de bovenstaande boom tot problemen zou leiden. In r.18 echter vindt E.L. de oplossing voor het probleem: door eerst ♥H te spelen voordat hij met ♥A naar dummy gaat, worden 4 troeven van de tegenstanders verwijderd (als deze tenminste 3-2 hebben gezeten). De oorzaak van de negatieve evaluatie in fig.5.1 is dan weggenomen: er kan nooit meer dan één troefslag verloren gaan, eenvoudigweg omdat de tegenstanders nog maar één troef hebben. De vraag is hoe deze speelwijze gevonden is.

Een eerste vereiste lijkt het onderkennen van de oorzaak van het mogelijke falen van het plan, nml. het introeven van een ruiten in de hand op het moment dat west nog schadelijk (d.w.z. niet van een vaste troefslag) kon overtroeven. Omdat gerekend wordt op 1 troefloser staat 'een vaste troefslag' gelijk met 'de derde troef van een oorspronkelijk bezit van drie'. Laten we de situatie waarin het plan faalt de 'faalconditie' noemen; de faalconditie van het plan 'ruiten vrijtroeven' bestaat dan uit de volgende componenten:

- de ruiten zitten 4-2 met de doubleton bij west;
- zuid troeft een ruiten af; en
- west troeft over van een oorspronkelijk bezit van 1 of 2 kleintjes.

Om het plan te laten slagen moeten 1 of meerdere componenten ongedaan gemaakt worden. De eerste component kan niet veranderd worden; het is immers een omgevingsvariabele. De tweede component kan ook niet veranderd worden; het is een essentieel onderdeel van het plan. Blijft over de derde component.

Om te zorgen dat een tegenstander niet kan aftroeven op een moment dat dit ongelegen komt, is er een standaard methode voor handen, nml. een speelwijze genaamd 'troef trekken'. Troef trekken bestaat uit het net zo lang spelen van hoge troeven tot de tegenstanders geen hinderlijke troeven meer hebben. Deze speelwijze zou hier uitstekend van toepassing zijn: als de leider 2 rondjes troef kan trekken (aangenomen dat deze 3-2 zitten), dan zou dat de eliminatie van de derde component betekenen, en dus het slagen van het plan. In het bestaande plan moet dus een zodanige wijziging aangebracht worden, dat het troeftrekken plaats vindt vóór het

aftroeven van de ruiten, maar dat de rest van het plan gehandhaafd kan blijven. Dit kan gerealiseerd worden door voor het oversteken met ♥A (met als effekt: 1 rondje troeftrekken) eerst uit de hand ♥H te spelen. Deze volgorde zorgt ervoor dat er twee rondjes troef getrokken kunnen worden, en dat de dummy aan slag is aan het eind van het troef trekken.

Waarschijnlijk speelt herkenning ook een niet onbelangrijke rol hier. Vaak is het mogelijk om een bepaalde speelwijze geheel 'logisch' af te leiden, ongeveer op dezelfde wijze als in de bovenstaande analyse gebeurde. Zeker in de analyse na afloop van het spel zijn kreten als 'dat is eigenlijk heel logisch' of 'dat had ik gemakkelijk kunnen vinden' schering en inslag. Aan tafel is het vaak een kwestie van 'zien'. Net zoals de expert in het spel van Charness (zie blz.45) de oplossing van het probleem in één oogopslag vond door herkenning, zo verloopt het oplossen van lokale problemen (zoals het vinden van het troeftrekken in combinatie met het naar de dummy oversteken) langs dezelfde weg.

Het spelen van de aas en de heer in een troefpositie van A x x tegenover H B x x x is ongebruikelijk, maar niet onbekend. Dat herkenning van deze figuur een rol speelt bij het vinden van de oplossing (d.w.z. dat het vinden van de speelwijze een heuristische stap is en geen logische) wordt aannemelijk als we kijken naar het protokol van M.V. Nadat hij gearriveerd is bij situatie1 vindt hij direkt de oplossing, zonder de tussenliggende stappen te noemen die in de andere twee protokollen na situatie1 naar voren komen.

Dat herkenning in het denken van M.V. een grote rol speelt wordt op meerdere plaatsen in zijn protokollen duidelijk. Een voorbeeld kan gevonden worden in probleem 1 r.9: "het lijkt me niet in ieder geval niet een handje om een soort safety-play in harten te nemen; ik heb bovendien ♥9 niet". Een safety-play is een speelwijze in een kleur, die ervoor zorgt dat er in die kleur minder dan een zeker aantal slagen verloren worden, onder opoffering van de kansen op meer slagen. Een standaard safety-play heeft betrekking op de positie A 9 x - H B x x x : door eerst de heer te spelen en dan klein naar de 9, kan altijd vermeden worden dat er meer dan 1 slag in die kleur verloren wordt (d.w.z. zolang ze niet 5-0 zitten). De kans echter dat er nu in die kleur 5 slagen gehaald worden (18% bij de default speelwijze: snit nemen) is nu praktisch nihil.

Interessant is nu dat M.V. bij het zien van de positie A 7 6 - H B 8 4 3 onmiddelijk aan deze safety-play denkt, terwijl er in het hele spel niets is dat zo'n safety-play toepasselijk doet zijn. Nog interessanter is, dat de match niet volledig is: A 7 6 - H B 8 4 3 is helemaal geen standaard safety-play situatie! Beide 'ontoepasselijkeden' worden onmiddelijk door de speler opgemerkt (r.9), maar dat doet niets af aan het feit, dat een herkenningsproces wel degelijk heeft plaats gevonden.

Dit soort sterke associatieve banden tussen posities en speelwijzen komen vaak voor bij bridge. Een ander voorbeeld is de positie A V 9 - x x x, die bij goede spelers onmiddelijk de speelwijze 'zijkleuren elimineren, dan een kleintje naar de 9' suggereert. Dit is de enige positie waarbij deze speelwijze waterdicht en technisch correkt is. In de positie A V 8 - x x x is deze speelwijze ook mogelijk, maar deze zal doorgaans alleen tot sukses leiden als de tegenstander vóór de A V 8 een fout maakt. In feite is het enige dat deze positie met de genoemde speelwijze de gelijkenis met de 'ideale' positie. Het herkennen van posities is dan niet alleen het vergelijken van de eigenschappen van een positie met een abstrakte klasse, maar ook het vergelijken van de eigenschappen van de positie met die van het 'ideale' lid van zo'n klasse. Op deze manier is het behoren tot een klasse geen ja-of-nee kwestie, maar een continuüm.

Nu er een oplossing voor het probleem gevonden is, moet het totale speelplan, zoals dat na het inpassen van de reparatie er uitziet, geevalueerd worden. De eindevaluatie van een speelplan vindt plaats door na te gaan bij welke zitsels het plan slaagt, en wat de kans is dat deze zitsels optreden. De kans van optreden van bepaalde zitsels, zoals 3-2, 3-3, 4-2, etc. is bij alle experts bekend, zoadt het slagingspercentage van een speelplan verkregen kan worden door een eenvoudig rekensommetje. Overigens wordt dit rekensommetje in de praktijk zelden gemaakt. Experts weten wanneer een kans 'acceptabel' is en wanner niet. Alleen als er 2 speelplannen met elkaar vergeleken moeten worden die elkaar 'op het gevoel' weinig ontlopen, worden expliciete berekeningen gebruikt.

M.V. vat samen wanneer het plan slaagt: "en dan ja, als de ruitens 4-2 en de hartens 3-2 zitten heb ik het altijd gehaald" (r.16). De kans hierop, die hij niet expliciet uitrekent, is (.48+.36)*.68= 57.12%. Dit is op zich niet zo erg veel, maar de evaluatie van een speelplan staat nooit op zichzelf: het plan moet altijd afgewogen worden tegen de aanwezige alternatieven. In dit geval bestaat het enige alternatief uit het nemen van de harten snit, een speelwijze die sukses heeft als oost V x of V x x heeft, een zitsel met slechts .50 * .36 = 18% kans. Dit is beduidend minder dan de ruim 57% van het andere speelplan.

Een tweede aspekt van het evalueren van een speelplan is het onderzoeken van alternatieven voor de defaults, die bij het plannen zijn aangenomen. Ook hier echter lijkt de tevredenheid met de reeds gevonden oplossing een rol te spelen. M.V., net als B.M. toen deze de situatie die ontstond nadat west de ingetroefde ruiten dreigde over te troeven, besteedt aan mogelijke alternatieven geen aandacht meer: hij stopt zijn analyse met de constatering dat het kontrakt gemaakt is met de ruiten 4-2 en de harten 3-2. E.L. houdt zich echter op verschillende plaatsen nog bezig met het onderzoeken van de alternatieven.

De eerste beschouwt hij in r.20: de situatie waar oost met ♦A aan slag komt en klaver kan naspelen. Deze situatie was hij al eerder tegengekomen (r.13), maar toen werd de analyse vanwege de te grote gedetaillerdheid voorlopig op de lange baan geschoven. Nu blijkt, dat het speelplan bij het default tegenspel van schoppen terug slaagt, kunnen de alternatieven van dit tegenspel bekeken worden. Dit doet hij in r.21-23. Bij het klaver naspel van oost zou het oorspronkelijke plan nog steeds uitgevoerd kunnen wrden, alleen E.L. lijkt dit niet te zien (nml. ♣A, ♦V, ♥H, ♥A, ♦ troef, ♠ troef, vrije ruiten). De reden waarom hij deze speelwijze niet ziet, kan het feit zijn dat na het klaver naspel van oost ♠H nog in de dummy ligt, terwijl die in de vorige varianten weggespeeld was.

Het tweede alternatief voor een default is de situatie waarin de harten niet 3-2 maar 4-1 zitten. Ook hier besteed M.V. geen aandacht aan, en E.L. ook niet in zijn oorspronkelijke analyse. Op het moment dat in het spel de eerste horde genomen is (oost heeft geen klaver nagespeeld), en hij in situatie1 is aanbeland (r.36), checkt E.L. nogmaals het plan, waarbij ook het 4-1 zitsel van de harten aan de orde komt (r.45-49). Voor dit probleem vindt hij in eerste instantie geen oplossing (r.46), denkt dan dat het plan toch slaagt als maar de ruiten 3-3 zitten (r.48), maar komt hier dan later weer op terug (r.49). Ondanks deze foute beoordeling (als de ruiten 3-3 zitten en de harten 4-1 slaagt het plan wèl) komt hij tot de conclusie dat een aanpassing van het plan, zodat het ook bij een 4-1 zitsel zou slagen, onmogelijk is: het plan blijft dan ook gehandhaafd.

Een derde situatie die mogelijk problemen kan opleveren onderkent E.L. in r.39, nml. als oost drie harten heeft en slechts twee schoppen. In dat geval zou het plan falen omdat oost de schoppen introever overtroeft, zodat er voor het vrijspelen van de ruiten een entree zou ontbreken. De kans hierop is erg klein, en bovendien weet E.L. voor deze situatie geen passende reparatie te vinden. Hij merkt wel op, dat zelfs in dat geval het plan nog steeds niet definitief gefaald heeft: als oost ♠H heeft is het spel nog steeds gemaakt.

Dit laatste punt raakt aan een belangrijk idee bij bridge: het 'behouden van kansen'. Een speelplan is gebaseerd op een bepaald zitsel. Als dit zitsel in de hand aanwezig is, dan slaagt het plan; zo niet dan faalt het. Echter, een plan is niet meer dan een <u>voorgenomen</u> volgorde van te spelen kaarten; de uitvoering van het plan kan tijdens het spel nog onderbroken worden, als blijkt dat het slaag-zitsel zich niet voordoet. In zo'n geval moet er een nieuw speelplan gemaakt worden, met weer nieuwe kansen. De kansen die nog overblijven als een plan dreigt te falen zijn een graadmeter voor de flexibiliteit van dit plan. Hoe flexibeler, hoe beter een plan is. Vaak moet tijdens het formuleren van het plan al rekening gehouden worden met het behouden van kansen. Het plan

moet zo ingericht worden, dat op het moment tijdens het spel dat er ontdekt wordt dat het plan faalt, er nog op een ander plan overgeschakeld kan worden. Hoe groter de kans van slagen van dit tweede plan, hoe beter het eerste plan was. Dit inrichten van het speelplan op zodanige manier, dat op zoveel mogelijk zitsels geanticipeerd kan worden, heet 'het combineren van kansen'. Voor het huidige spel is het belangrijk, dat als de gekozen speelwijze faalt, er nog altijd geprofiteerd kan worden van de kans dat oost ♠H heeft. Zowel E.L. ("dan moet die ♠H maar goed zitten", r.41), als B.M. ("maar ja, dan moet ♠H maar goed zitten", r.33) noemen deze mogelijkheid. Het kunnen combineren van kansen is een van de moeilijkste vaardigheden binnen het bridgespel. Het vereist: het kunnen onderkennen van veschillende speelplannen; het kunnen vertalen van die plannen in primitieve akties (afzonderlijke kaart-speelinstrukties); en tenslotte het kunnen combineren van deze akties tot een reeks. Voor elke aktie in die reeks moet gelden, dat die een onderdeel is van een uit te voeren deelplan, en tevens dat de effekten van die akties zodanig zijn dat deze de uitvoerbaarheid van mogelijke nog volgende deelplannen niet verstoren. Een voorbeeld kan dit alles misschien wat verduidelijken:

♠ A H x
♥ A H B 10 x
♦ A H
♣ x x x

N

Z

♠ V B x x x
♥ x x x
♦ x
♣ A V B 10

Het kontrakt is 7 SA (13 slagen), uitkomst ♦B.

Er zijn 10 topslagen, en er zijn twee mogelijke slagenbronnen voor de extra 3 slagen, nml. harten en klaveren. Als ♠H goed zit levert de snit deze 3 extra slagen; de kans hierop is 50%. Een alternatief plan is ♥A spelen en dan de hartensnit nemen; dit wint als west de vrouw heeft, of oost de vrouw sec: een kans van 52.8% Geen van beide is echter de beste speelwijze. Het plan waarin de meeste kansen gecombineerd worden, bestaat uit het spelen van ♥A en ♥H. Als de vrouw doubleton zit (27.2%) zijn er 13 slagen; is dat niet het geval, dan is er niets verloren, want dan kan nog steeds de klaver snit worden genomen (50%). Het plan 'eerst ♥AH

spelen, dan als de vrouw niet gevallen is, de klaver snit nemen' slaagt dus in 77.7% van alle mogelijke distributies.

Dit speelplan is het resultaat van het combineren van speelwijzen in beide kleuren. Het spelen van ♥AH is 'gratis': het laat de andere kansen intakt. Het nemen van de harten snit zou niet gratis zijn, want als oost de vrouw heeft ben je down.

Recapitulatie

In deze derde fase worden de abstrakte plannen die in de voorafgaande fase werden voorgesteld, uitgewerkt en onderzocht op hun toepasbaarheid. Dit gebeurt via simulatie: het te testen plan wordt 'in het hoofd' uitgevoerd; hierbij wordt rekening gehouden met verschillende distributies en verschillende soorten tegenspel. Als uitgangspunt wordt echter van beide variabelen defaultwaarden genomen om het zoekproces te struktureren; als de hoofdlijn vastligt kunnen dan eventueel ook nog de andere waarden worden nagegaan. Als tijdens het zoeken een distributie of tegenspel wordt tegengekomen waartegen het plan in de op dat moment bestaande vorm niet bestand is, dan wordt de oorzaak van dit falen onderzocht. Vervolgens wordt geprobeerd om het plan zodanig aan te passen, dat het ook in die gevallen slaagt. Dit repareren vindt alleen plaats als het noodzakelijk wordt bevonden; als de faalcondities een erg kleine kans van voorkomen hebben, dan wordt vaak genoegen genomen met het op dat moment bestaande plan. Als er concurerende plannen bestaan wordt er geprobeerd om deze te combineren tot één plan, waarin er gezorgd wordt dat het uitvoeren van het eerste deel van dit plan de mogelijkheid tot het uitvoeren van het eventuele tweede deel van het plan intakt laat.

Het gehele planningsproces verloopt dan via de volgende stappen (zie volgende bladzijde):

herkenning	het associeren van kaartconfiguraties met bepaalde speelwijzen;
selektie	het kiezen van een speelwijze op basis van het gestelde doel;
simulatie	het uitwerken van de voorgestelde speelwijzen bij verschillende distributies en tegenspel;
diagnose/ evaluatie	het vaststellen van de faal- en slaagcondities van het plan, en het vergelijken van deze met die van andere plannen;
reparatie	het zonodig aanpassen van het plan ter verkleining van de faalkansen ervan; en
selektie/ combinatie	het kiezen van een plan uit de mogelijke kandidaten, en het zo mogelijk combineren van deze laatsten tot één geheel.

HOOFDSTUK 6: HERKENNING NADER BESCHOUWD

In het vorige hoofdstuk heb ik drie protokollen besproken. Uit de protokollen was al een redelijk algemeen beeld van het oplossingsproces op te bouwen, maar ook nog niet veel meer dan dat. Het oplossingsgedrag op één spel kan ons wel een idee geven van de soorten processen die er een rol bij spelen, maar niet over de preciese aard van die processen.

Een voorbeeld daarvan is het herkennen van situaties en bijbehorende speelwijzen. Op verschillende plaatsen kwam dit proces in de protokollen naar voren, maar verschillende zaken blijven onduidelijk:

- of er restrikties zijn ten aanzien van de plaats in het oplossingsproces waarop het herkennen van speelwijzen kan voorkomen;
- wat er precies herkend wordt, d.w.z. wat zijn de termen die gebruikt worden om de te herkennen situatie mee te beschrijven; en
- hoe de speelwijzen, die met de herkende situatie geassocieerd worden, worden gebruikt bij het vormgeven van het speelplan; hiermee samen hangt ook weer de termen waarin de speelwijzen worden beschreven, en de mate van abstraktie.

Mogelijk inzicht in deze materie kan worden verkregen uit het bestuderen van leerboekjes. Er is een grote hoeveelheid boeken in de handel die als doel hebben beginnende en gevorderde bridgers meer van het spel bij te brengen. De stijl waarin dit gebeurt is erg interessant: de meeste suksesvolle leerboeken bestaan uit begeleide problemen. Dit houdt in dat de lezer een bepaald spel moet afspelen, of een tegenspel probleem moet oplossen; als de lezer zijn oplossing heeft bepaald kan hij die op de volgende bladzijde vergelijken met de oplossing van de auteurs van het boek. Uiteraard proberen de auteurs dan om de lezer duidelijk te maken, hoe zij in het vervolg die oplossing zelf kunnen vinden; zij leggen bijvoorbeeld uit welke denkstappen er genomen moeten worden, welke vragen er beantwoord moeten worden voor de oplossing duidelijk wordt en welke concepten in de oplossing van belang zijn. Bovendien proberen zij de principes te formuleren, die aan het vinden van de goede oplossing ten grondslag liggen.

Waarom kunnen dit soort boeken inzicht geven in het denken van experts? De boeken leggen een link van beginners-oplossen naar expert-oplossen: de boeken zijn geschreven door experts, die echter het hoe en waarom van hun technieken moeten beschrijven in termen, die beginners kunnen begrijpen. Uit de boeken wordt dan dus duidelijk:

- welke concepten experts gebruiken bij het oplossen;
- de 'oorsprong' van die concepten, en hun relatie met andere concepten;

- een gedeelte van de taakstruktuur: hoe de expert het probleem benadert, en in welke volgorde hij zijn vragen stelt;
- de aard van de 'principes': de heuristieken die de expert in staat stelt snel de oplossing te vinden, en die de beginner geacht wordt te beheersen na het lezen van het boek;
- de gebruikte 'logika': de niet uitgesproken denkstappen (de conclusies die 'logisch' voortkomen uit bepaalde feiten) geven een beeld van de aard van de kennis van de expert en de beginner, en van de processen die de ene vorm van kennis omzet in een andere.

Het te analyseren spel komt evenals het protokol-spel uit 'Bridge - spel en tegenspel' van Cees Sint en Ton Schipperheyn, een leerkursus voor beginners. De boekjes zijn geschreven in 'stap-voor-stap-stijl': de lezer wordt geacht de vragen te beantwoorden die de auteurs hen stellen, en zo wordt de lezer langzamerhand de juiste richting in geleid. De auteurs schrijven in hun inleiding:

> "Men heeft proeven gedaan met sterke denksporters, waaruit bleek dat hun kracht niet zozeer ligt in het zelf bedenken van fraaie oplossingen voor moeilijke problemen als wel in het herkennen van situaties die zij ooit eerder tegenkwamen.
> Een goede bridgespeler beschikt over een arsenaal van speelwijzen die hij ooit eens zelf heeft bedacht (meestal achteraf!), dan wel ergens heeft gezien of gelezen, en vooral dat laatste is belangrijk. Bij het uitspelen profiteert hij van deze kennis, waardoor hij zich de extra kansen verschaft die vaak nodig zijn om een moeilijk contract te maken" (blz.7).

Door het lezen van dit soort boeken kan de beginnende bridger dus een repertoir van speelwijzen opbouwen. Het leren herkennen is een eerste doel van het boek. Als het boek er in zou slagen om bridgers te leren om situaties te herkennen, dan moet bestudering van het boek ons wat kunnen leren over de aard van deze herkenning.

Hieronder volgt de complete tekst; het spel is het vijfde uit het hoofdstuk 'Hoe maak ik de tegenstander weerloos', waarin wordt uitgelegd hoe je een SA-kontrakt moet aanpakken (zie volgende bladzijde).

♠ A 8 3
♥ 8 7 3 2
♦ V B
♣ A V 10 2

N
?? W O ??
Z

♠ H 9 5
♥ A H
♦ A 7 4 2
♣ B 9 5 3

Langzamerhand kunt u wel raden wat het eindkontrakt is. West komt tegen 3 SA (goed geraden!) uit met ♥V. U moet de slag nemen met ♥H.

Hoe taxeert u de kansen?

U heeft goede papieren. Als de snit in klaveren goed zit bent u er al. En als die snit mis zit? Dan dreigen er problemen; oost zal immers weer harten spelen en als ♦H dan ook nog eens fout zit, maakt west wellicht nog drie harten slagen. Uit de spellen hiervoor heeft u echter geleerd dat eerst 'de gevaarlijke hand' moet worden aangevallen.

Is oost of west de gevaarlijke hand?

Gezien de uitkomst west; waarschijnlijk zit daar de lengte in harten.

Welke kleur moet u dus eerst spelen; klaveren of ruiten?

Ruiten; west mag eerst aan slag komen en zal harten vervolgen. Maar als oost vervolgens met ♠H aan slag komt zijn z'n hartens wellicht op. In slag twee speelt u schoppen naar tafel en neemt de snit in ruiten. West neemt ♦H en verzinkt in diep gepeins. Vervolgens legt hij ♠10 op tafel die oost overneemt met ♠B. U kunt nu duiken, maar oost gaat door met schoppen. Aan slag gekomen met ♣H neemt oost nog een aantal schoppenslagen mee. Oost-west maken ♦H, ♣H, en drie slagen in schoppen.

Wat is er fout gegaan?

De redenering op grond waarvan u eerst ruiten speelt en dan pas klaveren blijft correct. U heeft oost-west echter een kans gegeven een andere zwakke plek in uw spel te vinden door in schoppen over te steken om de ruiten snit te nemen.
U vindt het niet erg als die ruitensnit verkeerd zit. Maar

31. waarom speelt u dan niet in slag twee een kleine ruiten
32. uit de hand?
33. Er kan dan niets gebeuren. Op uw gemak komt u aan twee
34. schoppen-, twee harten-, twee ruiten-, en drie klaverslagen.

35. **EPILOOG**

36. Een spel met een dubbele bodem. Eerst komt u tot de conclusie
37. dat west gevaarlijk is en dat de ruitenkleur voorrang
38. heeft boven de klaverkleur. Het lijkt dan logisch om een
39. snit in ruiten te nemen. U komt dan echter op een andere
40. manier in gevaar. Als het even kan moet u alle zwakke
41. plekken beschermen.

(uit: Sint en Schipperheyn 1982, blz.23-24)

In eerste instantie zal ik me bij de analyse van het spel concentreren op de aard en volgorde van de inferentieprocessen. Aanwijzingen hierover kunnen komen uit de volgorde waarin de schrijvers de feiten presenteren, die over de hand bekend worden verondersteld te zijn.

In r.5-8 vindt al een groot gedeelte van het eigenlijke plannen plaats; de aktiviteiten in deze regels werden hierboven beschreven als het 'simulatie-proces'. Om r.5 te verklaren -'als de snit in klaveren goed zit bent u er al'- moet inderdaad gerefereerd worden aan de twee aan de simulatie voorafgaande processen herkenning en selektie. Van de lezer wordt namelijk verondersteld dat hij weet dat:

- hij 6 slagen aan top heeft, en geen 5 direkte losers;
- dat hij een 3 SA-kontrakt speelt, en dat de twee bovenstaande feiten leiden tot de strategie 'ontwikkel 3 winners';
- dat er twee kleuren zijn waarin mogelijkerwijs winners in te ontwikkelen zijn: klaveren en ruiten; en
- dat van deze twee de klaverkleur de hoogste verwachte 'opbrengst' heeft.

Het bepalen van de winner/loser telling, en het vaststellen van de mogelijkheden van de verschillende kleuren behoort tot de herkenning; het afleiden van de te volgen strategie, en de keuze van 'klaver spelen' en 'ruiten spelen' als mogelijke speelwijzen die passen in deze strategie, behoren tot het selektie-proces.

Na het selekteren van 'speel klaveren' wordt deze speelwijze d.m.v. simulatie aan een nader onderzoek onderworpen. Dit proces wordt beschreven in r.5-8. De default manier om de klaverkleur aan te spelen is het nemen van de snit op de heer over west. Met betrekking tot het nemen van een snit zijn er twee mogelijke distributies van belang: de heer zit 'goed' (de snit lukt), en de heer zit fout (de snit mislukt). In het simuleren van het

toepassen van de speelwijze 'neem de klaversnit' trekken de auteurs hun conclusies in de volgende volgorde:

> ♣H zit goed ---> leider heeft 9 slagen
> ♣H zit fout ---> oost komt aan slag ---> oost speelt harten ---> zuid komt aan slag.

In deze situatie moet de leider opnieuw bepalen wat te spelen. De winner/loser telling is nu veranderd: er zijn 8 topslagen, en OW hebben 1 slag. De te volgen strategie is echter niet veranderd, en omdat 'ruiten spelen' de enige speelwijze is die nog past binnen die strategie wordt deze geselekteerd. Het nieuwe aktieve plan, dat zuids zetten genereert is nu dus 'speel ruiten'. De default manier om dit te doen is weer het nemen van de snit, en dit bepaalt de volgende zetten. Nu volgt:

> ♦H zit goed ---> leider heeft 9 slagen
> ♦H zit fout ---> west komt aan slag ---> west speelt harten
> west heeft nog 3 harten over ---> west speelt 3 maal harten
> ---> OW hebben 5 slagen
> west heeft nog 2 harten over ---> west speelt 2 maal harten
> ---> OW hebben 4 slagen, de leider heeft er 9.

Het nu geformuleerde speelplan: 'neem de klaversnit, en daarna zonodig de ruitensnit' slaagt in drie van de vier relevante gevallen, nml. als ♠H bij west zit, òf als ♦H bij oost zit. In het overgebleven geval gaat de leider down.

Een plan dat bij een bepaalde distributie niet werkt moet gerepareerd worden, en hier komt weer een vorm van herkenning om de hoek kijken. Strikt genomen zijn er heel wat manieren om het bestaande speelplan te veranderen. Zo kan je het spel binnen één van de deelplannen 'speel klaver' en 'speel ruiten' veranderen: in plaats van de default manier 'neem de snit' kies je bijvoorbeeld een andere: 'speel eerst de aas'. Ook mogelijk is om het plan te laten voorafgaan door een ander plan: 'speel alle winners af'; nog een andere mogelijkheid is het veranderen van de volgorde van de twee deelplannen in het huidige speelplan. Uit de formulering van de auteurs (r.9-10) valt echter af te lezen, dat de lezer geacht wordt te herkennen, dat 'val eerst de gevaarlijke hand aan' het reparerende speelplan is in dit geval.

Net zoals uit de klaverconfiguratie A V 10 2 - B 9 5 3 herkend wordt dat er 4 slagen gemaakt kunnen worden als west de heer heeft door de speelwijze 'neem de snit' toe te passen, zo wordt ook herkend dat 'val eerst de gevaarlijke hand aan' een speelwijze is die in deze situatie in aanmerking komt. Het verschil is, dat in het eerste geval de beschrijving van het patroon plaats had in termen van afzonderlijke kaarten, maar dat er bij de beschrijving van de situatie in het laatste geval gebruik gemaakt wordt van ingewikkeldere concepten.

De kennis die gebruikt wordt bij het herkennen kan blijkbaar worden beschreven als regels van de vorm:

situatie ---> speelwijze

Zowel 'situatie' als 'speelwijze' kennen niet bijzonder veel beperkingen ten aanzien van de termen waarin zij worden beschreven. Een situatie is een verzameling van over de hand bekende feiten; een speelwijze is een verzameling akties en/of termen die naar een verzameling akties verwijzen. De regel die gebruikt wordt om de reparerende speelwijze te herkennen zou als volgt kunnen worden beschreven:

ALS er 2 kleuren gespeeld moeten worden EN
je moet in beide kleuren van slag EN
er is een gevaarlijke kleur EN
die kleur heeft minder stoppers dan
het aantal malen dat je van slag moet

DAN speel eerst de kleur waarin de mogelijke
entree van de gevaarlijke hand is gelegen.

De beschrijving van het **ALS**-gedeelte is van toepassing op de situatie die is ontstaan als in de simulatie blijkt, dat west aan slag kan komen met ♦H, en dat het kontrakt dan down is. De beschrijving is overigens ook van toepassing op 'de spellen hiervoor' die de auteurs in r.9 noemen.

In het **DAN**-gedeelte wordt gerefereerd aan een speelwijze, en aan het tijdstip waarop die speelwijze moet worden toegepast: het woordje 'eerst' heeft betrekking op de volgorde, waarin de twee deelplannen 'speel klaver' en 'speel ruiten' in het speelplan voorkomen. Omdat de regel uiteraard in meerdere situaties toepasbaar moet zijn, heeft het **DAN**-gedeelte een enigszins abstrakt karakter: 'de kleur waarin de mogelijke entree van de gevaarlijke hand is gelegen' moet om te kunnen fungeren in het speelplan eerst nog worden vertaald naar de huidige situatie.

In de regels 9-10 wordt de lezer er dus door de auteurs op attent gemaakt, dat bovengenoemde regel van toepassing is; in r.11-15 proberen de auteurs het vertalingsproces aan de lezer duidelijk te maken. Allereerst geven zij aan, wat de criteria zijn voor 'gevaarlijke hand'. In dit geval is 'gevaarlijke hand' equivalent met 'de hand met de lengte in de gevaarlijke kleur'. Via een andere heuristiek, nml. dat een verdediger tegen een SA-kontrakt in de regel altijd uitkomt van zijn langste kleur, en het feit dat west met harten uitkwam, en het feit dat harten de gevaarlijke kleur is, kan worden afgeleid dat west de gevaarlijke hand is.

De volgende stap is het vinden van de kleur waarin de entree van west is gelegen. Hiervoor is het weer van belang, dat de leider iets weet over de effekten die het spelen van een bepaalde kleur kan hebben, zoals in het vorige hoofdstuk al werd aangegeven. Bij de mogelijke effekten van het nemen van de klaver snit, zit het aan slag komen van oost. West kan onmogelijk aan slag komen; als hij de heer heeft lukt de snit immers. Het omgekeerde is het geval

in de ruitenkleur. Wests mogelijke entree is dus ♦H.

Dit invullen van het **DAN**-gedeelte van de heuristiek leidt tot de ontwerp-constraint 'speel eerst ruiten'. Omdat er ook nog een slag in de ruiten ontwikkeld moet worden (de reden dat er sowieso ruiten gespeeld moet worden) betekent dit, dat er twee mogelijke speelwijzen in de ruiten zijn die dit bewerkstelligen: nml. het nemen van de snit, en het spelen van een kleine ruiten uit de hand. De eerste speelwijze is de default.

De auteurs gaan er blijkbaar vanuit dat de lezer voor het 'eerst ruiten spelen' automatisch kiest voor de default-speelwijze, want in r.18 laten zij de lezer met schoppen naar de dummy gaan om de snit te nemen. Door te laten zien dat deze speelwijze ook tot down leidt, maken zij de lezer er op attent, dat er behalve 'speel eerst ruiten' nog meer constraints zijn, nml. 'creeer geen zwakke plekken'. Door schoppen naar de heer te spelen is aan deze eis niet voldaan: na het spelen van ♠H is immers de schoppenkleur net zo gevaarlijk als de hartenkleur.

In r.30-32 geven zij de oplossing van het spel. Omdat met het toepassen van de default-speelwijze in ruiten een constraint wordt geschonden, moet voor de andere speelwijze in ruiten gekozen worden: het spelen van een kleine ruiten uit de hand. De motivatie van de auteurs begint met 'u vindt het niet erg als die ruitensnit verkeerd zit'. Hiermee refereren zij aan het feit, dat deze reparatie (het eerst ruiten spelen in plaats van klaveren) bedoeld was voor de situatie, dat west ♦H en oost ♣H zou hebben. Er wordt dus al rekening gehouden met het feit, dat west een slag zal halen met ♦H. Eén van de effekten van ruiten spelen moet zijn, dat er een slag in ontwikkeld wordt. Of OW bij dit proces ook een slag halen is niet belangrijk.

Experts weten natuurlijk meer dan uit de eenvoudige heuristiek, die leidt tot de constraint 'speel eerst ruiten', duidelijk wordt. Zij zijn precies op de hoogte van de zaken, die er voor zorgen dat het plan, dat aan de constraint voldoet, ook inderdaad slaagt. Deze zaken hebben te maken met de <u>communikatie</u> tussen de tegenstanders, en de verdeling van de hartenkleur.

De enige manier waarop een vrije kleur geinkasseerd kan worden door de tegenstander, is als de bezitter van die lange kleur aan slag kan komen. Dit kan op twee manieren: hij komt aan slag in een kleur die door de leider is aangespeeld; of hij komt aan slag in de kleur die door zijn partner wordt gespeeld. Als dit laatste het geval is - de één komt aan slag, maar kan zorgen dat de ander aan slag komt - dan is er sprake van communikatie tussen de tegenstanders. De meest voorkomende manier van communikatie tussen tegenstanders vindt plaats in de vrije kleur zelf. Als west bijvoorbeeld in een kleur nog V x over heeft, en oost nog x (alle andere kaarten in die kleur zijn er al uit), dan maakt het niet uit wie van de tegenstanders aan slag komt. Als daarentegen de situatie bestaat uit V x x tegenover - (oost heeft ze niet

meer), dan moet per se west aan slag komen om de vrije slagen te inkasseren.

In de harten kleur in het bovenstaande spel is iets dergelijks aan de hand. OW hebben samen 7 harten. Twee distributies zijn hier van belang: 5-2 en 4-3. Als ze 5-2 zitten, dan ontstaat na twee rondjes harten de volgende situatie: V x x - -. Dat wil zeggen de vrije harten alleen inkasseerbaar zijn als west aan slag komt: OW hebben geen communikatie. Als de harten 4-3 zitten, ontstaat de situatie: V x - x; OW hebben dan wel communikatie. Echter, het tweede geval is helemaal niet relevant! Het enige gevaar in het spel bestond uit het feit, dat OW 3 harten slagen konden maken. Als de harten 4-3 zitten zijn dat er maar 2. Met andere woorden, als oost na het nemen van de klaver snit met ♠H aan slag komt, en hij kan nog harten naspelen, dan betekent dit dat de harten 4-3 hebben gezeten: het spel is dan automatisch gemaakt.

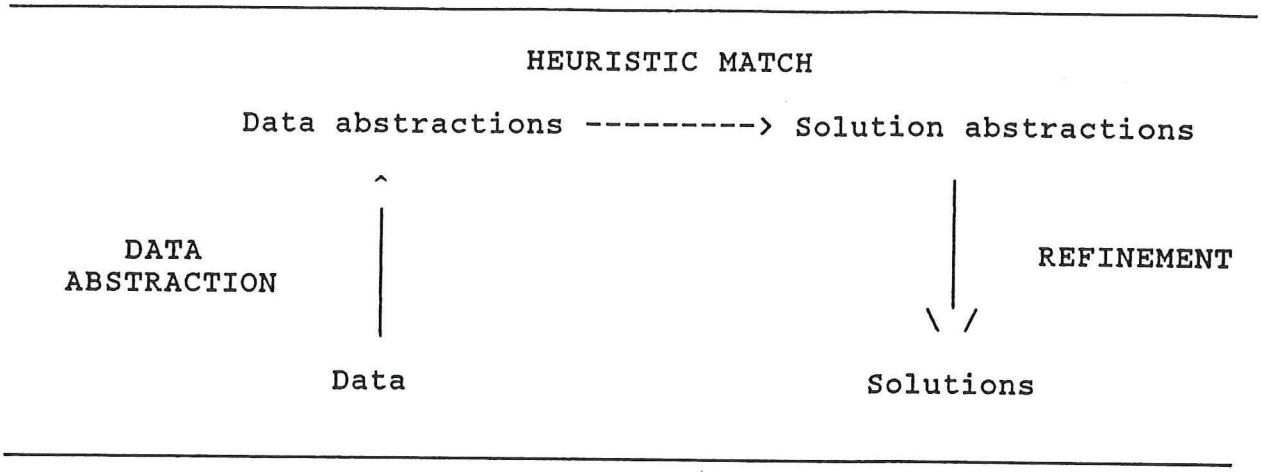
Dat dit verhaal over communikatie ten grondslag ligt aan de regel 'val eerst de entree van de gevaarlijke hand aan' wordt uit de regel niet duidelijk. Het lijkt ook niet zo te zijn, dat al deze kennis van belang is voor het suksesvol kunnen toepassen van de regel. Wie de regel uit zijn hoofd leert (zoals veel beginners zullen doen), zal in veel gevallen de juiste speelwijze vinden.

Andersom is het ook zo, dat de regel niet noodzakelijk is voor degenen, die de kennis die aan de regel ten grondslag ligt beheersen. Dat er in het boven beschreven spel eerst ruiten gespeeld moet worden, volgt geheel logisch uit de kennis over communikatie, entrees, en het inkasseren van vrije slagen. Belangrijk is echter, dat die kennis niet gebruikt hoeft te worden, omdat het herkennen van de toepasbaarheid van de regel een veel kortere weg is naar de oplossing dan het logisch af leiden van de laatste.

De regels vormen dus een 'short-cut' naar de oplossing voor de expert. Zij stellen de expert in staat om uit het ene feit in één stap een ander feit af te leiden, zonder dat daar een keten van tussenliggende inferenties aan te pas hoeft te komen. Het gebruik van een dergelijk soort regels om een oplossing te vinden wordt door Clancey (1985) 'heuristische classificatie' genoemd.

Regels van het soort zoals die in het bovenstaande spel gebruikt werd, leggen een heuristisch verband tussen een beschrijving van een situatie, en een klasse speelwijzen. Een dergelijk verband "is uncertain, based on assumptions of typicality, and is sometimes just a poorly understood relation. A heuristic is often empirical, deriving from problem-solving experience; heuristics correspond to the 'rules of thumb', often associated with expert systems" (Clancey 1985, blz.294-295). Heuristieken zorgen voor een beperking van het zoeken, omdat zij de tussenliggende relaties overslaan; daarom echter is de geldigheid van de relatie onzeker, omdat de te algemene regel in deze specifieke situatie misschien niet opgaat. Clancey geeft de inferentie struktuur van de heuristische classificatie methode als volgt schematisch weer (zie

fig.6.1):



figuur 6.1: Inferentie struktuur van heuristische classifikatie (uit Clancey 1985).

Drie inferentie processen zijn in dit model werkzaam: data-abstraktie classificeert data tot bepaalde abstrakte klassen; heuristisch matchen associeert zo'n abstrakte klasse met een abstrakte klasse oplossingen (of aspekten van oplossingen); verfijning 'vertaalt' de abstrakte oplossing naar de bestaande situatie. Het proces dat ik tot nog toe 'herkenning' heb genoemd - het associeren van kaartpatronen met speelwijzen - is precies te ontleden in de drie bovengenoemde inferentie-processen. Hieraan moet echter worden toegevoegd, dat de herkenning niet altijd via een abstrakte klasse behoeft te verlopen. In het eerder genoemde geval van de safety-play zagen we bijvoorbeeld, dat het de specifieke kaartcombinatie was, die als cue diende voor de herkenning van de safety-play. In dit geval was er dus sprake van een direkt verband tussen kaarten en speelwijze, zonder dat de tussenliggende stappen van abstraktie en verfijning werden genomen.

Recapitulatie

Het zoekproces zoals dat in de analyse van dit spel plaats heeft lijkt het zelfde verloop als dat in de protokollen. Er werd echter weinig gebruik gemaak van defaults; dit was niet nodig omdat distributie en tegenspel geen rol speelden in het spel, zodat de zoekboom maar erg weinig vertakkingen kende.

De manier waarop de reparatie gevonden werd was door herkenning: de situatie werd herkend als behorende tot een klasse situaties waarin een bepaald type speelwijze in toegepast kon worden. Na het vertalen van de abstrakte bewoordingen van de speelwijze naar de huidige situatie was echter de oplossing nog niet gevonden: de heuristiek noemde slechts de kleur die gespeeld moest worden, en

niet de wijze waarop dit moest gebeuren. De keuze tussen de verschillende manieren werd gemaakt op basis van 2 constraints: één die afkomstig was uit de voorafgaande analyse (nml. dat er een extra slag ontwikkeld moest worden), en een algemene, die stelde dat tijdens de uitvoering van het plan geen zwakke plekken gecreeerd mochten worden.

Het lijkt erop, dat herkenning niet beperkt is tot het begin van het spel. Op elk moment tijdens het zoeken kan een situatie herkend worden, zodat het oplossings proces zeer efficient kan verlopen. Het betekent wel, dat bridgers een enorme hoeveelheid kennis over het spel moeten bezitten (d.i. domeinkennis).

Interessant is ook de **flexibiliteit** van de herkenning. Omdat situaties herkend kunnen worden als behorende tot een bepaalde abstrakte klasse, moet de representatie van de situaties zodanig zijn, dat ten alle tijden de beschrijving van de situatie vergeleken kan worden met de beschrijving van een abstraktere klasse situaties. Omdat er geen beperkingen lijken te bestaan ten aanzien van de gebruikte heuristieken, moeten situaties wel op zeer veel verschillende abstraktienivo's bij de mens gerepresenteerd zijn.

HOOFDSTUK 7: RESULTATEN

In dit hoofdstuk zal een beschrijving worden gegeven van de taak in termen van het 4 lagen model van expertise, zoals dat in hoofdstuk 3 uiteen werd gezet. Bij de benaming van de verschillende metaklassen en KS-en is waar mogelijk de terminologie van Breuker (1987) aangehouden. Omdat echter hun typologieën van beide niet compleet zijn (zoals de schrijvers erkennen, zie b.v. blz.40), zullen zonodig eigen termen geintroduceerd worden.

Inferentie-struktuur

De inferentie-struktuur is weergegeven in fig.7.1. (blz.74). In de figuur zijn metaklassen voorgesteld door een rechthoekig kader, en KS-en door een ovaal. De richting van de pijl geeft de relatie tussen beiden weer: een pijl die loopt van een bepaalde metaklasse naar een KS geeft aan, dat die metaklasse de input vormt van de KS; een pijl van een KS naar een metaklasse geeft aan, dat die metaklasse de output is van de KS. Zie hoofdstuk 3 voor de preciese inhoud van deze begrippen.

In het nu volgende stuk zullen de in de struktuur voorkomende metaklassen en KS-en besproken worden.

Meta-klassen

Data

De data vormt de beschrijving van de probleemsituatie die de speler voor zich krijgt. Deze heeft drie aspekten, nml. de NZ-kaarten, de uitkomst, en het kontrakt. Uit de eerste twee komt de *system description* voort; uit de laatste de *constraints*. Een mogelijke aanvulling van deze drie kan bestaan uit gegevens uit het eerdere bieden (met betrekking tot de kaarten van OW), en uit gegevens over de kwaliteit van de tegenstanders. Het eerste vormt een constraint aangaande de mogelijke *environments*, de tweede is van belang voor het genereren van tegenspel in het *system model*.

System description

De system description vormt een uitgebreide beschrijving van de data. In deze beschrijving zijn de data georganiseerd in kleuren; alle kennis die over de kleuren in isolatie (lokaal) bekend is komt in de beschrijving tot uiting. Voorbeelden van concepten behorende tot deze metaklasse: winner/loser count; maximaal aantal te halen slagen; maximaal aantal te halen slagen voor de tegenstander; de condities hiervoor. Ook tot de system description behoren eventuele feiten over het voorafgaande bieden (en wat hieruit afgeleid kan worden), en feiten over de uitkomst.

Tactical hypotheses

Een tactical hypothesis is een door de data gesuggereerde mogelijk te ondernemen aktie, samen met alle kennis die in het kader van het uitvoeren van die aktie van belang is. Voorbeelden van die kennis: de uitvoeringskondities van de aktie; de relevantie distributies (*environments*); de effekten die het uitvoeren van de aktie heeft in elke environment (b.v. wie er aan slag komt, de kans hierop, het aantal slagen dat gehaald wordt door NZ en OW). Een tactical hypothesis kan verschillende vormen aannemen: het kan zijn: een primitieve aktie, een reeks primitieve akties (al of niet conditioneel), of een verwijzing naar een reeks primitieve akties (gecompileerd).

Alle kennis in de bovengenoemde beschrijving van een taktische hypothese hangt met elkaar samen. Een taktische hypothese met de voor die hypothese relevante kennis zou gerepresenteerd kunnen worden als een struktuur met de volgende slots:

```
speelwijze
   |  suggererend_patroon
   |     |  kaarten
   |     |  situatie_beschrijving
   |  akties
   |  uitvoeringscondities
   |  effekten (optioneel)
   |     |  omgeving
   |     |     |  OW-distributie
   |     |     |     |  kaarten
   |     |     |     |  kans
   |     |     |  slagen-NZ
   |     |     |  slagen-OW
   |     |     |  aan_slag
   |     |     |     |  speler
   |     |     |     |  kans
```

De waarde van elk slot kan waar nodig worden opgezocht door te kijken onder de desbetreffende speelwijze. De effekten van een bepaalde speelwijze hoeven niet altijd van te voren al duidelijk te zijn. Bij lokale speelwijzen als het nemen van een snit zal dit wel het geval zijn: er zijn maar twee relevante omgevingen, en de effekten laten zich beschrijven in eenvoudige termen (nml. het aantal door beide partijen gehaalde slagen, en wie er aan slag komt). Bij globale speelwijzen, waar meerdere kleuren in het spel zijn, is dit natuurlijk niet het geval: er zijn zoveel relevante omgevingen, dat er niet van 'het effekt' gesproken kan worden. De effekten die de speelwijze kan hebben moeten via simulatie aan het licht gebracht worden. Hoe abstrakter een speelwijze is, hoe eerder dit het geval zal zijn.

Constraints

Een constraint is een voorwaarde waaraan de oplossing moet voldoen. De eerste constraint kan gevonden worden in de data, nml. het kontrakt; tijdens het plannen wordt deze constraint steeds verder verfijnd. Een constraint kan de vorm aannemen van een eis, maar ook van een verbod. Wat geeist of verboden wordt zijn aspekten van een *situatie;* een eis heeft betrekking op het effekt dat de uitvoering van een plan heeft in termen van het aantal te halen slagen; een verbod heeft betrekking op het aantal te halen slagen van de tegenstanders, en op gebeurtenissen die hiermee samenhangen (zoals b.v. het aan slag komen van een 'gevaarlijke' tegenstander).

Plan system

Het plan systeem is een struktuur die bestaat uit drie componenten: een *plan*, een *situation*, en een *environment*.

Plan

Een plan is een geselekteerde taktische hypothese, of een kombinatie van meerdere hypotheses.

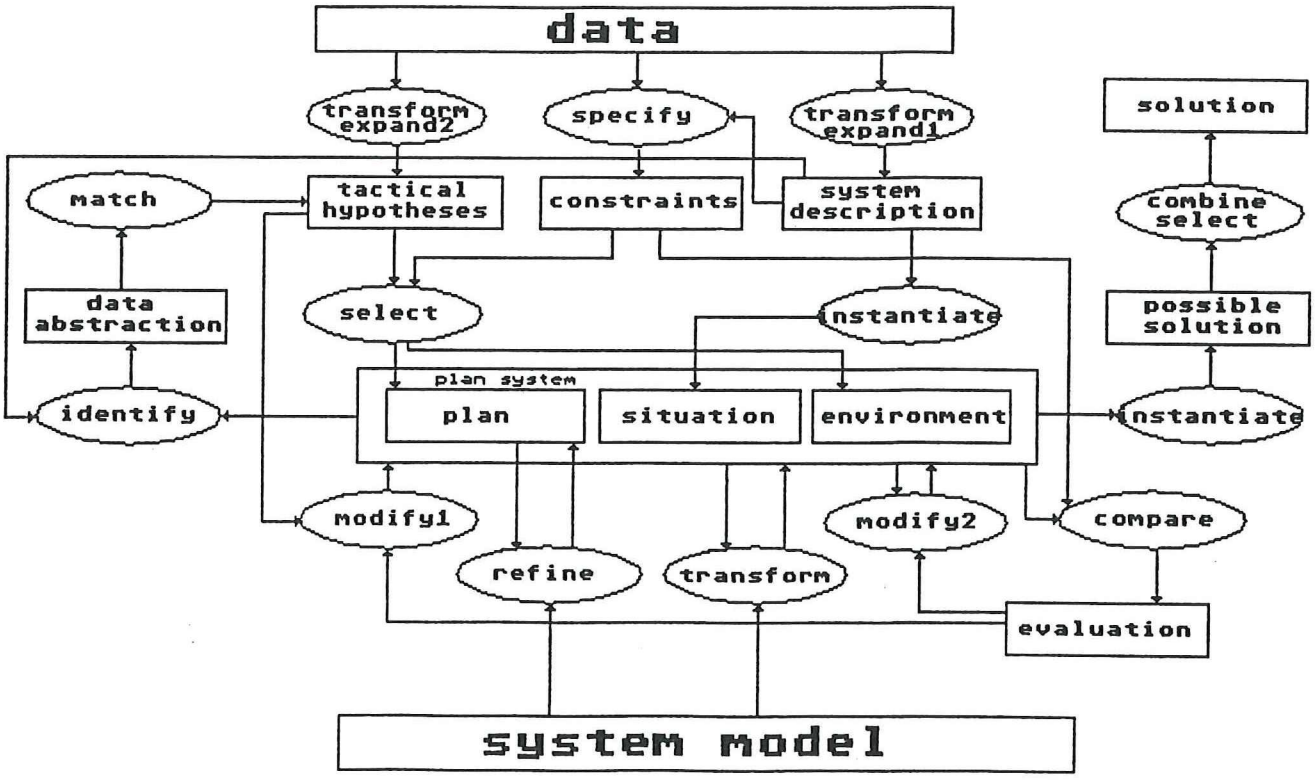
Situation

Een situatie is een beschrijving van een toestand. In de beschrijving komt tot uiting: de nog overgebleven kaarten van NZ en OW; het aantal reeds gehaalde slagen van beide partijen; wie er aan slag is; de geschiedenis van het plansysteem (b.v. welk gedeelte van het plan al voltooid is); de winner/loser count op dat moment.

Environment

Een omgeving vormt een reeks assumpties over de distributie, de plaats van honneurs, en het tegenspel van de tegenstanders. Een omgeving heeft een bepaalde kans van voorkomen. De omgeving wordt gegenereerd uit 1. de taktische hypotheses (distributie en honneur plaats); en 2. het systeem model (tegenspel).

Samen vormen deze drie het plansysteem. Het plansysteem is het uitgangspunt van het simulatie proces. Onder invloed van het plan kan een bepaalde situatie getransformeerd worden tot een andere situatie; na elke transformatie cyclus kan ofwel het plan veranderd worden, ofwel de omgeving.



figuur 7.1: inferentiestruktuur van de planningstaak

Data abstraction

Een data abstraktie speelt een rol bij het genereren van taktische hypotheses tijdens het simuleren, en vormt zelf een abstrakte beschrijving van een klasse situaties. De termen die gebruikt worden in die beschrijving kennen meerdere nivo's van abstraktie. De term 'data abstraktie' is afkomstig uit het model dat Clancey (1985) geeft voor heuristische classifikatie; dit model werd beschreven op blz.69. Een heuristiek is een regel die een klasse situaties associeert met een klasse oplossingen, b.v. 'In situaties van het type X is het normaliter goed om speelwijzen van het type Y toe te passen'. Om zo'n heuristiek te kunnen toepassen is het dus nodig om te herkennen dat een bepaalde situatie behoort tot 'situaties van het type X'. 'Situaties van het type X' is nu een voorbeeld van een data abstraktie: een verzameling situaties die bepaalde eigenschappen gemeen hebben.

Evaluation

Een evaluatie is een uitspraak over de vergelijking van een plansysteem en een verzameling constraints. Een evaluatie kan drie verschillende waarden aannemen: 'positief', 'negatief', en 'onbekend'. De waarde 'positief' geeft aan dat een plan slaagt in een bepaalde omgeving (de omgeving levert dan de *slaagcondities* van het plan); de waarde 'negatief' dat een plan faalt in een bepaalde omgeving (de omgeving levert de *faalcondities*); en de waarde 'onbekend' dat het op dat moment onmogelijk is één van de twee bovenstaande uitspraken te doen.

System model

Het systeem model vertegenwoordigt de kennis over de ontwikkeling van een plansysteem. Er in tot uiting komen: mogelijke vormen van tegenspel + de defaultwaarden ervan in verschillende situaties; hoe een bepaald abstrakt plan herleid kan worden tot een verzameling primitieve akties (mocht dit nodig zijn; dit hangt af van het feit of een speelwijze 'als een geheel' kan worden uitgevoerd, zie 'effekten' bij de beschrijving van de metaklasse 'taktische hypothese'); en hoe in een plansysteem de uitvoering van een bepaald plan in een situatie leidt tot een andere situatie. Een voorbeeld uit de protokollen: de hypothese 'ruiten vrijtroeven' moet eerst worden vertaald in 3 nieuwe hypothesen: 'verwijder kaarten uit de korte hand'; 'ga naar dummy'; 'troef ruiten af' (+ een volgorde constraint). Als elk van deze nog steeds abstrakte beschrijvingen zelf ook weer vertaald zijn naar afzonderlijke instrukties (primitieve akties), moet per primitieve aktie nog het lokale effekt bekend zijn: b.v. dat het spelen van ♦H tot gevolg heeft dat west nu moet bijspelen, dat oost of west aan slag kan komen, dat ♦V dan een slag wordt, etc.

Possible solution

Een mogelijke oplossing is een plan, met bijbehorende slaag- en eventueel faalcondities. Een plan heeft ook een slaagkans. Of een plan wordt beschouwd als mogelijke oplossing hangt af van de evaluatie van dit plan. Is deze evaluatie *positief* dan betekent dit dat het plan (in de in het plansysteem aangenomen omgeving) geen van de aanwezige constraints schendt; in dat geval leidt de uitvoering van het plan tot het gestelde doel, en het plan is dan dus een oplossing.

Solution

Een oplossing is een plan. Dit plan wordt gekozen uit één van de mogelijke oplossingen, of bestaat uit een combinatie van mogelijke oplossingen.

Knowledge sources

In het model van Breuker en Wielinga worden knowledge sources gekarakteriseerd door 4 zaken: de input, de output, de methoden die gebruikt worden om de input in de output om te zetten, en de domein kennis die hierbij nog wordt gebruikt. De methoden zijn in feite het belangrijkste aspekt: als in een KS een standaard methode gebruikt wordt (b.v. inheritance), betekent dit dat die KS in een toekomstig kennis-systeem zonder problemen geimplementeerd kan worden. Helaas is stand van zaken wat bekende formalismen betreft niet erg rooskleurig, zodat in plaats van te refereren aan een bekend formalisme een funktionele beschrijving gegeven zal moeten worden. Dit werd in Hoofdstuk 5 al aangegeven. Als het niet mogelijk is om de in de KS gemaakt inferenties in termen van bekende methoden te beschrijven, zal geprobeerd worden om in ieder geval in abstraktie enige suggesties te doen omtrent de aard van de gebruikte methoden. Om die hier ver uit te werken heb ik niet geprobeerd; dit zou het onderzoek te ver van de cognitieve psychologie verwijderen, en bovendien zou het de duur van dit onderzoek waarschijnlijk met enige jaren verlengen.

De analyse van de data ('perceptie')

knowledge source: **transform/expand1**
Deze KS zorgt ervoor, dat de beschrijving van het systeem zoals die geleverd wordt in de data, verdiept en uitgebreid wordt.

input: **data**
Het gedeelte van de data dat input is voor deze KS is niet meer dan 26 kaarten, gegroepeerd in 4 kleuren.

output: **system description**

methoden: zie onder
domein kennis: Serie patroon/concept definities

Voor de winner/loser count zijn eenvoudige definities in termen van dicht bij de kaarten liggende concepten voldoende (b.v. 'aantal kaarten in de korte hand' of 'ontbrekende honneurs'); methoden voor het classificeren van kaartconfiguraties zijn er in ruime mate. 'Direkte associatie' is ook mogelijk, en bij mensen waarschijnlijker; bij deze methode wordt bij elke configuratie direkt (dus zonder tussenliggende definities) de bijbehorende winner/loser count opgezocht. Vermoedelijk nog het meest in aanmerking komt een 'twee-traps-classifikatie': direkte associatie brengt de configuratie op basis van honneurs en lengte onder in een bepaalde klasse; daarna wordt de preciese winner/loser count bepaald op basis van de tussenkaarten. Voor de overige concepten in de system description kan iets dergelijks gelden.

knowledge source: **transform/expand2**
Deze KS genereert uit de data een reeks taktische hypotheses. De generatie is hoofdzakelijk lokaal; dit wil zeggen dat het kaartcombinaties binnen één kleur zijn die leiden tot speelwijzen die alleen op die kleur betrekking hebben.
input: **data**
Deze worden gevormd door de kaarten van één kleur.
output: **tactical hypotheses**
methoden: direkte associatie; zie onder
domein kennis: een hierarchie van patronen

Definitionele classifikatie is hier niet meer van toepassing, omdat in een kleur de tussenkaarten van het grootste belang kunnen zijn. Twee methoden lijken gebruikt te worden: de al eerder genoemde direkte associatie; en een methode die het best omschreven kan worden met de term 'lokale simulatie'. Als de configuratie niet direkt als bekend herkend wordt, past de expert een reeks abstrakte speelwijzen op de kleur toe, bij verschillende omgevingen. Deze methode werd toegepast in het programma DECLARER van Napjus (1971). Dat experts zich van deze methode bedienen wordt zeer duidelijk als hen naar een onbekende safety-play wordt gevraagd: zij beschouwen dan elke relevante omgeving afzonderlijk, en kijken welke speelwijze in dat geval tot sukses leidt; vervolgens kiezen zij de speelwijze, die in de meeste gevallen tot sukses leidt.

knowledge source: **specify**
Deze KS zorgt ervoor dat de constraints die door de data geleverd worden (de speelsoort en het aantal te halen slagen) met behulp van de system description verfijnd en uitgewerkt worden.

input: **data**
het kontrakt
system description
output: **constraints**
De constraints hebben betrekking op gebeurtenissen tijdens het plansysteem-transformatie-proces, maar ook op eigenschappen van de te selekteren taktische hypotheses, en op de volgorde van de te ondernemen lokale akties.
methoden: heuristische associatie
domein kennis: heuristische regels

Heuristische associatie is een methode die (in dit geval) via heuristische regels geabstraheerde data koppelt aan abstrakte akties. De geabstraheerde data komen uit de system description; de abstrakte akties worden slechts gekenmerkt door hun effekten. Deze effekten vervullen verder de rol van constraint.

knowledge source: **identify**
Deze KS identificeert een systeembeschrijving of een toestand van een plansysteem als behorende tot een bepaalde abstrakte klasse.
input: **plan system**
system description
De systeem beschrijving dient als input bij het herkennen van hypotheses in de uitgangssituatie; het plansysteem dient als input tijdens het simulatieproces.
output: **data abstractions**
Een data abstraktie is een abstrakte beschrijving van de toestand van het plansysteem of een systeem beschrijving. Een voorbeeld zou kunnen zijn: 'OW hebben al hun toegestane slagen gemaakt, en NZ komen nog 1 slag tekort, en NZ zijn aan slag'.
methoden: Definitionele abstraktie.
domein kennis: Definitie hierarchie

knowledge source: match
Deze KS associeert de abstrakte beschrijving met speelwijzen die mogelijk toepasbaar zijn in een dergelijke toestand van het plan systeem.
input: **data abstractions**
output: **tactical hypotheses**
In dit geval nemen de taktische hypotheses de vorm aan van een abstrakte beschrijving van een te ondernemen aktie, b.v. 'squeeze' of 'throw-in'.
methoden: heuristische associatie
domein kennis: heuristische regels

Samenstelling van het plansysteem

knowledge source: **select**
Deze KS selekteert uit de taktische hypotheses kandidaten, waaruit het te simuleren plan wordt samengesteld.

input: **tactical hypotheses**
constraints
de constraints zijn voorwaarden, waaraan het te simuleren plan moet voldoen. Zij specificeren de eigenschappen van de kandidaten, en de volgorde van hen binnen het plan.

output: **plan**
Het plan zorgt voor de 'zetten' die de ene situatie transformeren in de andere. Als zodanig is een plan 1. een reeks (al of niet conditionele) zetten; of 2. een concept, dat door decompositie is om te zetten in zo'n reeks.
environment
bij elke geselekteerde taktische hypothese behoort tevens de voor die speelwijze relevante vijandelijke distributie en honneur plaatsing. Hieruit wordt één specifieke gekozen (meestal de default).

methoden: matchen, inheritance

domein kennis: concept hierarchie (zie onder)

In principe hoeft er alleen maar een selektie plaats te vinden op alle mogelijke taktische hypotheses, met als criterium het in de constraints gespecificeerde effekt van de hypothese. Mogelijkerwijs bestaat er enige discrepantie tussen de beschrijvingen van de effekten van de afzonderlijke hypotheses, en die van de constraints. De in beide beschrijvingen gebruikte concepten moeten derhalve met elkaar in verband gebracht worden; dit gebeurt in een concept hierarchie. Inheritance is een standaard methode voor het vergelijken van eigenschappen van concepten op een verschillend abstraktie nivo.

knowledge source: **instantiate**
Deze KS creeert een situatie in het plansysteem. Deze is in eerste instantie gelijk aan de situatie vóór de eerste kaart gespeeld is.

input: **system description**

output: **situation**

methoden: vullen van slots van een concept

domein kennis: -

Deze KS doet in feite niet meer dan het kopieren van eigenschappen. De situation is een concept met een reeks slots, wier waarden geheel uit de system description verkrijgbaar zijn.

De simulatie van een plan

knowledge source: **refine**

Deze KS zorgt ervoor, dat het plan uit het plansysteem uitvoerbaar wordt gemaakt. Zoals uit de beschrijving van de metaklasse 'taktische hypothese' duidelijk werd, kan het plan op verschillende nivo's van abstraktie beschreven zijn. Als de hypothese via de KS **transform/expand2** direkt uit de data is voortgekomen, dan bestaat deze uit direkt toepasbare operatoren. Als de hypothese echter tot stand is gekomen door heuristische classifikatie, bestaat deze uit een abstrakte beschrijving van een speelwijze - b.v. 'de ruiten vrijtroeven', 'de zijkleuren elimineren en Oost ingooien', enz. Deze hypotheses behoeven niet altijd direkt uitvoerbaar te zijn, omdat het effekt van de uitvoering afhankelijk is van variatie in distributie en tegenspel. Om het effekt van de uitvoering van de hypothese te kunnen voorspellen, is het nodig dat deze *stap voor stap* wordt uitgevoerd. Om dit te kunnen doen is kennis nodig, kennis over hoe een abstrakt plan in een specifieke omgeving gestalte krijgt. Het plan 'ruiten vrijtroeven' uit het protokolspel kan zo b.v. de invulling krijgen: '1. speel ♦H en ♦V; herhaal tot de ruiten vrij zijn: 2. steek over naar dummy, en 3. troef een ruiten.

input: **plan**
system model

In het systeem model bevindt zich de kennis die nodig is om de vertaling van abstrakt naar concreet uit te voeren.

output: **plan**

Het 'ingevulde' plan bestaat nu uit direkt uitvoerbare operatoren.

methoden: inheritance, matching

domein kennis: Kennis over plandecompositie. Deze kennis kan in een hierarchie georganiseerd worden; door te matchen met de aktuele kaartcombinatie kunnen de waarden van de meest specifieke (maar nog steeds abstrakte) elementen in de hierarchie ingevuld worden.

knowledge source: **transform**

Deze KS zet een plansysteem - plan + situatie + omgeving - om in een ander plansysteem. Deze transformatie heeft voornamelijk betrekking op de situatie: deze wordt onder invloed van het plan, en gegeven de omgeving, veranderd. Het plan verandert ook, in die zin dat gedeelten van

het plan die uitgevoerd zijn, in het getransformeerde plansysteem worden weggelaten. De omgeving verandert tijdens een transformatie niet (zie onder).

input: **plan system**
system model
Het systeem model specificeert hoe een bepaalde operatie de ene situatie omzet in een andere. Dit houdt in belangrijke mate verband met de spelregels (b.v.: als Oost de hoogste kaart heeft gespeeld is hij aan slag; als Zuid een kaart heeft gespeeld dan moet West vervolgens bijspelen, enz.). Ook bevindt zich in het systeem model kennis die nodig is voor het genereren van het tegenspel. De transformerende operatie kan primitief zijn (het spelen van één kaart), maar kan ook bestaan uit een reeks primitieve operaties, die echter als één geheel kan worden uitgevoerd (zoals het nemen van een snit, of het uitvoeren van een eenvoudige squeeze).

output: **plan system**

methoden: zie onder

domein kennis: Een enorme hoeveelheid; zie onder

De hier 'Transform' genoemde KS is een typisch voorbeeld van een niet-primitieve inferentie. Voor het transformeren van een plan systeem naar een nieuw plan systeem is een enorme hoeveelheid kennis nodig, en een overeenkomstige hoeveelheid inferenties. De 'basis-handeling' in de transformatie is het toepassen van een operator - een kaart-speel-instruktie - op een situatie. De eigen operatoren (welke kaart NZ bijspelen) worden gegenereerd door het aktieve plan.

Het toepassen van een operator heeft een bepaald effekt op een situatie: verschillende eigenschappen van de situatie veranderen. Voor het transformatie proces moet het natuurlijk precies duidelijk zijn, welke effekten een operator heeft.

De operatoren van de tegenstanders worden gegenereerd door het systeem model. Hiervoor moet er ook weer enorm veel gebeuren: het tegenspel hangt af van de situatie, dus moet eerst een situatie herkend worden; vervolgens moet via regels worden afgeleid welk tegenspel er in zo'n situatie mogelijk is (en wat 'vervelend' is, en wat 'normaal'); vervolgens moet uit de verschillende mogelijkheden een keuze worden gemaakt (welke keuze moet worden vastgelegd, omdat bij eventuele reparaties later gebacktracked moet kunnen worden); tenslotte moet de gekozen operator worden toegepast op de situatie. Niet alleen hangt het tegenspel af van de situatie, maar ook van de omgeving. Er moet dus ook rekening gehouden worden met de <u>relevante</u> mogelijkheden wat deze omgeving betreft (en wat is 'relevant'??); genomen beslissingen (assumpties) moeten in de environment worden vastgelegd, voor het latere backtracken.

Twee zaken zijn hierbij nog van extra belang. Ten eerste blijkt, dat experts indien dit mogelijk is een plan 'gecompileerd' verwerken. Hiermee bedoel ik dat zij in het simulatie proces niet eerst een plan vertalen in primitieve akties, om daarna deze akties één voor één op de situaties toe te passen; zij beschouwen zo'n gecompileerd plan als een primitieve aktie, en passen deze in zijn geheel toe. De effekten van zo'n plan zijn van dezelfde soort als die van een primitieve aktie, alleen zijn het er wat meer tegelijk. In een gecompileerd plan is meestal alleen de omgeving van belang (en wel het bezit van degene die het laatst in de slag moet bijspelen); het spel van de tweede en derde man 'spreekt voor zich' (d.w.z. het draagt niet bij tot eventuele belangrijke variaties in de zoekboom). In een gecompileerd plan zijn de effekten voorspelbaar, meestal omdat de relevante omgevingen voor dat plan erg klein in aantal zijn, en zoals gezegd het tegenspel er niet toe doet.

Het tweede belangrijke punt betreft de 'extensie' van een transformatie. Zoals ik de transformatie hier heb beschreven omvat die precies de toepassing van 1 operatie op een situatie. Uit de protokollen blijkt echter dat er hele series van dergelijke transformaties plaats vinden, zonder dat er tussentijds geevalueerd wordt. Uit andere analyses komt naar voren, dat er na elke verandering in een situatie geevalueerd kan worden; dit wil dus zeggen, dat het niet veranderen van plan en/of omgeving een beslissing is, die tijdens het simuleren genomen wordt. In principe moet het dus mogelijk zijn op elk moment in het simulatie proces veranderingen in omgeving en/of plan aan te brengen.

Dat 'transformatie' geen primitief proces is, zal uit het bovenstaande blijken. Het is zelfs niet onmogelijk om transformatie als taak op zich te beschouwen. Dit idee komt voornamelijk voort uit het feit, dat 'transformatie' niet een proces is, dat altijd dezelfde taakstruktuur vertoont. Het hangt sterk van de aard van het te simuleren plan af, hoe de transformatie verloopt, en welke kennis er tijdens de transformatie gebruikt wordt. Dit varieren van de taakstruktuur aan de hand van de tegen gekomen problemen is iets wat door **strategische kennis** gestuurd wordt. Simulatie (ook wel **prediktie** genoemd) wordt inderdaad beschouwd als een generische taak in de literatuur. Zie verder blz.93.

knowledge source:	**compare** Deze KS vergelijkt de na de transformatie ontstane situatie met de bestaande constraints.
input:	**plansystem (situation)** **constraints**
output:	**evaluation** zie boven; de waarde van de evaluatie bepaalt mede de volgende taak in de taak-struktuur.
methoden:	matchen, inheritance
domein kennis:	concept hierarchie (zie onder)

Bij het matchen van constraints en situaties komen dezelfde problemen naar voren als bij het selekteren van hypotheses (zie blz.78), en bij het herkennen van deze (zie blz.70): beschrijvingen van zowel constraints als situaties kunnen voorkomen op meerdere nivo's van abstraktie. Een beschrijving van een constraint op een bepaald nivo moet echter kunnen matchen met een situatie-beschrijving op een lager nivo. Door verschillende beschrijvingstermen te organiseren in een hierarchie kan de match m.b.v. inheritance gerealiseerd worden.

knowledge source: **instantiate**
Deze KS creeert een concept 'mogelijke oplossing'. Alle waarden uit het plansysteem worden overgenomen. Het resultaat is een plan, dat als het in een bepaalde situatie en in een bepaalde omgeving wordt uitgevoerd, het gestelde doel realiseert.

input: **plansystem**

output: **possible solution**

methoden: Het vullen van slots van een concept door copieren van waarden.

domein kennis: -

knowledge source: **modify1**
Deze KS verandert het aktieve plan in het plansysteem als de evaluatie uitwijst dat het oude plan niet de gewenste effekten heeft.

input: **plansystem**
Het oude plan dient als uitgangspunt van de modifikatie. Hoe het plan gemodificeerd wordt is afhankelijk van de situatie die heeft geleid tot de negatieve evaluatie.

output: **plan**
Het oude plan kan worden uitgebreid met nieuwe akties; het kan worden voorafgegaan door andere akties; of het kan gedeeltelijk of in zijn geheel worden weggegooid. De nieuwe akties worden geselekteerd uit de op dat moment aanwezige taktische hypotheses.

situation
De nieuwe situatie wordt het uitganspunt van de volgende transformatie cycli. Deze is afhankelijk van de aard van de plan-modifikatie. Als b.v. het plan wordt uitgebreid, dan kan de oude situatie gehandhaafd blijven; worden er nieuwe akties in het oude plan gevoegd, dan wordt de nieuwe situatie de situatie zoals die was op het moment dat de laatste te handhaven aktie van het oude plan was uitgevoerd. Dit terugkeren naar een oude situatie wordt wel backtracken genoemd.

Hoe de omgeving wordt veranderd is hier ook van afhankelijk. Soms worden alle assumpties die reeds zijn gemaakt teruggetrokken; meestal worden deze echter gehandhaafd.

methoden: zie onder

domein kennis: heuristische regels, hierarchieën, causale relaties tussen akties en situaties

'Modify' is ook een KS die moeilijk primitief genoemd kan worden. De volgende stappen zijn nodig voordat verandering van plan kan plaats vinden: uitgangspunt vormt de evaluatie, die duidelijk maakt welke constraints er zijn geschonden; vervolgens is het nodig, dat een causaal verband kan worden aangeduid tussen reeds eerder uitgevoerde akties, en de 'constraint-schendende' situatie. Soms zijn de foutveroorzakende akties de laatst uitgevoerde akties, maar soms zijn het ook akties die 5 slagen geleden zijn uitgevoerd (b.v. het vergeten van een deblokkade). Dit causale verband is nodig om de plaats aan te geven, waar het oude plan veranderd moet worden.
Vervolgens moet beslist worden, welke verandering van het plan de constraint-schendende situatie zal doen vermijden. Hiervoor zijn 2 soorten kennis nodig. Ten eerste is nodig, dat uit de ongewenste situatie een wel gewenste situatie kan worden afgeleid. Deze laatste bestaat niet slechts uit de negatie van de eerste, maar kan ook nog worden aangevuld met andere aspekten van die situatie. Bijvoorbeeld: als de constraint-schendende situatie was: 'west troeft hinderlijk over', dan is een voorwaarde voor de negatie hiervan ('west troeft niet hinderlijk over') het feit: 'west heeft geen hinderlijke troeven'. Ten tweede is er kennis nodig over hoe deze gewenste situatie bereikt kan worden, d.w.z. welke akties van NZ deze situatie teweeg kunnen brengen.
Alle hier genoemde inferenties lijken m.b.v. standaard representaties en technieken te realiseren te zijn.
Een 'snelle' methoden van reparatie kan plaats vinden door middel van het direkt herkennen van reparerende speelwijzen. Het gebruiken van de door herkenning gevonden reparaties wordt waarschijnlijk gestuurd vanuit het strategische nivo (zie blz.88).

knowledge source: **modify2**

Deze KS brengt veranderingen aan in de omgeving, als de evaluatie van het oude plansysteem positief was, of als deze negatief was, maar er geen verdere modifikaties van het oude plan mogelijk zijn (zie onder).

input: **evaluation**

De faal- en slaagcondities dienen als uitgangspunt voor de modifikatie.

plansysteem

Alle informatie aanwezig in het plansysteem is nodig voor deze vorm van backtracken. Zowel plan als situatie moeten terug naar het moment dat de veranderde omgeving van invloed is op het verdere verloop van de simulatie.

output: **plansysteem**
Het resultaat van de modifikatie is een nieuwe omgeving. De 'richting' van de modifikatie is afhankelijk van de evaluatie, en van de totale toestand waarin het simulatie-proces zich bevindt (zie onder). Als een plan slaagt in een bepaalde omgeving, dan verandert de omgeving in ongunstige zin, d.w.z. dat de nieuwe omgeving ontstaat uit de negatie van de slaagcondities van het plan. Als een plan faalt, en er geen ander plan mogelijk is, dan verandert de omgeving in gunstige zin: de nieuwe omgeving ontstaat uit de negatie van de faalcondities van het oude plan.
Het plan en de situatie worden overeenkomstig aan de verandering van omgeving aangepast.

methoden: vullen van slots, associatie
domein kennis heuristische regels

De beslissing of en hoe een omgeving wordt veranderd is van veel faktoren afhankelijk. In principe is het 'hoe' in een eenvoudige set regels te vangen, maar of de omgeving veranderd wordt, wordt sterk beinvloed door de stand van zaken in het totale planningsproces, bijvoorbeeld welke mogelijke oplossingen er reeds gevonden zijn, en hoe de slagingskansen hiervan zich verhouden tot de verschijningskansen van de nog mogelijke omgevingen. Dit soort kennis hoort echter niet thuis op inferentie nivo, maar op strategisch nivo. Ditzelfde gold overigens voor **modify1**: of een plan veranderd wordt hangt onder meer af van de hoogte van de kans van de faalcondities van het plan, in vergelijk met de eigen slaagcondities en die van andere plannen. Dit werd geillustreerd in het protokol van B.M. (zie blz.38).
Het veranderen van een omgeving heeft als doel om het oplossingsproces zowel snel als doelmatig te laten verlopen, zodat de uiteindelijke oplossing in zoveel mogelijk omgevingen werkt. De doelmatigheid wordt gegarandeerd door het hanteren van default waarden; het veranderen van de omgeving is een aanpassing van deze default waarden. Die aanpassingen kunnen een verschillend karakter hebben. Als blijkt dat een plan werkt in een default omgeving, dan dient de verandering in 'ongunstige' zin te verlopen, d.w.z. dat in de nieuwe omgeving die faktoren die het plan in de oude omgeving deden slagen, vervangen worden door alternatieve faktoren. Bijvoorbeeld: als een snit goed zit in de oude omgeving, zit hij mis in de nieuwe; als een kleur 3-2 zit, wordt dat 4-1, enz. In principe kunnen alle default assumpties die in de loop van de simulatie zijn gemaakt op deze wijze worden vervangen door hun ongunstigere alternatief.
Als een plan niet slaagt in een bepaalde omgeving, wordt eerst geprobeerd om het plan te repareren. Als echter blijkt dat er in die omgeving geen enkel plan bestaat waarvan de slaagcondities een grotere verschijningskans hebben dan die van de slaagcondities van het te repareren plan, dan heeft repareren geen zin; de slaagcondities van het plan krijgen nu de status van voorwaarden.
Een voorbeeld: planA slaagt als de snit goed zit, maar faalt als

die fout zit. De slaagcondities van planA ('snit goed') hebben een kans van .5. In het geval dat de snit mis zit is er maar één reparerende speelwijze planB ('aas slaan'). De slaagcondities van planB (heer sec) hebben echter een veel kleinere kans dan .5, en planB slaagt dus in veel minder gevallen dan planA. Echter, planA slaagt alleen in de omgeving 'snit goed'. Uit het onderzoeken van planA en planB blijkt dus, dat er geen plan bestaat met een slagingskans van 1: de 'grootste' omgeving heeft een kans van .5. Deze omgeving zal als basis dienen voor eventueel verder onderzoek. Het is b.v. mogelijk dat er nog andere default variabelen te veranderen zijn; dit moet dan nog gechecked worden: 'Hoe minder je vraagt, hoe eerder je bediend wordt'.

Het kiezen van de definitieve oplossing

knowledge source:	**combine/select** Deze KS zorgt ervoor, dat uit de gevonden mogelijke oplossingen een keuze wordt gemaakt; zomogelijk wordt geprobeerd om deze mogelijke oplossingen te combineren, zodat de uiteindelijke oplossing een grotere slagingskans heeft dan elke mogelijke oplossing afzonderlijk.
input:	**possible solutions** Een mogelijke oplossing is ieder plan dat ooit in het simulatie proces als positief geevalueerd is.
output:	**solution** De oplossing is een plan: een reeks voorgenomen handelingen die tijdens het afspel de te spelen kaarten genereert.
methoden:	zie onder
domein kennis:	-

Het vergelijken van mogelijke oplossingen is geen probleem; nodig is alleen het opzoeken van de slaagcondities van de plannen; het plan waarvan de slaagcondities de hoogste kans van voorkomen hebben wordt geselekteerd. Het combineren van plannen is ingewikkelder. Uit de bestudeerde protokollen wordt niet duidelijk hoe experts dit probleem aanpakken, maar het probleem op zich is in de literatuur wel bestudeerd. Sacerdoti (1977) hield zich bezig met een planningstaak, waarin 2 opdrachten uitgevoerd moesten worden, op zodanige manier dat het uitvoeren van de ene taak het uitvoeren van de andere taak niet onmogelijk zou maken. In zijn geval betrof het het schilderen van een ladder, en het schilderen van een plafond; het laatste moest uiteraard het eerst gedaan worden, omdat het schilderen van de ladder het schilderen van het plafond onmogelijk zou maken. Sacerdoti loste dit op door 'hierarchische decompositie': ieder plan werd ontleed in deelplannen, met verschillende nivo's van verfijning. Op elk nivo werd gekeken of de voorwaarden, die aan de verschillende deelplannen werden gesteld, niet door het uitvoeren van andere deelplannen geschonden werden. Was dit het geval, dan werd de volgorde van de uitvoering van de deelplannen veranderd. Dit

proces ging net zo lang door, tot ieder plan ontleed was in primitieve akties; de volgorde van die akties voldeed aan alle constraints op verschillende nivo's.
Het combinatie probleem bij bridge is precies hetzelfde als Sacerdoti's probleem. De vraag is echter of de gebruikte oplossings methoden wel van toepassing kunnen zijn. Sacerdoti had maar te maken met 2 te combineren plannen, die elk bovendien maar uit een klein aantal stappen bestonden. Het aantal mogelijke oplossingen bij bridge kan aardig groot zijn; elke oplossing bestaat uit een groot aantal primitieve akties. De combinatie hiervan kan het oplossingsproces wel eens uit de hand doen lopen. Ongetwijfeld zal het oplossingsproces flink te optimaliseren zijn met behulp van bijvoorbeeld 'plausibele combinatie kennis', maar een definitieve uitspraak kan aan de hand van de geanalyseerde protokollen niet gedaan worden.

De taakstruktuur

Een typische taakstruktuur is de volgende:

```
make-plan (data) :=

    transform/expand1 (data) -> system description
    transform/expand2 (data) -> tactical hypotheses
    identify (system description) -> data abstractions
    match (data abstractions) -> tactical hypotheses
    specify (data, system description) -> constraints
    select  (tactical hypotheses,  constraints) -> plan,
                                                   environment
    instantiate (system description) -> situation
    simulate (plansystem) -> possible-solutions
    combine/select (possible-solutions) -> solution

    Simulate (plansystem):=

    WHILE (tactical hypotheses OR no clear best-solution)
       refine (plan) -> plan
       transform (plansystem) -> plansystem
       compare (plansystem, constraints) -> evaluation
       identify (plansystem) -> data abstractions
       match (data abstractions) -> tactical hypotheses
          IF evaluation=neg. modify1 (plansystem,tactical
                                hypotheses) -> new-plansystem
                           *modify2 (plansystem) ->
                                              new-plansystem
                             simulate (new-plansystem)
             evaluation=pos. instantiate (plansystem) ->
                                          possible-solution
                             modify2 (plansystem) ->
                                              new-plansystem
                             simulate (new-plansystem)
             evaluation=?    simulate (plansystem)
```

*tweede optie

Strategisch nivo

Het strategische nivo bestuurt de uitvoering van de taakstruktuur. Dit wil zeggen dat aan de hand van de eigenschappen van het op te lossen probleem wordt bepaald hoe de taakstruktuur wordt opgebouwd. Bovendien vindt op dit nivo het monitoren van de taakuitvoering plaats: als er iets mis gaat, of als er ontwikkelingen zijn die een snellere oplossingsmethode mogelijk maken, dan moet de taakstruktuur hieraan overeenkomstig aangepast worden.

Op dit nivo bevindt zich de kennis, die vooral bepaalt hoe de coordinatie van de verschillende simulaties verloopt. De eerste stadia van het oplossingsproces lijken voor alle experts min of meer hetzelfde te verlopen: de beschrijving van de data met behulp van hogere orde concepten, het herkennen van mogelijk te ondernemen akties, en het formuleren van constraints, zijn allen absolute voorwaarden voor het suksesvol oplossen van de taak. Daarna kunnen op diverse manieren variaties optreden.

Ten eerste is van belang of een te simuleren plan 'gecompileerd', of kaart voor kaart uitgevoerd moet worden. Ten tweede moet na elke evaluatie beslist worden, of één van de tot dan toe gevonden mogelijke oplossingen goed genoeg is om het simulatie proces te stoppen. Ten derde is een dergelijk overzicht van belang bij het bepalen of een bestaand plan nog gerepareerd moet worden, of dat dat plan het niet meer waard is om gerepareerd te worden, met alle gevolgen van dien. Ten vierde is op dit nivo bekend, of een situatie direkt herkend is (b.v. een bepaalde reparatie), of dat er via de lange weg (backtracken en herhaalde simulatie) naar de reparatie gezocht moet worden. Strategische kennis is ook van belang bij beslissingen ten aanzien van de tijdsduur van het oplossings-proces.

Dergelijke kennis is bijzonder moeilijk te beschrijven. Er bestaat grote variatie tussen de experts onderling, en in de faktoren die bepalen of een expert 'tevreden' is met een gevonden oplossing, of dat hij nog verder zal moeten ploegen in het woud van distributies en soorten tegenspel. Vaak, als het te ingewikkeld wordt, besluiten veel spelers voor het eerste. Wat 'te ingewikkeld' is, is van speler tot speler verschillend. De vaardigheid om bij het simuleren van een bepaald plan de verschillende takken van de boom strikt gescheiden te houden is hier van groot belang. In de praktijk blijkt, dat zelfs topspelers in gevallen met veel variatie wat mogelijke plannen en relevante distributies betreft, door de bomen vaak het bos niet meer zien.

HOOFDSTUK 8: TOETSING VAN HET MODEL

Om de gevonden inferentiestruktuur en taakstruktuur te toetsen werd aan de proefpersonen een tweede probleem voorgelegd. Dit probleem is enigszins anders van karakter dan het eerste probleem in die zin, dat in het eerste probleem de oplossing goed geevalueerd kon worden. In het tweede probleem is dit veel minder het geval: zelfs van het 'beste' plan kan niet gezegd worden, dat het tot sukses zal leiden. Hierdoor ligt de nadruk in het probleem meer op het herkenningsaspekt van de taak dan op het simulatieaspekt, alhoewel beiden natuurlijk naar voren komen.

Hieronder volgen de drie protokollen. Omdat voor de toetsing slechts het maken van het speelplan van belang is, heb ik alleen dat deel van de protokollen weergegeven, dat het denkproces reflekteert tòt het moment dat de leider zijn kaart voor de tweede slag wil gaan spelen. Na elke opmerking van de proefpersonen volgt een specifikatie van de KS-en die volgens het model aan de opmerkingen ten grondslag hebben gelegen.

Probleem 2

Het biedverloop was als volgt:

Zuid	Noord
1 ♠	1 ♥
2 SA	pas

♠ A 9 6 3
♥ B 9 8 2
♦ 8 7 4
♣ 10 4

N

Z

♠ H V 5
♥ A 5
♦ A V 6 3
♣ V B 7 2

E.L.

(♦2)

1. derde vijfde
transform/expand1: consequenties van uitkomst voor ♦-distributie

2. even kijken
3. 3 schoppens, een harten, twee ruiten, een klaver
transform/expand1: winners tellen; Transform/expand2: herkent ♠ spelen als mogelijkheid

4. 3 5 6 7 geen flauw idee
transform/expand1: totale winnercount
specify (constraints): doel=8 winners; het plan moet 1 winner

extra creeren

5. nou de schoppens kunnen zo zitten
transform/expand2: schoppen spelen kan een extra slag opleveren bij bepaalde distributies

6. uh hoe zal ik dit gaan spelen
7. nou we kunnen eindelijk uiteindelijk nog altijd eindigen met uh squeezen squeezen
identify, match: herkent 'squeeze' als mogelijke speelwijze

8. pff caramba
9. ik begin maar eens een kleintje ik kan er nog niet zoveel aan verzinnen
(♦4) (♦10)
10. die ga ik niet zakken, dat lijkt me helemaal debiel
transform/expand2: herkent 'ruiten zakken'; select,transform: simuleert 'ruiten zakken'; compare: vergelijkt situatie na het duiken van ♦V met constraints: evaluatie negatief, want het kost een slag (♦V is dan geen winner meer).

11. die kan toch niets meer terug...
12. misschien moet ik die mensen wel laten spelen
13. die mensen aan slag, dat is ook niet zo gek
identify, match: herkent de mogelijkheid 'tegenstanders laten spelen'

14. naah blll wat een kutzooi
select-refine-transform-compare: negatief

15. ja we kunnen ook gewoon...
Transform/expand2: herkent ♥ spelen als mogelijkheid

16. nee dat slaat ook nergens op V 10 x harten links gaan spelen
select-transform-compare: negatief (entree tekort)

17. in het eindspel kan dat altijd nog gebeuren, V 10 x
18. even denken hoor
19. 1 2 3 4 5 6 7
herhaalt r.4

20. klaver voor de 10 van rechts en die zal wel harten naspelen
21. welke ruiten kwam die uit de laagste geloof ik
22. dus links zal waarschijnlijk 5 ruitens hebben
23. als ik nou klaver voor de 10 speel dan speelt die man harten terug voor links, en die speelt ruiten terug
select-refine-transform: simuleert ♣ ontwikkelen. De transformaties verlopen als volgt: Z: ♣x; W: (default) ♣x; N: ♣10; O: (default) ♣H; O: (default) ♥ na; Z: (default) ♥x; West: (default) ♥V; N: ♥x; W: (default) ♦ na (r.21/22: transform/expand1)

24. en dan ben ik 3 ruitens, 2 klavers en een harten kwijt
25. dat is toch niet gunstig
compare: negatief (1 loser meer dan toegestaan)

26. wat is dit weer moeilijk zeg ik heb geen flauw idee
27. ja dan moet ik dus toch gewoon ♥A de tweede keer pakken
28. ♠V nee dat slaat toch helemaal nergens op dit
modify-transform-compare: negatief; repareert plan door in de voorgaande simulatie de laatste eigen default speelwijze (♥x) te vervangen door ♥A. De simulatie vervolgt met de volgende stap uit de realisatie van het plan: ♠V; dit leidt niet tot een gunstigere situatie.

29. ik moet gewoon stom zijn en dan moet rechts op een gegeven moment 4 schoppens vasthouden en 4 klavers
select-refine ('squeeze' vertaald naar '♠♣-squeeze tegen Oost')

30. maar ja dat zijn veel te veel slagen
refine-transform-compare: negatief

31. wat ik ook gewoon kan doen is die man laten spelen
zie r.12

32. ja maar ja de schoppens kunnen ook gewoon 3-3 zitten
herhaling van een reeds bekend feit (zie r.5)

33. harten naar de 8
herhaalt beschouwing van ♥-spel (zie r.15-16)

34. ja als rechts dan ♠AH heeft inderdaad dat is helemaal niks
select (solution): als Oost ♠AH heeft is ♣ ontwikkelen te prefereren boven ♥ spelen; eerst ♣ spelen behoudt de meeste kansen

35. ik heb geen flauw idee
(♠2)

M.V.

(♦2)

1. wat een ballen kaart heeft ie daar, 5 punten, een 4-4 hoog
transform/expand1

2. er zitten allemaal opties in de schoppen zie ik al
transform/expand2

3. even kijken
4. ik krijg ruiten start
transform/expand1

5. ik heb 2 ruiten, 3 schoppen van boven is 5, een harten is 6
transform/expand1

6. als ik een klaver kan ontwikkelen
transform/expand2

7. ik kan maar 1 klaver ontwikkelen
specify (constraints): nodig zijn 2 slagen, ♣ spelen levert maar 1 slag

8. als ik de ruiten pak, ik speel klaver, spelen ze weer ruiten door, of misschien wel harten, de hufters
select-refine-transform-compare: negatief; de negatieve evaluatie komt voort uit het ♥ naspel van Oost

9. denken hoor (zucht)
10. ik kan niet naar dummy om die klaver uit dummy te spelen
modify1-(identify, match)-refine-transform-compare: negatief; het plan uit r.8 wordt gerepareerd door eerst ♣ uit dummy te spelen. De default speelwijze voor Oost is dan klein, zodat West de eerste ♣ slag wint. Voor West is dan de default speelwijze ♦ na, en niet ♥. Deze speelwijze vermijdt dus inderdaad het ♥ naspel. Evaluatie is negatief omdat de uitvoering van het gerepareerde plan ♠A te vroeg wegspeelt.

11. want dan als links aan slag komt dan is het veel minder aantrekkelijk om harten in te kaarten
verklaart de gekozen reparatie

12. wat zijn m'n potentiele verliezers
13. 2 klavers, als ze harten doorspelen wat harten slagen, 2 of 3, een ruitje
transform/expand1: extra concept maakt meer heuristieken beschikbaar

14. ruiten ophouwen lijkt me ook niks
15. als ze dan harten switchen ben je helemaal de lul
16. ik moet in ieder geval de ruiten pakken als rechts de heer komt
transform/expand2-select-transform-compare: negatief; herkent 'ruiten ophouden', voorspelt effekt: Oost blijft aan slag, zal ♥ naspelen (default, zie eerder)

17. nou we zien wel waar het schip strandt
(♦4) (♦10)
18. de 10 nou ja zal ik toch moeten pakken
(♦V)
19. O kijk ik heb 7 ruitens samen dan zitten ze misschien wel 5-1 of 3-3
20. nou toch maar aan de klaver beginnen
select (solution): ♣ spelen. Deze speelwijze maakt in ieder geval van de 6 vaste slagen 7; een speelwijze voor 8 slagen was niet aanwezig

(♣2)

B.M.

(♦2)

1. dat zal wel werken worden
2. nou 2 ruiten slagen 3 schoppens een harten is 6 en een klaver is 7 om mee te beginnen

transform/expand1

3. misschien een extra ruiten slag, misschien een extra klaver slag, em misschien een extra schoppen slag

transform/expand2 (drie maal): herkent de mogelijkheden van resp. ♦, ♣, en ♠ spelen.

4. met die harten is misschien ook nog wel iets te doen maar dat is wel heel vaag

transform/expand2: herkent ♥ spelen, maar bemerkt dat dit niet zo kansrijk is

5. die schoppens kan ik waarschijnlijk wel voor 4 slagen spelen maar dan moet ik ze wel goed doen

transform/expand2: de verwachte 'opbrengst' van schoppen spelen is 1 slag meer dan het aantal winners in die kleur

6. nou ik denk dat ik maar eens een keer de klavers ga ontwikkelen

select (solution): ♣ ontwikkelen. Waarschijnlijk via een heuristiek, b.v. 'als je te weinig winners hebt, speel dan een kleur die wel extra winners voor jou, maar geen extra winners voor de tegenpartij veroorzaakt'. Via refine-transform-compare wordt ♣ spelen als positief beoordeeld: het ontwikkelt een een zevende winner, zonder extra slag te verliezen.

7. alleen even kijken of ik die ruiten moet bukken als de heer verschijnt

transform/expand2: herkent ♦ zakken.

8. als de ruiten 5-1 zitten en ik buk niet, ja dan moet hij 2 entrees hebben om z'n ruitens vrij te krijgen
9. nou dat is niet zo erg

select-transform-compare: positief; selekteert ♦H meteen nemen(!); leidt in dat geval af dat West 2 maal aan slag moet komen om zijn ruiten èn vrij te spelen èen te inkasseren; evalueert dit als niet-negatief.

10. nee ik denk maar dat ik gewoon op de klavers afga

herhaalt gekozen oplossing

11. ja speel maar een kleine ruiten
(♦4) (♦10)
12. ♦10 nou ja dan kan ik gewoon niet meer zakken
13. (♦V)
14. als die derde vijfde start, HB9 vijfde en hij heeft niet gevolgd

15. dat is een beetje typisch
16. dat betekent dat ie er niet veel meer naast heeft
17. nou ik speel in ieder geval eerst een klavertje naar de 10 dan zie ik wel
(♣2)

Net als in de drie eerder geanalyseerde protokollen is E.L. verreweg het volledigst in zijn analyse van het spel. In zijn protokol komen achtereenvolgens de volgende KS-en naar voren:

regel	KS
1.	transform/expand1
2/3.	transform/expand1;transform/expand2 (♠)
4.	transform/expand1; specify constraints
5.	transform/expand2 (♠)
6/7.	identify-match (squeeze)
10.	transform/expand2-select-transform-compare (♦ zakken)
12/14.	identify-match-select-refine-transform-compare (mensen laten spelen)
15.	transform/expand2 (♥)
16.	select-transform-compare (♥ spelen)
20/25.	select-refine-transform-compare (♣ spelen)
27/28.	modify1-transform-compare (variant ♣ spelen)
29/30.	select-refine-transform-compare (Oost in ♠-♣ squeezen)
31.	select solution

Alls we het bovenstaande vergelijken met de taakstruktuur zoals die werd geformuleerd op blz.86, dan blijken deze goed overeen te komen. Na een periode van herkenning en inventarisatie (r.1-7) volgt een periode van simulatie van de herkende taktische hypothesen (r.10-30). Hierin is interessant, dat het geselekteerde plan slechts één keer gerepareerd wordt in die zin, dat in het gerepareerde plan het oorspronkelijke plan nog herkenbaar is. Dit gebeurt in r.27/28. In alle andere gevallen bestaat de 'reparatie' uit het geheel en al weggooien van het oude plan, om het te vervangen door een nieuw plan. Dit nieuwe plan wordt weer gekozen uit de set taktische hypothesen. Volgens de taakstruktuur op blz.86 worden beide type reparaties uitgevoerd in de KS 'modify1'. Wellicht zou het beter zijn om bij reparaties onderscheid te maken tussen complete vervanging en gedeeltelijke aanpassing. Het eerste geval zou plaats vinden met behulp van 'select', het tweede door 'modify1'. De keuze tussen deze twee zou gemaakt moeten worden op strategisch nivo, op basis van de aard van de evaluatie.

Het simulatieproces (de cyclus select-refine-transform-compare) komt op 6 verschillende plaatsen in het protokol naar voren. E.L. onderzoekt blijkbaar alle herkende hypothesen. In de taakstruktuur is gespecificeerd, dat het simuleren van plannen door gaat tot òf een bevredigende oplossing is gevonden, òf alle aanwezige

hypothesen gesimuleerd zijn. Gezien de ontevreden opmerkingen over de moeilijkheidsgraad van het probleem en zijn eigen gevonden oplossingen was het eerste zeker niet het geval. Het afzoeken van alle mogelijkheden doet E.L. erg consciëntieus.

Een tweede punt van belang is het kiezen van de oplossing. In dit praktijkprobleem was er geen echt goede oplossing in die zin, dat van het gekozen plan voorspeld kon worden, hoeveel procent slagingskans het had. Het klaver spelen ontwikkelde een slag; dit realiseerde weliswaar niet het doel van 8 slagen, maar bracht het oorspronkelijke deficiet terug van 2 slagen naar één slag. De achtste slag, blijkens de analyse van E.L., was nog ver weg. Dit maken van een keuze tussen voorgestelde oplossingen wordt volgens het model gemaakt in de KS 'combine/select'. Als het model voor het maken van een programma gebruikt zou moeten worden, dan moet in de door die KS gebruikte kennis in ieder geval duidelijk gemaakt moeten worden, welk speelplan er gekozen wordt als geen van de gesimuleerde plannen bevredigend waren. Ook hier zou gebruik van heuristieken uitkomst bieden. In het model werd dit nog niet onderkend.

Het protokol van M.V. volgt het zelfde verloop als dat van E.L. Of de eerste minder spraakzaam was dan de laatste, of dat hij minder precies was in zijn analyse is niet bekend, maar een feit is dat uit M.V.'s protokol een minder diepgaande analyse van het spel naar voren komt dan uit dan van E.L.

1. transform/expand1
2. transform/expand2 (♠)
3. transform/expand1
4. transform/expand1
6. transform/expand2 (♣)
7. specify constraints
8. select-refine-transform-compare (♣ spelen)
10. modify1-(identify-match)-refine-transform-compare (variant ♣ spelen)

12/13. transform/expand1

14/16. transform/expand2-select-transform-compare (♦ ophouden)

20. select solution (♣ spelen)

Ook hier vinden we weer een periode van herkenning en inventarisatie (r.1-7), en een periode van simulatie (r.8-16). Interessant is de vergelijking tussen de simulatie van het plan '♣ spelen' van E.L. en M.V. Beide arriveren bij het punt waar Oost aan slag is en harten naspeelt; E.L. zet de simulatie hierna voort, maar M.V. lijkt sneller te onderkennen dat dit tegenspel vervelende gevolgen kan hebben (E.L. arriveert ook bij deze conclusie, maar bekijkt hiervoor nog 2 varianten: ♥A spelen, en ♥A duiken). De negatieve evaluatie wordt veroorzaakt door de situatie 'Oost is aan slag, en Oost speelt harten'. Nu is het tweede niet

te veranderen, dus om die situatie te vermijden moet verhinderd worden dat Oost aan slag komt. Dit is in theorie mogelijk door klaver van tafel te spelen; Oost kan niet de Aas of Heer spelen zonder Zuid een extra slag te geven. West zal dan aan slag komen, en die zal ruiten vervolgen. Een goed plan, maar helaas in dit specifieke spel niet toepasbaar, vanwege het entreeprobleem. Deze reparatie lijkt M.V. gevonden te hebben d.m.v. een heuristiek; zowel de snelle herkenning ervan, als de abstrakte vorm ervan duiden hierop. Dit komt goed overeen met wat in het model over het repareren van plannen gezegd wordt.

B.M. tenslotte bereikt het snelst van de drie de conclusie, dat er klaver gespeeld moet gaan worden. Dit is geheel in stijl met de manier waarop hij het eerste probleemspel oploste: als een gevonden oplossing aardig lijkt, kiest hij die, zonder verder de eventuele alternatieven uit te werken. In dit spel is 'klaver spelen' nodig om een zevende slag te ontwikkelen; slagen ontwikkelen is nodig om eventueel het kontrakt te maken; conclusie: klaver spelen is goed. De alternatieven, die hij uiteraard wel ziet (r.3/4), acht hij niet veelbelovend genoeg om ze nader te analyseren.

2. transform/expand1;specify constraints
3. transform/expand2 (drie maal: ♦ ♣ ♠)
4. transform/expand2 (♥)
5. transform/expand2 (♠)
6. select solution (♣ spelen)
7. transform/expand2 (♦ zakken)
8/9. select-transform-compare (♦ zakken)

Opvallend is, dat het mogelijk moet zijn via een heuristiek een niet geheel bevredigende oplossing te vinden, en tevens te zien dat andere kandidaten niet veelbelovend genoeg zijn om ze te onderzoeken. Deze strategie laat zich omschrijven als b.v.: 'rangschik eerst de taktische hypothesen op basis van verwachte opbrengst; als er één uitspringt, kies die dan'. Misschien kun je hier spreken van een heuristiek op strategisch nivo: een vuistregel die een bepaalde oplossingsmethode van de taak specificeert, die <u>in de meeste gevallen</u> wel werkt. Het voordeel van het volgen van een dergelijke strategie is snelheid; het nadeel is, dat je in een klein aantal gevallen wel eens wat over het hoofd kunt zien.

Een aanwijzing dat B.M. deze strategie volgt kunnen we ook vinden in zijn eerste protokol. Daarin zegt hij op een gegeven moment:

25. zou ik misschien toch nog iets met die klaver moeten doen
26. schoppen getroefd, dan kan ik eerst een klaver spelen
27. dat had ik misschien trouwens gelijk wel moeten doen
28. waarom deed ik dat eigenlijk niet
29. omdat ik een beetje te intuitief speel denk ik

Die speelwijze kiezen die op het eerste gezicht goed lijkt noemt B.M. 'intuitief spelen'. Hoe suksesvol zo'n strategie is hangt natuurlijk af van de sterkte van de speler die hem toepast. Als 'de verwachte opbrengst' uit bovengenoemde regel door veel ervaring nauwkeurig te schatten is, dan zal 'intuitief spelen' zeer vaak sukses hebben; de tijdwinst die met deze strategie gepaard gaat kan zelfs onverwacht voordeel brengen als de tegenstanders, die minder goed zijn, proberen ook zo snel te spelen. De topspeler Hans Kreyns is hier bijzonder bedreven in. 'Intuitie' lijkt te rusten op een rijke systeembeschrijving en een grote hoeveelheid heuristieken, zodat zonder simulatie toch een nauwkeurige schatting van de effekten van een taktische hypothese gemaakt kan worden. 'Intuitief spelen' is dan de strategie om als oplossing de meest belovende taktische hypothese te kiezen, zonder deze verder uit te werken. Welke faktoren bepalen welke strategie gekozen wordt ('intuitief spelen' vs. 'alle mogelijkheden door simulatie uitwerken') is niet geheel duidelijk. Het kan een bewuste beslissing zijn ('deze tegenstanders zijn erg slecht; als ik snel speel maken ze meer fouten'), of ook een kwestie van concentratieverlies of geestelijke luiheid (niet zelden veroorzaakt door te veel alcohol).

Conclusie

De inferentie- en taakstruktuur, zoals die gegeven werd op resp. blz.74 en blz.87, lijken ook de processen die ten grondslag liggen aan de in deze serie protokollen gemaakte opmerkingen, goed te verklaren. Op twee punten had het model echter wat duidelijker kunnen zijn. Het eerste punt betreft de uitwerking van het reparatie-proces, belichaamd in de KS 'modify1', waarbij wat meer nadruk had kunnen liggen op de complete vervanging van plannen. Het tweede punt betreft de gebruikte kennis in de KS 'combine/select', een KS die uit mogelijke oplossingen de uiteindelijke oplossing kiest. In het geval dat er geen, of slechts incomplete oplossingen gevonden zijn, kan er m.b.v. heuristieken voor gezorgd worden, dat de relatief beste oplossing gekozen wordt.

HOOFDSTUK 9: CONCLUSIES EN GENERALISATIES

In dit laatste hoofdstuk wil ik vier dingen doen. Ten eerste wil ik de beschrijving van de planningstaak, zoals ik die in het vorige hoofdstuk gegeven heb, relateren aan onderzoek dat naar gelijksoortige taken gedaan is; hierbij wil ik laten zien, dat het maken van een plan bij bridge uiteenvalt in een reeks taken, die in de literatuur als generisch beschouwd worden. Ten tweede wil ik aan de hand hiervan enige algemene uitspraken doen over het denken van bridgers. Ten derde (en hier was het, lang geleden, in eerste instantie om te doen!) wil ik nagaan, welke implikaties het gedane onderzoek heeft voor de construktie van een kennissysteem voor de planningstaak; of een dergelijk systeem op korte termein te realiseren is; en de eventuele moeilijkheden die bij de construktie een rol kunnen spelen. Ten vierde zal ik enige evaluerende opmerkingen maken over het 4 lagen-model van expertise, en mijn ervaringen met het gebruik van de bijbehorende terminologie.

9.1 Generalisaties

Eén van de doeleinden van de in dit onderzoek gebruikte methodologie was het struktureren van menselijk probleemoplossend gedrag door middel van het aanleggen van een bibliotheek van zog. generische taken (zie hoofdstuk 4). In dit kader is het dan ook belangrijk, om de beschrijving van de planningstaak zoals die in het vorige hoofdstuk gegeven werd, direkt in verband te brengen met reeds gedaan onderzoek, zodat duidelijk kan worden of de planningstaak als generisch beschouwd kan worden, of dat in de taak verschillende generische taken gecombineerd moeten worden (zie b.v. blz.27).

Aan het einde van hoofdstuk 5 kwam ik op basis van de analyse van de protokollen tot de conclusie, dat het plannen bij bridge de volgende processen een rol spelen:

herkenning
selektie
simulatie
diagnose/evaluatie
reparatie
selektie/combinatie

In de daarop volgende hoofdstukken werden deze processen als volgt uitgewerkt:

Binnen *herkenning* bleken er twee varianten te onderscheiden. De ene variant werd gekenmerkt door een vrij direkte manier van herkenning: een bepaald aspekt van de data werd direkt geassocieerd met een bepaalde, concrete, te ondernemen aktie. De andere variant werd gekenmerkt door wat Clancey (1985) 'heuristische classifikatie' noemt; via heuristische regels wordt

een abstrakte klasse van data in verband gebracht met een abstrakte klasse akties.
Selektie werd beschouwd als primitief inferentie-proces.

Voor *simulatie* gold dit niet; in de inferentiestruktuur werd dit proces gerealiseerd door een serie transformaties op een plansysteem. Hierbij werd veelvuldig gebruik gemaakt van defaults om de doeltreffendheid van de simulatie te vergroten.

In *diagnose/evaluatie* werd bekeken of het gesimuleerde plan wel voldeed aan de aan de oplossing gestelde eisen. In de inferentiestruktuur werd dit proces gerealiseerd door de KS 'compare'; deze vergeleek de eigenschappen van een tijdens de simulatie bereikte situatie met de tot op dat moment geformuleerde constraints.

De *reparatie* van een foutief plan werd gerealiseerd door de KS 'modify1'. De complexiteit van de door deze KS te verrichten funkties (zoals het bepalen van de plaats en de aard van de verandering in het bestaande plan) doet echter vermoeden, dat 'reparatie' beter door meerdere KS-en gerealiseerd zou kunnen worden.

Het selekteren van de oplossing uit de verzameling mogelijke oplossingen, of de eventuele combinatie van mogelijke oplossingen tot één oplossing heb ik laten uitvoeren door de KS 'combine/select'. Deze lichtzinnige beslissing wordt bepaald niet ondersteund door de aard van dit proces; vanwege haar complexiteit geldt hier hetzelfde als voor 'modify1'. Omdat ik echter uit geen van mijn bronnen iets wijzer ben geworden over 'het combineren van kansen' lijkt het me niet verstandig hier te gaan spekuleren over mogelijke realisaties van dit proces. Zie ook blz.86 e.v.

In de literatuur zijn drie van bovengenoemde processen als 'generisch' terug te vinden. Herkenning wordt zowel door Breuker (1987) als door Clancey (1985) beschouwd als een vorm van 'identify', een taak in de klasse van systeem-analyse-taken. Simulatie (door beide schrijvers ook wel 'prediktie' genoemd) wordt ook door beide onder de analyse-taken geschaard. Reparatie wordt door Breuker 'repair' en 'remedial' genoemd, en beschouwd als een systeem-modifikatie-taak; Clancey noemt het 'modify' en beschouwt het als een systeem-synthese-taak. Planning wordt overigens door beide beschouwd als een generische taak; bij beide valt planning onder de synthese-taken. Enige diskussie lijkt gezien alle diskrepanties wel zinvol.

Herkenning

Herkenning is een aspekt van de analyse van een systeem. Op basis van uiterlijke kenmerken moeten uitspraken gedaan kunnen worden over de onderliggende struktuur van het systeem, door deze

uiterlijke kenmerken te vergelijken met een model van het systeem. Onder 'systeem' kunnen veel verschillende zaken verstaan worden. Zo vormt het menselijk lichaam een systeem; door de analyse van allerlei waarneembare kenmerken kan de struktuur van het systeem aan het licht gebracht worden. Een dergelijk analyse-proces heet dan 'diagnose'. In de bridge-planningstaak bestaat dit systeem uit het geheel van de door bridgers gebruikte begrippen en relaties tussen de begrippen. Eigenschappen van dit systeem moeten afgeleid worden uit de waarneembare kenmerken ervan, nml. de 26 zichtbare kaarten.

Strikt genomen lijkt me de eerste vorm van herkenning zoals wij die in de planningstaak tegen kwamen niet te beschouwen als probleemoplos-taak. Er was geen sprake van een bepaalde inferentie-struktuur; de oplossing van de taak (een beschrijving van de struktuur van het systeem) was d.m.v. direkte associatie uit de data verkrijgbaar. Problemen die met deze vorm van herkenning samen hangen liggen meer op het gebied van pattern-matching en geheugen-organisatie deskundigen. Ik zal me er hier dan ook niet verder in verdiepen. Clancey noemt deze vorm van herkennen overigens 'simple classification'.

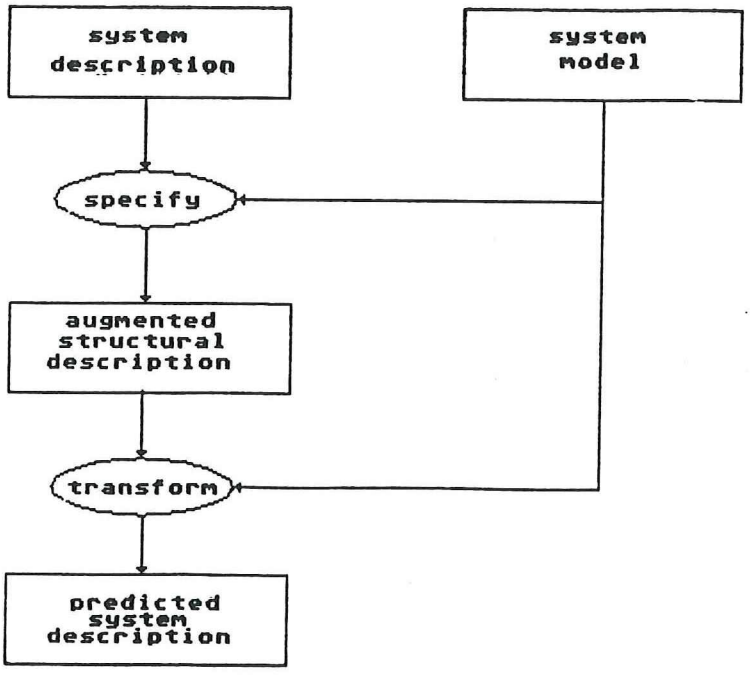
Over de tweede vorm van herkenning valt meer te zeggen. De hierbij gebruikte methode wordt door Clancey (1985) 'heuristische classifikatie' genoemd; deze methode werd beschreven op blz.70. Clancey stelt, dat heuristische classifikatie een methode is die karakteristiek is voor interpretatie-taken, waarbij het probleem bestaat uit het identificeren van één oplossing uit een opsombare verzameling oplossingen: "In heuristic classification the solution space is known to the problem solver as a *set* of explicit alternatives, and problem solving takes the form of 'proving' that one of them is best or acceptable" (Clancey 1985, blz.334). De bestudeerde planningstaak is echter een synthese-taak, waarbij de oplossing geconstrueerd moet worden. De funktie van heuristische classifikatie in construktie-problemen wordt echter duidelijk in Clancey's vervolg: "Often problems of DESIGN and DIAGNOSIS are not amenable to solution by heuristic classification because possible final 'states' cannot be practically enumerated, exhaustively learned (from experience or direct teaching), or for some reason a previously used solution is just not acceptable; solutions must be constructed rather than selected. However, even when solutions are constructed, classification might play a role, for example, in planning the problem-solving process or *in selecting the pieces to be configured.*" (blz.334, mijn accent).

Dit laatste is precies wat we in de protokollen hebben zien gebeuren. De proefpersonen herkenden lokaal (beperkt tot één kleur) of globaal (de gehele situatie) mogelijke speelplannen; deze speelplannen, die ik 'taktische hypotheses' heb genoemd, dienden vervolgens als bouwstenen van het uiteindelijk, via de simulatie, te ontwerpen plan. We kunnen dus concluderen, dat het denkproces zoals de proefpersonen dit ten toon spreidden, wat het gebruik van heuristieken (en de plaats hiervan binnen het totale

proces) betreft niet afwijkt van de reeds in de literatuur beschreven processen.

Simulatie

Simulatie (ofwel prediktie) wordt door Breuker (1987) beschreven als "determining what will happen next in a certain situation" (blz.72). In de taak onderscheidt hij de 'system description', de beschrijving van een bepaalde situatie waarin de objekten in die situatie en de processen die op die objekten werkzaam zijn gespecificeerd worden. De oplossing van de taak bestaat uit een andere beschrijving, een 'state description', die het resultaat is van de eerder genoemde processen in de oorspronkelijke beschrijving. De inferentiestruktuur van een generische prediktie-taak ziet er als volgt uit (zie fig.9.1):



figuur 9.1: inferentie struktuur van een generische prediktie taak (uit Breuker 1987).

De metaklasse 'system description' stelt hier elk soort beschrijving van een toestand voor, zowel formeel als b.v. in common-sense bewoordingen. Deze beschrijving wordt aangevuld met kennis over de objekten en processen in de toestand, en levert de 'augmented structural description'. Deze beschrijving nu vormt de input van de KS 'transform', die de oude toestand laat overgaan in een nieuwe. Van groot belang hierbij is het 'system model', die de kennis representeert over de interaktie van processen en objekten.

Ongelukkigerwijs laat Breuker het belangrijkste element van de inferentiestruktuur, de KS 'transform', grotendeels oningevuld. Van de in deze KS gebruikte methoden zegt hij: "All kinds of methods are possible, e.g. heuristic association, means-end-analysis, constraint propagation, default reasoning" (blz.76).

Tussen de inferentiestruktuur van de generische taak 'prediktie' en die van de bridge-plannings-taak bestaan grote overeenkomsten. In het plansysteem wordt gespecificeerd: een complete beschrijving van een situatie op een bepaald moment; de default aannames over de ongeziene kaarten van de tegenstanders; en het plan, de operatoren volgens welke de ene situatie wordt omgezet in de ander. Ook hier is een systeem-model noodzakelijk, waarin gespecificeerd wordt wat de effekten van de uitvoering van een bepaald plan zijn op de bestaande toestand. Voor de invulling van de KS 'transform' in de bridge-plannings-taak verwijs ik naar blz.79-81; meer dan de constatering van een funktionele overeenkomst zit er gezien de karigheid in de beschrijving van beide 'transforms' niet in.

Hetgeen er plaats vindt in de planningstaak komt dus inderdaad overeen met een serie achter elkaar uitgevoerde predikties. In de planningstaak zijn echter de werkzame processen voor een groot deel onder controle van de planner zelf, wat in het geval van ongewenste situaties kan leiden tot ingrijpen in die processen.

Reparatie

Breuker (1987) onderscheidt twee verschillende soorten reparaties, die hij 'repair' en 'remedial' noemt. In 'repair' wordt een foutieve component van een systeem vervangen door een ander; in 'remedial' wordt de ongewenste werking van een proces tegengegaan door een ander proces in werking te stellen (b.v. door het toedienen van een medicijn). Beide taken vallen onder de klasse van 'systeem modifikatie taken', echter: "no specific models for system modification tasks have been developed yet" (blz.89). De reden die Breuker hiervoor aanvoert is, dat systeem modifikatie taken in zuivere vorm nauwelijks voorkomen in real life tasks (zie blz.25 van dit verslag): ofwel de reparatie is triviaal, en vloeit direkt voort uit een gestelde diagnose (b.v. 'transistor kapot --> transistor vervangen); ofwel de reparatie is zeer gecompliceerd, zodat de taak meer het karakter krijgt van een synthese-taak (bijvoorbeeld, het vervangen van de fundering van een huis vereist een enorme hoeveelheid planning). Aan beide taken gaat noodzakelijkerwijs vooraf de detektie van het ongewenste gedrag veroorzakende element.

In de planningstaak zijn zowel repair als remedial van toepassing, maar in principe zal bij het tegenkomen van een ongewenste situatie eerst geprobeerd worden om het bestaande plan te laten voorafgaan door een 'tegengif'; pas als dit niet mogelijk blijkt zullen er strukturele veranderingen in het plan worden aangebracht. Dit vloeit voort uit het gebrek aan 'materiaal' dat

een bridgespeler heeft: het is niet mogelijk om veranderingen in de kaarten aan te brengen. Het zou erg gemakkelijk zijn om te zeggen: 'het enige dat fout is aan dit plan, is dat ik schoppen aas te kort kom; laat ik die daarom maar lenen van een ander spel'. De spelregels laten dit helaas niet toe. In bridge moet je roeien met de riemen die je hebt; het komt dan ook zelden voor dat er in een plan één component vervangen kan worden door een funktioneel equivalent, terwijl de rest van het plan geheel in takt kan blijven. Het vervangen van een plan betekent vaak het geheel en al weggooien van het plan.

Ik kan het niet eens zijn met het gezichtspunt van Breuker, dat niet-triviale reparatie als een planningstaak op zich beschouwd moet worden. Het repareren van een systeem behoeft niet te betekenen, dat de stap tussen een niet werkend systeem en een werkend systeem in eens overbrugd wordt. Repareren betekent mijns inziens het aanbrengen van gerichte veranderingen in een systeem, veranderingen die gebaseerd zijn op het begrijpen van de werking van het systeem. Soms zijn er een reeks reparaties nodig voor het systeem het gewenste gedrag vertoont (b.v. door onvoorziene bijwerkingen); de coördinatie van die reparaties is wèl een planningstaak. De term 'reparatie' moet dus betrekking hebben op de handeling van het veranderen, en niet op de evaluatie van die verandering. De term die Clancey voor dit type taak reserveerde, 'modify', is waarschijnlijk dan ook beter geschikt dan 'repair/remedial'.

De diagnose die aan een reparatie voorafgaat is in de bridge-plannings-taak relatief eenvoudig, om twee redenen. Ten eerste is de definitie van 'ongewenst' vrij nauwkeurig vastgelegd (een situatie is ongewenst als die één van de bestaande constraints schendt); ten tweede is het gemakkelijk om de oorzaken van zo'n ongewenste situatie na te gaan. Immers, de operatoren die een vorige situatie (waar nog niets mee mis was) transformeerden tot een ongewenste, waren zelf gekozen. De operatoren die in aanmerking komen voor het veroorzaken van de ongewenste situatie zijn dus èn allemaal bekend, èn relatief klein in aantal. De diagnose (de detektie van een foutieve component) is dus in het planningsproces vrij triviaal.

9.2 Het denken van bridgers

Hetgeen dat gevonden werd in dit onderzoek komt grotendeels overeen met de bevindingen in eerder onderzoek naar het denken van denksporters in het algemeen, en bridgers in het bijzonder. Drie zaken zijn hierbij van bijzonder belang: de grote rol die herkenning speelt; het gebruik van simulatie bij het vinden van het juiste plan; en het gebruik van defaults om de simulaties zo efficient mogelijk te laten verlopen.

Dat herkenning een grote rol speelt in het denken van denksporters is een bekend gegeven in zowel schaak- bridge-, als Go-onderzoek. Dit werd in dit onderzoek bevestigd. Herkenning komt op meerdere

plaatsen in het denkproces naar voren. Allereerst worden uit de enorme hoeveelheid van mogelijke speelwijzen plausibele kandidaten geselekteerd; dit gebeurt door direkte associatie (een A V-combinatie suggereert het nemen van een snit) alsook door het gebruik van heuristieken ('kijk naar de mogelijkheid van een squeeze als je niet meer van slag mag, en je komt nog 1 slag tekort'). Deze herkenning is echter niet beperkt tot de beginfasen van het planningsproces. Ook bij de reparatie van gesimuleerde, maar niet goed bevonden plannen worden heuristieken gebruikt om de juiste reparatie zo snel en efficient mogelijk te vinden. Opvallend is de enorme flexibiliteit van de herkenning: zowel zeer concrete als zeer abstrakte beschrijvingen van een situatie kunnen als cue voor een kandidaat-oplossing dienen. Dit betekent dat een situatie op verschillende abstraktie-nivo's bij een bridger gerepresenteerd moet zijn. Dit gegeven komt goed overeen met het door Wilcox (1985) gebruikte model (zie blz.12 van dit verslag). Hierin maakt hij in de perceptie van zijn systeem onderscheid tussen 'clusters' en 'constellations'; de eerste "are general classes of entities upon which one can hang 'universal' truths"; de tweede "are special instances of clusters (..) which you immediately recognize, and about which you have special knowledge" (Wilcox 1985, blz.32).

Het gebruik van simulatie is binnen denksporten niet zo verwonderlijk. In iedere taak waarin een eenmaal ondernomen aktie niet meer teruggenomen kan worden is het voorspellen van de gevolgen van die aktie een noodzakelijke voorwaarde voor sukses. De wijze van simulatie verschilt door de aard van de taal echter van het zoeken bij schaken. De bridger moet in hoge mate struktuur aanbrengen in zijn simulaties, door het grote aantal mogelijke oplossingen, en door het grote aantal onzekere faktoren. Herkenning zorgt voor het verkleinen van de eerste faktor: de plausibelste kandidaat-plannen worden het eerst onderzocht. De tweede faktor wordt onder controle gehouden door het gebruik van default-waarden. Een plan wordt eerst getest in de vaakst voorkomende omstandigheden; pas als het plan hierin slaagt behoeft pas te worden gekeken naar de minder vaak voorkomende omstandigheden. De defaults gelden zowel distributie van de ongeziene kaarten, als het tegenspel. Interessant in deze wijze van plannen, is dat mensen in situaties met veel onzekere faktoren zichzelf tijdelijke zekerheid verschaffen door het maken van assumpties. Dit wijst erop, dat mensen dergelijke problemen niet plegen op te lossen door de probleemsituatie te beschouwen als een verzameling logische expressies waarop ze een reeks operatoren loslaten. Zij creeren daarentegen voor zichzelf een concrete situatie, en nemen die situatie als uitgangspunt voor de door hen geplande veranderingen. Dit creeren van een concrete situatie, zelfs als daar in de probleemstelling helemaal geen sprake van is, is iets dat kenmerkend lijkt te zijn voor menselijk denken. De Engelse psycholoog Phil Johnson-Laird (1983) heeft zich jarenlang bezig gehouden met de eigenaardigheden van menselijk logisch redeneren, en vond dat mensen logische problemen plegen op te lossen door het gebruik van 'mentale modellen'; zij redeneren door zo'n model te manipuleren. Bij het plannen in bridge gebeurt iets

dergelijks in extremere mate: een onzekere situatie wordt 'vertaald' naar een aantal concrete situaties die allen een bepaalde mate van waarschijnlijkheid hebben; het switchen van de ene mogelijke wereld naar de andere wordt gestuurd door de aard van de effekten van het uitgevoerde plan in zo'n mogelijke wereld. Door de uitgekiende coördinatie van het veranderen van het plan en de gemaakte assumpties is het resultaat van al deze aktiviteit een plan dat in een zo groot aantal mogelijke werelden sukses heeft.

9.3 Een kennis-systeem voor bridge-planning?

Gezien de complexiteit van de onderzochte taak, de complexiteit van de domeinkennis, en de huidige stand van zaken wat benodigde inferentiemechanismen betreft, zal de construktie van een kennissyteem met een inferentiestruktuur als getoond in fig.8.1 op zijn minst een zeer zware opgave zijn. Ervaringen op het gebied van Go bevestigen dit vermoeden: alleen al de construktie van het herkenningsgedeelte van hun Go-programma kostte Reitman en Wilcox 5 jaar arbeid (zie Wilcox 1985).

De twee gebieden waar naar ik vermoed de grootste problemen zullen rijzen zijn de ontwikkeling van een goed herkenningsmechanisme, en de construktie van dat gedeelte van het systeem dat zorgt voor het monitoren van het planningsproces: de coordinatie van de verschillende simulaties, het veranderen van de defaults, het nemen van beslissingen over het veranderen van plannen, etc. Het eerste vormt de basis van efficient plannen, en veel onderzoek is nog nodig voor een mechanisme met de kracht en flexibiliteit van menselijke herkenning geconstrueerd kan worden. Het tweede betreft de belichaming van kennis op het strategisch nivo, en de lacunes in het recente werk van b.v. Breuker (1987) doen het ergste vermoeden. Wat nodig is om die lacunes op te vullen is een manier om strategische kennis weer te geven, analoog aan b.v. de representatie van de inferentiestruktuur. Zolang een goede wijze van beschrijving ontbreekt, is enige vooruitgang op dit gebied niet te verwachten.

9.4 Evaluatie van het gebruikte model

Tot slot wil ik nog enige woorden wijden aan het gebruikte 4 lagen model van expertise zoals dat in Breuker (1987) uiteen werd gezet. De gedachte om de kennis die een mens gebruikt bij het oplossen van een taak, te beschrijven op vier verschillende nivo's bleek in dit onderzoek zeker vruchtbaar. Op diverse plaatsen in de taakanalyse komt naar voren, dat de in het model gemaakte onderscheidingen met betrekking tot de verschillende soorten kennis die experts gebruiken, goed getroffen zijn. Het indelen van menselijke kennis met behulp van de vier nivo's is echter één ding; het geven van een goede beschrijving ervan binnen die nivo's is iets heel anders.

De ambitieze presentatie van het model in Breuker (1987)

suggereert, dat met behulp van het model en de uitwerking van het model door de KADS-groep, op snelle en doeltreffende wijze tot een 'conceptueel model' gekomen kan worden, de funktionele specifikatie van een toekomstig expertsysteem. Dat bleek in dit onderzoek anders uit te pakken.

Een korte herhaling van de KADS-methodologie kan dit verduidelijken. Om het kennisacquisitieproces te versnellen wordt gebruikt gemaakt van een bibliotheek van 'interpretatiemodellen': domeinonafhankelijke beschrijvingen van de kennis, behorende bij een bepaald type taak. Deze beschrijvingen hebben betrekking op de voor zo'n type taak karakteristieke inferentie-, taak- en processtruktuur. Bij de analyse van een taak kan dan eerst in de bibliotheek een passend interpretatiemodel gekozen worden; vervolgens kan met behulp van de in het interpretatiemodel gespecificeerde 'mallen' de door de expert gebruikte kennis in de juiste vorm worden gegoten.

Duidelijk zal zijn, dat welslagen van de KADS-methodologie rust op de kwaliteit van het interpretatiemodel. De beschrijving van kennis in het interpretatiemodel moet precies de juiste graad van abstraktie hebben: concreet genoeg om te kunnen leiden tot een funktionele specifikatie van een systeem (maar niet zo concreet dat het een bepaalde implementatie impliceert), en abstrakt genoeg om geldig te blijven zijn voor de klasse taken waarvoor het model bedoeld is. En in de interpretatiemodel-bibliotheek van Breuker c.s. is dit bepaald niet altijd het geval.

De voornaamste reden voor het misplaatste enthousiasme voor de KADS-methodologie was de vele ervaring, die in de praktijk van kennissyteem-bouwen was opgedaan met *analytische taken*. Hierdoor was het mogelijk, om in het licht van de architekturen van reeds gebouwde systemen een invulling te geven aan het abstrakte idee van de vier kennisnivo's, die concreet genoeg was om mee uit de voeten te kunnen. Bij het maken van systemen voor analytische taken is KADS dan ook uitstekend te gebruiken: de bibliotheek is goed gevuld, en de interpretatiemodellen zijn specifiek genoeg. Zoals al in hoofdstuk 4 uiteen werd gezet, is de stand van zaken bij niet-analytische taken een stuk minder rooskleurig. Er bestaan nog maar weinig interpretatiemodellen voor systeem-modifikatie en -synthese taken, en de bestaande modellen zijn vaak zo slecht uitgewerkt dat ze nauwelijks als hulp kunnen dienen bij het kennisacquisitie proces.

Dit is niet verwonderlijk: er bestaan nog maar weinig systemen die dit soort taken goed op kunnen lossen, en even zo weinig formalismen. De termen die de cognitieve psychologie levert voor het beschrijven van kennis zijn bij lange na niet nauwkeurig genoeg om te leiden tot een bruikbaar interpretatiemodel voor een taak.

Het is jammer dat Breuker c.s. de lacunes in hun bibliotheek hebben opgevuld met onzorgvuldige taakbeschrijvingen. In plaats van dat een onderzoeker gestimuleerd wordt in onderzoek naar

synthese-taken door het gebrek aan kennis op dit gebied, wordt hij nu gehinderd door interpretatiemodellen die niet blijken te funktioneren.

Desondanks is het een grote verdienste van de KADS-groep dat zij het 4 lagen model van expertise van een terminologie hebben voorzien, die het mogelijk maakt om taken en kennis nauwkeuriger te beschrijven dan voorheen mogelijk was. Het verfijnen en voltooien van die terminologie zou een vooruitgang betekenen in zowel de cognitieve psychologie als in het onderzoek naar nieuwe inferentiemechanismen.

LITERATUUR

Berliner, H. (1977): Experiences in evaluation with BKG - a program that plays backgammon. Proceedings of the fifth IJCAI, blz.428-433.

Borel en Cheron (1940): Théorie mathématique du bridge. Paris: Gauthier-Villars.

Breuker, J.A. (ed.)(1987): Model-driven knowledge acquisition interpretation models. Rapport Esprit projekt 1098, Universiteit van Amsterdam.

Breuker, J.A. en B.J.Wielinga (1985): KADS: structured knowledge acquisition for expert systems. VF Memo 40, SWI/Universiteit van Amsterdam.

Charness, N. (1979): Components of skill in bridge. Canadian Journal of Psychology 33 (1), blz.1-16.

Chase, W.G. en H.A.Simon (1973): The mind's eye in chess. In Chase, W.G. (ed.): Visual information processing. New York: Academic Press.

Clancey, W.J. (1983): The epistemology of a rule-based expert system - a framework for explanation. Artificial Intelligence 20, blz.215-251.

Clancey, W.J. (1985): Heuristic classification. Artificial Intelligence 27, blz.289-350.

Davis, R. (1980): Meta-rules: reasoning about control. Artificial Intelligence 15, blz.179-222.

Groot, A.D. de (1946): Het denken van den schaker; een experimenteel-psychologische studie. Amsterdam: Noord-Holland.

Hayes-Roth, F., D.A.Waterman, en D.B.Lenat (1983): Building expert systems. Reading, MA: Addison-Wesley.

Johnson-Laird, P.N. (1983): Mental models. Cambridge: Cambridge University Press

Kearse, A.L. (ed.) (1976): The official encyclopedia of bridge (third edition). London: Benn.

Levy. D. (1981): Intelligent computer games - backgammon. Creative computing, februari, blz.182-185.

Lindelöf, E.T. (1983): COBRA - the computer designed bidding system. London: Victor Gollancz.

Miller, G.A. (1956): The magical number of seven, plus or minus two. Psychological Review 63, blz.81-97.

Napjus, C.N. (1971): Declarer; a learning bridge-playing program which generalizes and infers. Behavioral Science vol.16, blz.404-410.

Neumann, J.von en O.Morgenstern (1944): Theories of games and economic behavior. Princeton: Princeton University Press.

Newell, A. en H.A.Simon (1972): Human problem solving. Englewood Cliffs NJ: Prentice-Hall.

Nilsson, N.J. (1980): Principles of artificial intelligence. Palo Alto, CA: Tioga Publishing Company.

Quinlan, J.R. (1979): A knowledge-based system for locating missing high cards. Proceedings of the sixth IJCAI, blz.705-707.

Reitman, J.S. (1976): Skilled perception in Go: deducing memory structures from inter-response times. Cognitive pychology 8, blz.336-356.

Reitman, W. en B.Wilcox (1978): Pattern recognition and pattern-directed inference in a program for playing Go. In Waterman, D.A. en F.Hayes-Roth (eds.): Pattern directed inference systems. New York: Academic Press.

Sacerdoti, E.D. (1977): A structure for plans and behavior. Noord-Holland: Elsevier.

Simon, H.A. en K.Gilmartin (1973): A simulation of memory for chess positions. Cognitive Psychology 5, blz.29-46.

Sint, C. en T.Schipperheyn (1982): Bridge spel en tegenspel 5 - de kunst van het uitspelen. Amsterdam: Meulenhoff.

Sternberg, R.J. (ed.) (1982): Handbook of human intelligence. Cambridge: Cambridge University Press.

Wagner, D.A. en M.J.Scurrah (1971): Some characteristics of human problem solving in chess. Cognitive Psychology 2, blz.454-478.

Wielinga, B.J. en J.A.Breuker (1986): Models of expertise. Proceedings of the ECAI.

Wilkins, D.E. (1983): Using chess knowledge to reduce search. In Frey, P.W. (ed.): Chess skill in man and machine. New York: Springer.

Wilcox, B. (1985): Reflections on building two Go programs. ACM SIGART newsletter no.94, blz.29-43.

Wouden, H.G. van der (1987): Denksporten en computers; een literatuuroverzicht van schaak-, Go-, en bridge-programma's. Fakulteit sociale wetenschappen, vakgroep psychonomie, R.U.Utrecht.