

JURIDISCHE KENNISREPRESENTATIESYSTEMEN EN DE STAGE-PROP-ACTOR-NET METHODE.

Jeroen Van Nieuwenhove

Onder wetenschappelijke begeleiding
van Prof. J. DUMORTIER (K.U. Leuven)

I. INLEIDING.

A. INFORMATICA EN RECHT.

Voor de jurist is de informatica de laatste jaren een belangrijk hulpmiddel geworden. Na de tekstverwerker kwamen de databanken, waarmee rechtspraak, rechtsleer en wetgeving rond een bepaald onderwerp opgezocht kunnen worden. Hier blijft het echter niet bij. Stilaan worden systemen ontworpen om de rechtsregel tot leven te wekken, om hem na te bootsen met een computer. Deze simulatietechniek wordt *juridische kennisrepresentatie*¹ genoemd. De rechtsregel wordt door een computer gehanteerd voor diverse doeleinden. Zo kan men juridische kennisrepresentatie gebruiken om concrete vragen van leken te beantwoorden (bv. opzeggingsvergoeding), om advies te geven in verband met het wetgevend beleid door aan te duiden wat de gevolgen zullen zijn van bepaalde maatregelen (bv. belastingwetgeving²), voor het overnemen van routinetaken van de juristen, enz...

Juridische kennisrepresentatie bestaat in verschillende gradaties, van heel eenvoudige berekeningssystemen (bv. ontslagvergoeding) tot zeer complexe systemen. Hier worden eerst de eenvoudige beslissingsschema's behandeld, vervolgens de logische systemen en tenslotte de object-georiënteerde systemen.

(1) Men zou ook de term *artificiële intelligentie (A.I.)* kunnen gebruiken, maar deze benaming roept teveel ongewenste associaties op. Andere benamingen zijn *expertsystemen* en *knowledge based systems (K.B.S.)*.

(2) "With the computerprograms it will be easy to see the effect of proposed changes in the tax and social security legislation on various households." STRIPINIS, D., "Integrated Marginal Tax Rates and Government Planning", in *Artificial Intelligence and Legal Information Systems*, CIAMPI, C. (ed.), Amsterdam, North-Holland Publishing Company, 1982, 381-387.

B. DE STAGE-PROP-ACTOR-NET METHODE.

In 1988 startte aan de rechtsfaculteit van de K.U. Leuven een onderzoek onder leiding van Prof. J. DUMORTIER ³ en J. GOOSSENAERTS ⁴ naar de toepassing van de *stage-prop-actor-net methode* in de juridische kennis-representatie ⁵. Deze methode en de bijhorende programmeertaal TIE was aanvankelijk door J. GOOSSENAERTS ⁶ ontwikkeld voor de beschrijving van organisaties, na een voorafgaand onderzoek onder leiding van Prof. LEWI ⁷.

II. BESLISSINGSSCHEMA'S.

Een manier om de juridische kennis te automatiseren is het opstellen van een beslissingsschema waarbij de rechtsregel a.h.w. in een keurslijf wordt gegoten. Dit keurslijf ziet er als volgt uit :

ALS voorwaarde DAN beslissing

Een iets ingewikkelder systeem hanteert meerdere voorwaarden en meerdere beslissingen :

*ALS voorwaarde 1 EN voorwaarde 2 OF NIET voorwaarde 3
DAN beslissing 1 EN beslissing 2*

Het beslissingsschema gaat na of de voorwaarde(n) vervuld zijn en trekt hieruit de nodige conclusies. Een programma dat opzegvergoedingen in het arbeidsrecht berekent, kan na opgeven van een aantal gegevens (leeftijd, loon, anciënniteit,...) een schatting van de opzegvergoeding berekenen. Dit systeem is ook voor de leek toegankelijk, omdat het zeer concrete vragen stelt en zeer concrete antwoorden geeft.

(3) Centrum voor Juridische Informatica, Faculteit Rechtsgeleerdheid, K.U. Leuven

(4) Departement Computerwetenschappen, Faculteit der Toegepaste Wetenschappen, K.U. Leuven

(5) GOOSSENAERTS, J., LEWI, J. en DUMORTIER, J., *A Stage-prop-actor Approach to Legal Representation*, Report CW 78, oktober 1988.

(6) GOOSSENAERTS, J., *Tie: preliminary version (October 1989)*, 123 p.

(7) Departement Computerwetenschappen, Faculteit der Toegepaste Wetenschappen, K.U. Leuven

III. LOGISCHE SYSTEMEN.

A. BEGRIP.

De toepassing van *logica* in het recht is een voor de hand liggende werkwijze voor juridische kennisrepresentatiesystemen ⁸. Het uitgangspunt van deze benadering is dat het recht bestaat uit een min of meer logisch corpus van regels, die ontleed kunnen worden met de gangbare technieken van de *logica*.

Bij de beslissingsschema's werd de rechtsregel vertaald in een reeks voorwaarden aan de hand waarvan een beslissing werd genomen. Bij de logische systemen wordt met de rechtsregels als dusdanig gewerkt en worden deducties en interferenties gemaakt. De logische systemen staan daarom op een veel hoger niveau dan de gewone beslissingsschema's.

B. LOGISCHE SYSTEMEN EN DE RECHTSREGEL.

Binnen de rechtsleer bestaat er een klassieke theorie over de structuur en de opbouw van de rechtsregel. Een rechtsregel bestaat uit rechtsfeiten die de toepassingsvoorwaarden vormen voor de rechtsgevolgen in het beschikkend gedeelte. De corresponderende feiten in de levensfeitelijke omgeving die aan de toepassingsvoorwaarden voldoen, worden rechtsfeiten en door het beschikkend gedeelte van de rechtsregel worden er rechtsgevolgen aan verbonden.

Sommige rechtsvoorschriften zijn complexer dan de rest, maar dat stelt in wezen geen probleem voor de verwoording van deze rechtsregels in een dergelijk schema :

1. Sommige rechtsgevolgen creëren rechtsfeiten die op zich weer beantwoorden aan de toepassingsvoorwaarden voor andere rechtsregels. Op die manier krijgt men ketens van rechtsregels, die vaak van één rechtstak naar een andere verwijzen.

2. De toepassingsvoorwaarde is soms verzwegen of wordt in een ander wetsartikel geformuleerd. Dit is hooguit een interpretatieprobleem. Hetzelfde geldt voor het beschikkend gedeelte (het rechtsgevolg).

(8) VON WRIGHT, G.H., "Norms, Truth and Logic", in *Deontic Logic, Computational Linguistics and Legal Information Systems*, II, MARTINO, A.A. (ed.), Amsterdam, North-Holland Publishing Company, 1982, 3-20.

C. EEN VOORBEELD.

Met een taal als bijvoorbeeld *Prolog* kan men de structuur van de rechtsregel herschrijven in programmacode ⁹. Zo kan men de aquiliaanse aansprakelijkheid (art. 1382 B.W.) in twee stappen in *Prolog* omschrijven :

Art. 1382 B.W.: "Elke daad van de mens, waardoor aan een ander schade wordt veroorzaakt, verplicht degene door wiens schuld de schade is ontstaan, deze te vergoeden."

Men kan in deze regel drie toepassingsvoorwaarden herkennen : *primo*, de daad van een mens, *secundo*, de schade van een ander, *tertio*, het oorzakelijk verband tussen de twee. Het rechtsgevolg is de vergoeding van de schade van de andere. Dit schema is het resultaat van de ontleding van de rechtsregel.

Prolog is een programmeertaal die gericht is op het verzamelen van een reeks gegevens (*a database of facts*) die bestaan uit een reeks relaties tussen verschillende entiteiten. Bijvoorbeeld : Jan studeert rechten, Mieke studeert rechten, Jan lust tomatensoep, enz... Wanneer voldoende gegevens bekend zijn, kan men bepaalde vraagstellingen formuleren, zoals bijvoorbeeld : Wie lust tomatensoep en studeert rechten ? *Prolog* zal in elk geval Jan aanduiden.

Toegepast op ons voorbeeld creëert men in *Prolog* de volgende relaties:

steltDaad(symbol, symbol)
lijdtSchade(symbol, symbol)
oorzakelijkVerband(symbol, symbol)
vergoedtSchade(symbol, symbol)

Dit is de manier waarop *Prolog* relaties definieert : *steltDaad* is een relatie tussen twee symbolen, entiteiten en nu kan deze relatie gelegd worden tussen de bestaande entiteiten, zodat we feiten krijgen. Wanneer we zeggen dat Mieke een ruit ingooit kunnen we dit in de pasvorm van deze relatie gieten :

steltDaad(Mieke, RuitIngooien)
Mieke stelt een daad, nl. het ingooien van een ruit.

lijdtSchade(Jan, Ruit)
Jan lijdt schade, want zijn ruit ligt aan scherven.

oorzakelijkVerband(RuitIngooien, Ruit)
Er bestaat een oorzakelijk verband tussen het ingooien van de ruit en de beschadigde ruit

(9) Voor een inleiding tot *Prolog* : CLOCKSIN, W.F. en MELLISH, Ch.S., *Programming in Prolog*, Berlijn, Springer-Verlag, 1981, 279 p.; SUSSKIND heeft een uitgebreide formalisatie van een deel van het Schotse echtscheidingsrecht uitgewerkt. SUSSKIND, R.E., *Expert Systems in Law*, Oxford, Clarendon Press, 1987, 262-269.

Hierdoor hebben we een reeks *feiten* opgegeven, die eigenlijk bestaan uit het invullen van concrete elementen in de bestaande relaties.

Nu kan de *rechtsregel* beschreven worden als volgt :

vergoedtSchade(*schadeverwekker, gelaedeerde*)
if steltDaad(*schadeverwekker, X*) and
lijdtSchade(*gelaedeerde, Y*) and
oorzakelijkVerband(*X, Y*).

Dit is eigenlijk de verwoording van art. 1382 B.W. in Prolog : wanneer de schadeverwekker daad X stelt en de gelaedeerde schade Y lijdt en een oorzakelijk verband bestaat tussen daad X en schade Y, dan moet de schadeverwekker de gelaedeerde vergoeden.

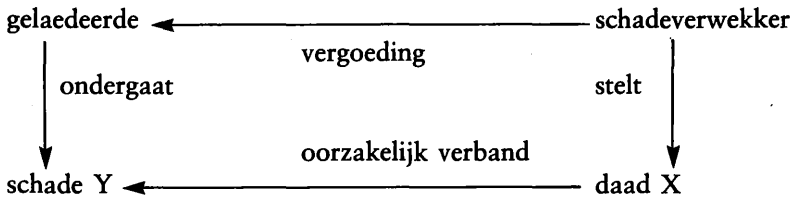


Fig. 1 : De relaties tussen de elementen.

Prolog zal uit de gegeven feiten (*facts*) m.b.t. Jan en Mieke afleiden dat :

vergoedtSchade(*Mieke, Jan*)

Elke keer dat er voldoende feiten bestaan kan Prolog besluiten tot het bestaan van een plicht tot schadevergoeding. Uiteraard is dit een formalisatie die voorbijgaat aan de jurisprudentiële verfijningen van de regels over de noodzakelijke onrechtmatigheid van de daad en andere vragen zoals de omvang van de vergoeding, bepaling van oorzakelijk verband, tussenkomende factoren,... De logische kennisrepresentatiesystemen gaan er echter van uit dat ook de rechtspraak in regels te gieten is en geformaliseerd kan worden (zie *infra*).

D. PROBLEMEN BIJ DE LOGISCHE AANPAK.

Vooraf in de discipline van de rechtsvergelijking wijst men op het onderscheid tussen "*law in the books*" en "*law in action*". Bij het beschrijven van het recht in regel-gebaseerde systemen loopt men een dubbel gevaar : enerzijds, voorbijgaan aan de andere bronnen dan de wetgeving (1) ; anderzijds, voorbijgaan aan de werkelijkheid (2).

(1) Naast de wetgeving bestaan er nog **andere rechtsbronnen** die voor veel rechtsregels van uiterst groot belang zijn. Voor art. 1382 B.W. is de bijdrage van de rechtspraak van ontzettend groot belang, vooral in verband met het concretiseren van de drie bekende toepassingsvoorwaarden. Dit probleem kan men in regel-gebaseerde systemen opvangen door de *ratio decidendi* van een belangrijk arrest om te zetten in een rechtsregel en in het bestaand schema te verwerken. Op die manier wordt een *prima facie* eenvoudig wetsartikel een zeer complex stuk code ¹⁰.

Daarnaast is er ook de invloed van de gebruiken (die soms *contra legem* zijn), de standaardbedingen in toetredingscontracten en zelfs de gebeurtenissen die aan het recht ontsnappen. Een moord is pas een moord als het feit wordt vastgesteld door een bevoegd persoon. In het voorbeeld van art. 1382 B.W. kan men de *small claims* vermelden. Vaak is de schade zo gering dat er geen vergoeding wordt geëist, soms wordt de zaak in der minne geregeld en soms primeren machts- of vriendschapsverhoudingen op schadeclaims.

Al deze gegevens zijn stoorzenders voor een legislatief perfecte oplossing. We gaan er echter van uit dat we de juridische werkelijkheid zo nauwgezet mogelijk willen beschrijven. Vanuit die doelstelling moet men alle pogingen tot juridische kennisrepresentatie evalueren. Voor sommige rechtsdomeinen levert een puur legislatieve benadering een sterk benaderend resultaat op, dat in vele gevallen voldoet wanneer men rekening houdt met het feit dat bij een verdere verfijning de marginale kosten sterk toenemen. In sommige gevallen heeft men geen behoefte aan een perfecte overeenstemming met de werkelijkheid (wat overigens steeds een streefdoel blijft) en volstaat de relatief eenvoudige en goedkope oplossing van een puur legislatieve beschrijving. Men denke hierbij vooral aan meer technische rechtstakken zoals fiscaal recht, sociaal recht, procedure, enz...

In andere rechtsdomeinen levert een dergelijke werkwijze een hoogst onbetrouwbaar resultaat op. Het voorbeeld van art. 1382 maakt dit zeer duidelijk.

(2) Bovendien bestaat er ook een soort *interferentie* tussen verschillende rechtsregels die duidelijk wordt in de werkelijkheid. De gelaedeerde in het voorbeeld van art. 1382 moet ook nog *bewijzen* dat de andere een fout heeft gemaakt, enz... Zolang hij dat niet heeft gedaan, krijgt hij uiteraard geen schadevergoeding (tenzij misschien een regeling werd getroffen met de tegenpartij). Men zou dus de toepassingsvoorwaarde(n) van elk wetsartikel moeten omzetten in "van zodra de eiser in rechte bewijst dat...". Dan nog wordt er nog geen rekening gehouden met de vermoedens (al dan niet weerlegbaar) en met o.m. de rol van de rechter die niet louter passief moet blijven

(10) "Moreover, the addition of more and more items of knowledge to the system may lead to the overall operation of the system becoming incomprehensible to the user, even though each item is comprehensible in itself." BRAMER, M.A., "A Survey and Critical Review of Expert Systems Research", in *Introductory Readings in Expert Systems*, MICHIE, D. (ed.), Gordon and Breach Science Publications, 1982, 23.

tegenover de partijen die het geschil komen aandragen, maar bijvoorbeeld mag bevelen dat bepaalde bewijsstukken voorgelegd moeten worden. De verzachtingen en amendementen op de regel van de bewijslast behoren gedeeltelijk tot de wetgeving (vooral het Gerechtelijk Wetboek) en gedeeltelijk tot de rechtspraak ¹¹.

Volgens de regel-gebaseerde methode moeten deze "complicaties" vertaald worden in rechtsregels en vervolgens omgezet worden in code en ingepast worden in het bestaande schema. Een grotere nauwkeurigheid vereist dus niet iets anders, maar eenvoudig meer van hetzelfde ¹². Overigens zal men bij de implementatie op bestaande computersystemen na een tijdje tegen de fysische grenzen van het systeem stoten (geheugen, snelheid,...), maar dit doet geen afbreuk aan de theoretische haalbaarheid van deze methode. Bij sommige problemen zal men vrij snel het gewenste resultaat bereiken; bij andere problemen, die bijvoorbeeld een groter rechtsdomein betreffen, zal men wellicht nooit een resultaat bereiken (met economisch verantwoorde inspanningen).

IV. OBJECT-GEORIENTEERDE SYSTEMEN.

A. BEGRIP.

De object-georiënteerde methode (*object-oriented programming systems* of *O.O.P.S.*) is ontstaan in de jaren '60 en '70 ¹³. Het kwam neer op een alternatief voor de gangbare denkwijze in programmeertechnieken, die alles bekeek als instructies, uitdrukkingen en stuurinstructies. De voorstanders van de object-georiënteerde systemen betogen dat de werkelijkheid niet moet beschreven worden als groepen van regels, maar als objecten, elementen, entiteiten, atomen, met onderlinge relaties. Er zijn elementen die, een bepaalde daad verrichten waardoor andere elementen beïnvloed worden (bv. het berokkenen van schade). De werkelijkheid is een *interactie van entiteiten* (althans volgens de klassieke natuurkundige inzichten).

In de object-georiënteerde talen wordt dan ook gewerkt met beschrijvingen van *objecten*, die een bepaalde betekenis hebben in het systeem dat men beschrijft. Wanneer men de beschrijving uitbreidt, moet men niet alles opnieuw beginnen, maar kan men de bestaande beschrijving grotendeels overnemen en aanpassen, uitbreiden naar de nieuwe kenmerken die het krijgt. Wanneer men een simulatie maakt van een huurovereenkomst en men wenst het systeem uit te breiden naar een handelshuurovereenkomst,

(11) Bv. de diverse vermoedens in art. 1384 B.W.

(12) Satinder P. GILL: "Taylorism is the extreme fragmentation of work and skills. Tasks are broken down into simple, explicitly and scientifically defined units." in GILL, S.P., "On two A.I. traditions", *A.I. & Society*, 1988, (321), 322.

(13) Een inleiding tot object-georiënteerde systemen: THOMAS, D., e.a., "Object-oriented Programming", *Byte*, 1989, afl. 3, 228-271.

kan men de beschrijvingen van de huurder en de verhuurder gewoon overnemen en aanpassen. Deze werkwijze noemt men *overerving of inheritance*.

De object-georiënteerde talen gaan uit van een groep objecten die met elkaar kunnen communiceren door middel van berichten die ze aan elkaar doorgeven. De toepassing wordt eigenlijk ontleed in entiteiten verbonden door relaties. Een bedrijf wordt ontleed in verschillende afdelingen, die op hun beurt bestaan uit verschillende leden. Tussen alle objecten bestaan verbindingen, bijvoorbeeld in de vorm van verslagen van afdelingen aan het management. Het is duidelijk dat men gedwongen wordt de interne organisatie te gaan ontleden wanneer men object-georiënteerde talen gaat gebruiken. Deze benadering is wellicht voor de mens een meer natuurlijke denkwijze dan het sequentieel instructie-denken van de regelgebaseerde talen.

B. TECHNISCHE KENMERKEN.

Ten eerste is er het contrast met de gangbare programmeertalen, die uitgaan van actieve procedures, instructies op passieve data. Object-georiënteerde talen gaan uit van een object dat niet louter passief is, maar een combinatie vormt van toestanden (*states*) en procedures (*methods*) die de toestanden kunnen wijzigen. Zo'n procedure kan opgestart worden met een bericht aan het object in kwestie. Belangrijk is dat deze procedures eigen aan het object zijn, en geen globale begrippen. Daarom kunnen verschillende objecten gelijknamige procedures hebben die toch in wezen verschillen. Het is een beetje zoals studenten en assistenten, die allebei boeken kunnen uitlenen in de bibliotheek, maar met verschillende modaliteiten (bv. aantal boeken dat men mag meenemen).

Het binden van toestanden en procedures die het object beïnvloeden noemt men *data abstraction*. De gebruikers moeten enkel de *berichten* kennen die ze aan een object moeten doorgeven om het te bewerken. Het is mogelijk om de interne structuur van het object te wijzigen zonder dat de gebruiker nieuwe berichten moet leren.

Ten tweede is er het mechanisme van *overerving of inheritance* dat hierboven al werd beschreven. Als men een nieuw object wil toevoegen dat enkel op een paar punten verschilt van een bestaand (beschreven) object, kan men de kenmerken van het bestaand object laten overerven naar het nieuwe object en de nodige aanpassingen maken. Op die manier kan men de code opnieuw gebruiken en tegelijkertijd de overeenkomsten en verschillen duidelijk maken.

Een derde kenmerk is het onderscheid tussen *class* en *instance*. *Classes* zijn groepen objecten die gelijkaardig zijn en in principe dezelfde bevoegdheden en verantwoordelijkheden hebben. Voorbeelden hiervan zijn navorser van een laboratorium, parlamentsleden, lopers in een estafetteploeg. *Instances* zijn dan de individuele elementen van de class: een navorser, een

parlementslid, een estafetteeloper. Het is te vergelijken met het verschil tussen de *type*- en de *variable-declaration* in hogere programmeertalen. Een *class* definitie krijgt toestanden en procedures mee die er eigen aan zijn, en in een later stadium worden er *instances* van afgeleid die allemaal die eigenschappen van de *class* bezitten en toch onderling verschillen.

Op die manier kan men twee tegenstrijdige belangen verzoenen. Men hoeft geen beschrijving te geven van elk parlementslid afzonderlijk (d.w.z. elk een afzonderlijke *stage*) en men kan elk lid apart adresseren.

Ten vierde varieert het doorgeven van berichten (*messages*) naargelang het gaat om **publieke of private berichten**. Publieke berichten kunnen door iedereen opgestart worden, terwijl private berichten enkel door het object kunnen worden gebruikt. In de SPAN-methode wordt deze techniek nog verder verfijnd met behulp van de *links*.

V. DE STAGE-PROP-ACTOR-NET METHODE.

A. INLEIDING.

De *stage-prop-actor-net* (SPAN) methode heeft twee belangrijke kenmerken ¹⁴: het gaat om een *object-georiënteerde* methode en er is een *differentiatie* tussen de verschillende soorten objecten. Wat die differentiatie precies inhoudt, wordt later toegelicht. In de werkelijkheid kan men niet zomaar van entiteiten spreken. Er bestaan verschillende soorten entiteiten. Zo zijn een natuurlijke persoon en een stoel twee objecten die wezenlijk verschillend zijn. In de SPAN-methode wordt geopteerd voor een bepaalde classificatie van deze objecten.

De object-georiënteerde aanpak van de SPAN-methode betekent dat er *objecten* zijn die elkaar beïnvloeden. De objecten zijn *statische* gegevens. Wanneer zij elkaar beïnvloeden, kunnen ze elkaars toestand veranderen. Wanneer een man en een vrouw zich aanbieden bij de gemachtigde ambtenaar, kunnen zij een huwelijk sluiten. De interactie tussen de ambtenaar en het echtpaar *in spe* verandert dus hun rechtstoestand. De ambtenaar en de man en vrouw zijn de objecten, het al dan niet gehuwd zijn is de toestand van man en vrouw, het sluiten van het huwelijk is het *dynamische* gebeuren dat de toestand (het statische element) verandert.

Een beschrijving van de SPAN-methode valt dus uiteen in de beschrijving van de soorten objecten die er bestaan en hun mogelijke toestanden, ofwel de statische elementen, en de beschrijving van de gebeurtenissen die vereist zijn opdat objecten elkaar van toestand kunnen laten veranderen, ofwel de dynamische elementen.

(14) GOOSSENAERTS, J., *o.c.*, 20.

B. DE STATISCHE GEGEVENS.

1. De soorten objecten.

De objecten die gebruikt worden in de SPAN-methode worden ook wel de *calculi* genoemd. Er bestaan drie soorten van *calculi* of objecten in de SPAN-methode: *stages*, *props* en *actors*. Ze staan respectievelijk voor bevoegdheid (*authorization*), gegevens (*data*) en gebeurtenissen (*processes*)^{15 16}.

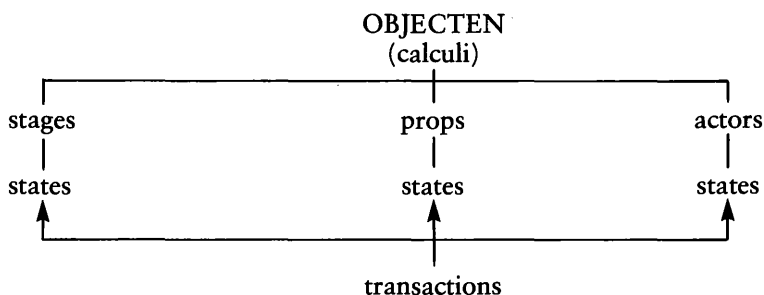


Fig. 2: De *calculi* in de SPAN-methode.

a. De stages.

De belangrijkste *calculus* in de SPAN-methode is de *stage*. Een stage is een plaats, maar men moet dit niet letterlijk opvatten. De nadruk ligt op de bevoegdheden en de verantwoordelijkheden¹⁷ die eraan verbonden zijn. Een voorbeeld: iemand is ziek en gaat naar een geneesheer, die zijn praktijk op een bepaald adres heeft. Als de geneesheer verhuist, verhuist zijn praktijk met hem. De bevoegdheden en de verantwoordelijkheden die aan de dokterspraktijk verbonden zijn, blijken richtinggevend. Voorbeelden van stages zijn een advocatenkantoor, een laboratorium, een parlement.

(15) GOOSSENAERTS, J., VAN NIEUWENHOVE, J., DUMORTIER, J. en LEWI, J., *Stage-prop-actor Nets as Legal Knowledge Systems*, 3.

(16) Ter verduidelijking herinneren we er aan dat de vertaling van *stage*, *prop*, *actor*, (en *net*) respectievelijk "podium", "rekwisiet", "acteur" en ("spel") is. Deze vergelijking met de theatrale wereld nag echter niet te ver doorgetrokken worden. Een *stage* is niet steeds een plaats en een *actor* niet per definitie een plaats.

(17) HOHFELD, W.N., "Fundamental Legal Conceptions as Applied in Judicial Reasoning", *Yale Law Journal*, 1913, 16-59.

Belangrijk is dat de stages een groepering van rechten en plichten vormen en dus een aangrijpingspunt vormen voor het recht¹⁸. Men kan verdedigen dat een beschrijving van stages grosso modo neerkomt op een beschrijving van natuurlijke personen en rechtspersonen. Weliswaar is het een *contextgevoelige* beschrijving. Niet de natuurlijke persoon als geheel wordt beschreven, maar wel de natuurlijke persoon in een bepaalde situatie (bv. de natuurlijke persoon als dokter). Uit alle informatie die we kunnen vergaren rond een rechtssubject wordt dus enkel een bepaald fragment bestudeerd.

Een interessante analogie kan men vinden in de sociologie¹⁹. Een fundamentele mogelijkheidsvoorwaarde voor een normale samenleving ligt in het beperken van de mogelijkheden die men heeft. Auditoria zijn zo ingericht dat men er niet kan gaan tennissen. Stages groeperen bepaalde mogelijkheden, maar sluiten andere mogelijkheden uit en zijn dan ook gedragsrichtende situaties. Wat betreft de stage als groepering van bevoegdheden en verantwoordelijkheden kan men inspiratie vinden bij de theorie van de *rolpatronen*. De rolpatronen zijn quasi-objectieve, principieel van het individu onafhankelijke complexen van gedragsvoorschriften. Deze sociale rollen worden door de maatschappij omschreven, aangeleerd (socialisering) en gesanctioneerd. Belangrijk is dat men verschillende rollen kan uitoefenen, die o.m. met mekaar strijdig kunnen zijn (rollenconflict). Eén persoon kan verschillende stages bezetten. Een geneesheer heeft zijn praktijk, maar is eventueel ook huisvader, lid van een rederijderskamer, lid van de orde van geneesheren, enz... Als rechtssubject is hij actief in verschillende situaties, maar één situatie wordt eruit gelicht en beschreven, nl. dat rechtssubject als geneesheer²⁰.

b. De props.

De *props* zijn de passieve objecten in het systeem dat men beschrijft: bijvoorbeeld een steen, een gebouw, een dossier, een rekening-courant, enz... In principe ondernemen zij zelf niets, maar ondergaan zij de activiteiten van andere objecten.

c. De actors.

De *actors* zijn de actieve calculi binnen het systeem. Zij verrichten acties waardoor de situatie van de omringende wereld gewijzigd wordt. Er zijn twee soorten actors. Enerzijds zijn er de menselijke actors, die op een niet

(18) Over het gebruik van de rechtspersoonstechniek: BROEKMAN, J.M., *Mens en mensbeeld van ons recht*, Leuven, Acco, 1986, 33-88.

(19) VAN DOORN, J.A.A. en LAMMERS, C.J., *Moderne sociologie. Systematiek en analyse*, Utrecht, Spectrum, 1959, 104-112.

(20) Deze aflijning kan problemen meebrengen die men toepasselijk *context cutting up problems* noemt.

zuiver voorspelbare manier handelingen kunnen stellen. Voor hun beslissingen zullen zij informatie en dialoog nodig hebben, die door het model kunnen verstrekt worden via andere calculi. Anderzijds zijn er de programma's, die voorspelbare en routine-taken verrichten. Deze taken zullen dan ook in principe voordien nauwkeurig moeten omschreven worden.

2. De toestanden van objecten.

Elk object kan zich een aantal mogelijke toestanden (*states*) bevinden. Een auto kan in aanbouw zijn, kan stilstaan, rijden of op het autokerkhof liggen. Een vennootschap kan opgericht worden, actief zijn of ontbonden worden. Hoe meer men een vennootschap bestudeert, hoe meer toestanden men kan ontdekken.

We gaan er van uit dat een bepaald object op een bepaald moment één en slechts één toestand kan aannemen. Een object kan wel van toestand veranderen, maar die verandering van toestand (*state change*) is onderworpen aan bepaalde voorwaarden. Een meer uitgewerkt voorbeeld kan men vinden in de verschillende (rechts)toestanden van een echtgenoot (of een echtgenote). De echtgenoot "ontstaat" als echtgenoot wanneer hij huwt. Daardoor komt hij in de "gewone" toestand. Andere toestanden zijn bijvoorbeeld een hangende echtscheidingsprocedure, een feitelijk scheiding en dringende, voorlopige maatregelen. Uiteraard is de echtgenoot dan nog steeds gehuwd, maar het huwelijk is in een kritische fase getreden en het is mogelijk om ondanks het huwelijk tamelijk verre gaande maatregelen te nemen t.a.v. de echtgenoot. Daarom beschouwen we deze regelingen beter als aparte toestanden.

stage Echtgenoot

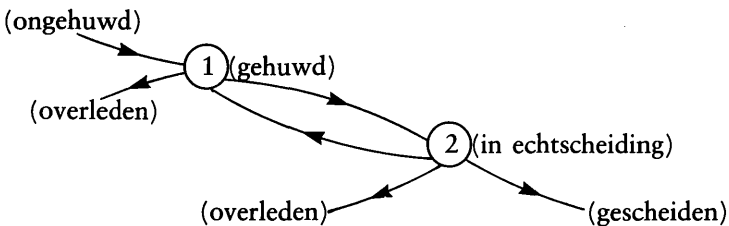


Fig. 3: De states van een echtgenoot (vereenvoudigd).

Merk op dat de toestanden ongehuwd, overleden en gescheiden geen echte toestanden zijn, maar vertrek- en eindpunten van respectievelijk in- en uitgaande pijlen. Het zijn voorlopige benamingen, omdat de problematiek van het ontstaan en verdwijnen van objecten later gepreciseerd zal worden.

3. De links tussen de objecten.

De toestandsveranderingen van de objecten zijn gebonden aan bepaalde voorwaarden. De verschillende objecten kunnen mekaar van toestand doen veranderen, maar niet onbeperkt. Een voldoende aantal mensen kan een vennootschap ontbinden, indien ze bepaalde formaliteiten vervullen en aan bepaalde voorwaarden voldoen. Zij veranderen gezamenlijk de toestand van de vennootschap omdat ze daartoe de *bevoegdheid* hebben. Die bevoegdheid kan men tegelijkertijd zien als een soort van *onderworpenheid* van de vennootschap aan de vennoten. Aan de ene kant heeft men een bevoegdheid, aan de andere kant een ondergeschiktheid.

Het eigenlijke veranderen van de toestand is een actief gebeuren, dat behoort tot het dynamische gedeelte van de beschrijving.

C. DE DYNAMISCHE GEGEVENS: DE TRANSACTIONS.

Wanneer een object de bevoegdheid heeft om een ander object van toestand te laten veranderen en het andere object is ondergeschikt (in technische zin) aan het eerste object, dan kan het de toestand van het andere object veranderen door een *procedure* te starten. Deze procedure bevat de materiële en formele voorwaarden en verandert de toestand. Wanneer men met een auto wil rijden, moet men bevoegd zijn om met die auto te rijden en moet de auto (onder)geschikt zijn om mee te rijden. Het rijden met de auto is dan een procedure die bepaalde voorwaarden stelt (niet dronken zijn, rijbewijs bezitten,...) en die de toestand van de auto verandert van stilstaand in rijdend.

De *transactions* zijn de gebeurtenissen die onder bepaalde voorwaarden toelaten dat een object of calculus van toestand verandert. Wanneer een lezer in de bibliotheek een boek uitleent, laat hij dit door de bediende aan de balie registreren. De bediende kijkt na of voldaan is aan bepaalde voorwaarden (lezer is ingeschreven, boek is uitleenbaar en niet gereserveerd) en leent het boek uit. In de SPAN-methode starten de lezer en de bediende samen een transaction op die de toestand (state) van het boek verandert, van "beschikbaar" naar "uitgeleend". De bediende is hiertoe bevoegd, maar eerst moet hij een aantal voorwaarden nagaan. Dan begint de transactie, die men *uitleenRegistratie* zou kunnen noemen ²¹. Als een spiegelbeeld van de uitleenRegistratie is er de transaction *teruggaveRegistratie*, waarbij de

(21) GOOSSENAERTS, J., o.c., 68.

lezer en de bediende de toestand van het boek weer veranderen : van "uitgeleend" naar "beschikbaar".

De inhoud van de transaction *uitleenRegistratie* ; is een kwestie van programmeertechniek. Het nummer van het boek wordt ingevoerd en de uitleenperiode wordt bepaald, samen met het nummer van de lezer. Nu is het mogelijk om met een andere, regelmatig uitgevoerde transaction na te gaan of er al lezers zijn die hun boek op die dag moesten binnenbrengen, zodat hen een aanmaning kan worden gestuurd ²².

In het voorbeeld van de echtgenoot zal de overgang tussen de verschillende rechtstoestanden mogelijk zijn dankzij bepaalde procedures. Om van de "normale" toestand naar de toestand te gaan waarin de echtscheidingsprocedure loopt, moet men uiteraard de echtscheiding aanvragen. Men kan ook vanuit de toestand van feitelijke scheiding naar die toestand en wel met (in principe) dezelfde transaction. Uiteraard bestaan er verschillende soorten van echtscheidingsprocedures, waarbij het mogelijk is om elke soort aan een eigen *state* van de echtgenoot te verbinden en een aangepaste transaction mee te geven.

stage Echtgenoot

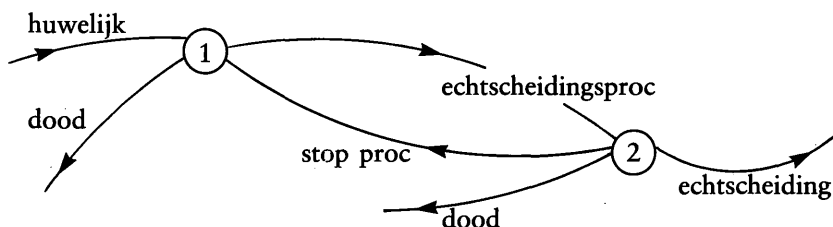


Fig. 4: De states en transactions van een echtgenoot (vereenvoudigd).

Merk op dat de transactions huwelijk, dood en echtscheiding in- of uitgaande pijlen zijn die het object echtgenoot doen ontstaan of vervallen. Zij vertrekken uit het "niets" of verdwijnen in het "niets" en zijn dus een bijzondere soort transactions omdat ze niet van een toestand naar een andere gaan. Wanneer de echtgenoot overlijdt, is zijn "rol" als echtgenoot uitgespeeld.

(22) GOOSSENAERTS, J., o.c., 56-57.

D. DE IMPLEMENTATIE.

1. Het netwerk.

Het doel van de SPAN-methode is de gebruiker zoveel mogelijk bij de ontwikkeling te betrekken en de nood aan technische kennis te minimaliseren ²³. Het is de bedoeling om de gebruiker van het systeem te begeleiden d.m.v. een *editor*, een programma dat zoveel mogelijk werk overneemt van de gebruiker en hem wijst op fouten en inconsistenties in zijn beschrijvingen. De transactions worden uitgevoerd door een *interpreter*, een programma dat de instructies van de transaction omzet naar machinetaal die voor de computer begrijpbaar is.

Het is mogelijk om tijdens de werking van het systeem met zgn. *query procedures* ²⁴ informatie te krijgen over het systeem, net alsof het een databank betreft. Zo is het mogelijk om in een systeem dat de werking van een bibliotheek simuleert, informatie te krijgen over bepaalde boeken : de titel, de auteur, de uitleenstatus, enz.

2. De communicatie met de gebruiker.

In het ontwerp van een expertsysteem moet voldoende aandacht uitgaan naar de omgang van het systeem met de gebruiker. Het praktische nut van een expertsysteem staat of valt met de kwaliteit van deze *user interface*.

Men kan verschillende wegen bewandelen in het ontwerpen van een user interface. De meest rechtlijnige manier is de directe communicatie van het systeem met de gebruiker door volzinnen in natuurlijke talen uit te wisselen ²⁵. In dit *open* systeem rijzen verschillende problemen : taalfouten van de gebruiker, beperkingen in de woordenschat van het systeem, analyse van de zin in interpreteerbare onderdelen, het omzetten van omschrijvingen en synoniemen naar de begrippen die het systeem gebruikt, enz...

Omwille van deze praktische problemen opteert men vaak voor een onderdeel (*subset*) van de taal van de gebruiker, die dan enkel bepaalde woorden op een bepaalde manier mag gebruiken. Hij mag bijvoorbeeld niet meer opgeven "Wij willen samen een Naamloze Vennootschap oprichten", maar kan enkel opgeven "RichtOp N.V.". Wanneer deze beperking door de gebruiker overschreden wordt, stuit hij op foutmeldingen van het systeem. Het voordeel van deze werkwijze is dat de kosten van de user interface lager liggen, het nadeel is dat men eerst de beperkingen moet aanleren en dat in sommige gevallen (wanneer de *subset* beperkt is tot een technisch jargon) de meeste mensen niet kunnen werken met het systeem.

(23) GOOSSENAERTS, J., o.c., 25-30 en 77-78.

(24) GOOSSENAERTS, J., o.c., 58-62.

(25) SUSSKIND, R.E., o.c., 64-66.

Een tweede optie is een *gesloten* systeem waarbij de gebruiker wordt geconfronteerd met de keuzen die het systeem biedt ²⁶. Zo kunnen oprichters van een vennootschap geconfronteerd worden met de verschillende keuzen die het systeem hen biedt en niets anders. In een open user interface zullen ze eerst zelf de vennootschapsvorm moeten opgeven, met alle communicatiestoornissen die daarbij kunnen optreden (bv. P.V.B.A. i.p.v. B.V.B.A.). Er wordt vaak gebruik gemaakt van *menu's* die deze mogelijkheden bevatten. Deze menu's zijn logisch geordend en veranderen naargelang de situatie. Wanneer er bijvoorbeeld niet voldoende kandidaat-vennoten aanwezig zijn, zal het systeem de vennootschappen die niet in aanmerking komen, uit het menu verwijderen. In vele gevallen werkt een gesloten systeem sneller dan een open systeem, maar de gebruiker moet eerst de betekenis van de verschillende opties leren. Door een doordacht en uniform ontwerp kan deze drempel voor de gebruiker gevoelig verlaagd worden. Bovendien blijkt een gesloten systeem vaak *gebruiksvriendelijker* dan een open systeem met een *subset*.

E. DE SPAN-METHODE EN KENNISREPRESENTATIE.

1. Kennisrepresentatie en voorstellingen.

Vele filosofen wezen reeds op het ontbreken van een menselijke voorstelling van de werkelijkheid *an sich*. De werkelijkheid benaderen wij steeds met de instrumenten die wij hebben om deze werkelijkheid te beschrijven, maar de instrumenten zijn onvolmaakt. Daarom is ook deze beschrijving van de werkelijkheid samen met de instrumenten niet volmaakt. Taal is een van deze instrumenten. WITTGENSTEIN maakte reeds duidelijk dat de taal een grens vormt voor ons voorstellingsvermogen. Wat we niet kunnen bespreken, kunnen we ons niet voorstellen. Omdat onze taal, onze woordenschat, onze zinsbouw, beperkt zijn, kunnen we maar een beperkte en vervormde weergave krijgen van de werkelijkheid. Ook bij kennisrepresentatie doet dit fenomeen zich voor. Net zoals de natuurlijke taal zijn kennisrepresentatietaal instrumenten om de werkelijkheid te beschrijven. Dit betekent dat een verkeerde keuze van de kennisrepresentatietaal onvermijdelijk een *inadequate* weergave van de werkelijkheid oplevert.

2. Taal van recht of taal naast recht.

In juridische kennisrepresentatie moet men (minstens impliciet) een keuze maken. Ontwikkelen we systemen die het recht beschrijven of ontwikkelen we systemen die de levensfeitelijke omgeving beschrijven die wij met recht beschrijven? In het eerste geval maken we een beschrijving van een beschrijving, een taal van

(26) SUSSKIND noemt dit *an interactive user interface*. SUSSKIND, R.E., o.c., 66-68.

een taal. In het tweede geval staan het representatiesysteem en het recht naast elkaar als beschrijvingen van de werkelijkheid.

Deze problematiek blijkt duidelijk uit de constructie van de nietigheden in het recht. De juridische nietigheid is één grote "fictie". Vaak gaat het om iets dat al jaren werkelijkheid was. Het recht kan dan ook niet volledig doen alsof zijn neus bloedt en lanceert "correcties" zoals bv. de theorie van de feitelijke ambtenaar. De vraag is wat een kennisrepresentatiesysteem hiermee moet doen. Moeten alle gegevens "retroactief" vernietigd worden wanneer er sprake is van een nietigheid in de betreffende rechtsregel? Moet rekening gehouden worden met de werkelijkheid, waar alles toch heel anders verloopt? Beschrijft het systeem de "werkelijkheid" achter het recht of beschrijft het de rechtsregel?

3. De representatie.

De methode van kennisrepresentatiesystemen kan voorgesteld worden met het volgende schema:

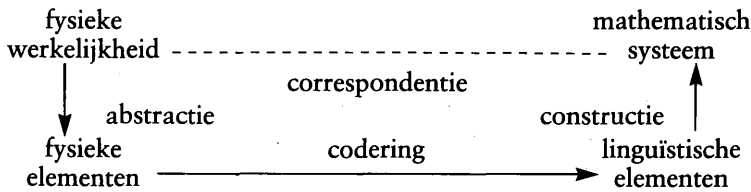


Fig. 5: Representatieschema (algemeen).

In de werkelijkheid onderscheiden we fysieke elementen, die we door codering als linguïstische elementen beschouwen. Door een bepaalde constructie wordt een mathematisch systeem verkregen, dat moet corresponderen met de werkelijkheid. Toegepast op de SPAN-methode wordt dat:

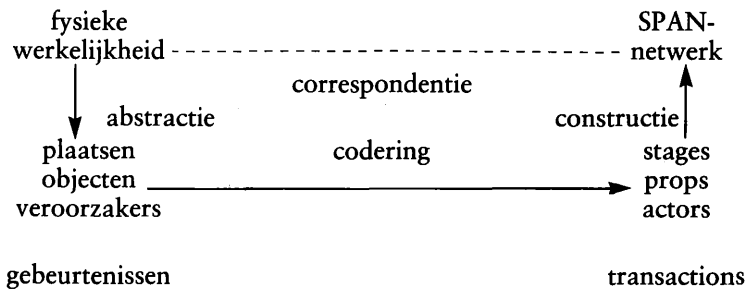


Fig. 6: Representatie in de SPAN-methode.

In de (juridische of gejuridiseerde) werkelijkheid herkennen we plaatsen, objecten, veroorzakers en gebeurtenissen. Toegespitst op de juridische kennisrepresentatie betekent dit dat we rechtssubjecten, rechtsobjecten en "juridische veroorzakers" ²⁷ onderscheiden. Daarnaast herkennen we ook rechtsfeiten en rechtshandelingen, m.a.w. juridische gebeurtenissen. In de SPAN-methode coderen we deze entiteiten naar stages, props, actors en transactions. Door de constructie van de editor en de interpreter (zie *supra*) ontstaat een stage-prop-actor net, een netwerk waarin de stages, props en actors verbonden zijn door links (zie *supra*) die bepalen wie welke transactions mag invoeren. In theorie moet op die manier een betere correspondentie met de werkelijkheid bereikt worden dan met de logische systemen. In de praktijk blijkt dit tot nu toe wel te kloppen. Zonder twijfel zijn er een aantal domeinen waar de logische systemen momenteel een hogere correspondentie bereiken, maar de object-georiënteerde systemen bieden een overtuigend alternatief in vele juridische domeinen.

VI. EEN TOEPASSING : DE VERBINTENIS.

Het verbintenissenrecht is een puur juridisch onderwerp en maakt voor vele andere rechtstakken het gemene recht uit. Nochtans lenen regelgebaseerde talen zich niet zo goed tot het beschrijven van dergelijke toepassingen. De SPAN-methode is weliswaar niet ontworpen met het oog op het assimileren van juridische modellen, maar lijkt voor sommige situaties veruit superieur aan de logische talen, zeker voor wat betreft de puur juridische onderwerpen.

A. EEN EERSTE BENADERING.

De vorming van het contract wordt best beschreven vanuit het perspectief van de contractant. In het Belgisch verbintenissenrecht geldt immers het beginsel van de wilsautonomie, waarbij de partijen onder meer de inhoud van hun contract vrij kunnen bepalen (er zijn enkel negatieve beperkingen). Daarom beschouwt men de contractant best als een stage. Hij is namelijk een drager van rechten en plichten.

Een natuurlijke persoon is vrij om te contracteren (het andere facet van contractsvrijheid), maar ook hier zijn beperkingen. De minderjarige, de geestesgestoorde, de ontvoogde minderjarige, de meerderjarige,... hebben elk een eigen statuut. Het lijkt dus aangewezen om aan de *stage Contractant* meerdere toestanden (*states*) te verbinden. Elke *state* is een bepaald privaatrechtelijk statuut.

(27) Von Jhering wees er al op dat het recht geen statisch gebeuren was, maar dat het recht in de rechtsstrijd ontstaat. Men hoeft het niet zo drastisch onder woorden te brengen, maar het recht begint pas wanneer men een rechtshandeling stelt of wanneer een rechtsfeit zich voordoet. Degenen die deze rechtsfeiten of rechtshandelingen bewerkstelligen, zijn juridische veroorzakers. Zij geven het recht zijn dynamisch karakter.

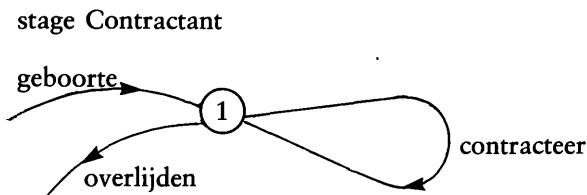


Fig. 7: De states van de stage Contractant (vereenvoudigd).

Het sluiten van een contract is een transaction die grafisch een lus (*loop*) vormt van de state naar dezelfde state. Naargelang de state (afhankelijk van het privaatrechtelijk statuut) kan de persoon die de transaction inroept meer of minder bevoegdheden laten gelden. Dit betekent niet dat al de rest (wilsgebreken, inhoud van het contract, formaliteiten,...) in één transaction moeten beschreven worden. Er zijn nog andere calculi te definiëren (bijvoorbeeld en vooral de prop Contract) die belangrijke delen van de beschrijving van de vorming van het contract overnemen.

De consensus van twee personen (twee stages Contractant) over een bepaalde tekst doet het contract ontstaan. Dit beschrijft men door een transaction *contracteer* op te stellen, die een lus vormt van en naar een state van de stage Contractant. Deze transaction is verbonden met de transaction van de prop Contract die de eerste state inleidt, zodat het contract door de consensus van de partijen "geboren" wordt.

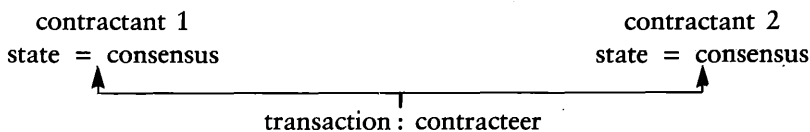


Fig. 8: Het sluiten van een contract.

B. WANPRESTATIE, ONTBINDING EN NIETIGHEID.

Bij de prop Contract kan men nog andere states verzinnen. Zo kan het voorkomen dat de uitwerking van het contract (of liever, de naleving) opgeschort wordt omdat een van de partijen de verbintenissen die hij opnam, niet nakomt. In dat geval is het contract niet minder geldig (behalve bij overmacht). In de werkelijkheid zal dan in de praktijk een kritische fase intreden. De andere partij zal waarschijnlijk uitvoering eisen en/of schadevergoeding of eenvoudige ontbinding van het contract. Hiervoor voert men beter een andere state in voor het contract.

Het is mogelijk dat het contract achteraf nietig blijkt te zijn, bijvoorbeeld omdat er een wilsgebrek aanwezig was bij minstens één van de contractspartijen. Dit wilsgebrek moet door de rechter voldoende ernstig bevonden worden. Het probleem bij nietigheid is dat het contract geacht wordt juridisch nooit bestaan te hebben. In de SPAN-methode betekent dit dat de betreffende props Contract er nooit zijn geweest. Dit wordt beschreven door een *out-toestand* ²⁸ bij deze calculi. Hoe dan ook blijft de nietigheid een sanctie die in het recht niet vaak wordt toegepast (door het stellen van bijkomende voorwaarden en door andere sancties in te voeren) en veel verzachtingen kent ²⁹. Uiteraard is het een juridische fictie die wordt toegepast en in de werkelijkheid blijken de resultaten van nietigheid zeer divers. Rechters manipuleren (binnen de marge van de wet) de sanctie van nietigheid naargelang het individuele geval om bijvoorbeeld een schadevergoeding te simuleren. De vraag is welke criteria een beschrijving van de nietigheidssanctie zal moeten aanwenden (zie *supra*).

C. DE GELDIGHEDSVEREISTEN.

Er zijn een aantal eisen van het recht om te verhelpen aan mogelijke onbillijke gevolgen van een verbintenissysteem dat gekenmerkt wordt door wilsautonomie. Er zijn eisen m.b.t. de rechtsbekwaamheid van de partijen, hun wilsovereenstemming, hun vrije keuze (eigenlijk een deel van hun wilsovereenstemming), de openbare orde en de goede zeden. Sommige criteria zijn louter formeel, sommige vereisen een appreciatie ten gronde. De niet-naleving wordt op verschillende manieren gesanctioneerd. Het gaat hier hoofdzakelijk om regels die door de rechtspraak zijn uitgediept en gedifferentieerd, bijvoorbeeld het civielrechtelijke statuut van de minderjarige in de rechtspraak.

Het lijkt het beste om deze voorwaarden te vertalen in aparte onderdelen binnen de transaction contracteer.

D. DE GRENZEN VAN DE BESCHRIJVING.

Bij object-georiënteerde talen worden de (soorten) objecten omschreven en tegenover de rest van de wereld gekarakteriseerd. Dit is de slagkracht en de beperking ervan. In dit geval zijn er bepaalde grenzen aan een beschrijving van het ontstaan van overeenkomsten.

Een contract ontstaat door de wilsovereenstemming van de rechtssubjecten. Het is onmogelijk om deze wilsovereenstemming te beschrijven. Wat is uiteindelijk een wilsovereenstemming? Wat is goede trouw, overmacht,

(28) Een out-toestand is een pijl die vertrekt vanuit een toestand van een object en nergens naartoe gaat, m.a.w. het object verdwijnt uit het systeem.

(29) Men denke aan het zgn. putatieve huwelijk, de gedeeltelijke nietigheid (waarbij slechts de gewraakte clause vervalt en niet het hele contract), de relatieve nietigheid en de adagia "pas de nullité sans texte" en "pas de nullité sans grief".

dwaling? In de SPAN-methode kan men het niet beschrijven³⁰. Overigens kan het recht deze begrippen evenmin beschrijven. Het zijn juridische begrippen, zeker, maar ze blijven abstract. Het beste bewijs daarvan is de trend naar formalisme en dus rechtszekerheid. Het recht op zich is niet in staat deze begrippen verder te expliciteren, doet een beroep op het gezond verstand van de rechter en geeft hooguit een paar richtlijnen mee.

E. EEN SEMANTISCHE DATABASE.

Na een tijd creëert de rechtspraak een reeks concrete gevallen. Het is duidelijk dat met gezond verstand een aantal gevallen onder dat begrip vallen en een aantal niet. Over de grensgevallen is men verdeeld. Men hanteert dus a.h.w. een lijst van concrete gevallen, waarneembare fenomenen (in de SPAN-methode zijn dit gegevens, data) die onder het begrip lijken te ressorteren. Het is mogelijk om deze rechtspraak in een *database* onder te brengen en in de beschrijving in te passen. Zo krijgt men een lijst van situaties die wel onder het begrip vallen en situaties die er niet onder vallen. Eventueel kan het systeem aan de gebruiker vragenlijsten presenteren waarmee het concrete geval al dan niet onder het begrip kan worden gesubsu-meerd. Bij het foutbegrip in de echtscheiding bijvoorbeeld is het in de rechtspraak duidelijk dat overspel nagenoeg steeds als fout wordt erkend. Wanneer een gebruiker in de vragenlijst overspel aanduidt, kan het systeem besluiten dat het hier om een fout gaat. Bij een verdeelde rechtspraak kan men eventueel een kanspercentage inbouwen. Bovendien kan het systeem dan een dubbele "simulatie" maken, naargelang de mogelijke interpretatie.

VII. BESLUIT.

Bij een kennismaking met de SPAN-methode springen twee zaken in het oog. Enerzijds is de SPAN-methode niet in de eerste plaats geschreven met het oog op juridische kennisrepresentatie. De SPAN-methode beschrijft de werking van organisaties, en de hele calculustechniek en state-transactions zijn daarop afgestemd. Dit betekent dat de SPAN-methode misschien in sommige juridische domeinen minder goed werkt of zelfs faalt.

Anderzijds is de SPAN-methode een programmeertaal en een denkwijze. Prolog (één van de voornaamste "opponenten" van de SPAN-methode) geeft in zekere zin aanleiding tot eng denken in de vorm van relaties en facts en zet aan om uit die feiten bepaalde afleidingen te maken. Wie zich intensief met de SPAN-methode bezighoudt, begint dadelijk te zoeken naar objecten en hun relaties en vooral naar de causaliteit in bepaalde processen die de toestand van bepaalde objecten beïnvloeden. De SPAN-methode is

(30) Dit probleem staat bekend als het *open texture*-probleem. GARDNER, A., "Overview of an Artificial Intelligence Approach to Legal Reasoning", in *Computing Power and Legal Reasoning*, WALTER, Ch. (ed.), St. Paul, West Publishing Company, 1985, (247), 260-266; SUSSKIND, R.E., o.c., 186-193.

niet enkel een denkwijze, ze *vervormt* de werkelijkheid tot op zekere hoogte. Wellicht geldt dit ook voor elke natuurlijke taal.

Een van de kritieken op de rationalistische benadering van regelgebaseerde talen zoals Prolog luidt immers dat "this humanist ideology (...) has a reductionist effect upon knowledge and skills" ³¹. Toch heeft elke representatietechniek een reductionistisch effect. Een taaldaad is niet louter descriptief, maar ook constitutief in zekere zin. Elke "taal", elke manier van spreken is een spreken *over* de levensfeitelijkheid en kleurt diezelfde werkelijkheid.

De SPAN-methode "claimt" geen universele, adequate beschrijving van de werkelijkheid, geen algemene correspondentie. Er kan niet nadrukkelijk genoeg op gewezen worden dat de noodzakelijke consistentie *binnen* het stage-prop-actor net nog niet garant staat voor een correspondentie *met* de werkelijkheid. Consistentie is geen correspondentie. De SPAN-methode "claimt" wel een alternatief te zijn voor de rationalistische benadering van de regel-gebaseerde talen en is er in verschillende domeinen duidelijk superieur aan.

SUSSKIND ³² geeft zes criteria voor een systeem van algemene kennisrepresentatie. Het spreekt vanzelf dat deze criteria aangepast moeten worden aan de meer specifieke eisen voor juridische kennisrepresentatie.

Ten eerste moet het kennisrepresentatiesysteem **flexibel en uitbreidbaar** zijn. De SPAN-methode is hier met de techniek van de actors, het *post-schema* ³³ en de *inception transaction* ³⁴ volledig op afgestemd. Het is mogelijk om de beschrijving van objecten over te nemen en aan te passen wanneer het kennisrepresentatiesysteem wordt uitgebreid. Bij logische talen stellen zich hier problemen. Men zal dan soms het hele model moeten herzien.

Ten tweede moet de kennisrepresentatie **eenvoudig en in haar concept zo beknopt mogelijk** zijn. De standaard "woordenschat" van de SPAN-methode is zeer beknopt en er wordt gebruik gemaakt van zeer krachtige instructies. Er is weinig technische bagage vereist en er wordt naar gestreefd om deze drempel zo laag mogelijk te houden.

Ten derde moeten de **relevante elementen zo duidelijk mogelijk** naar voor worden gebracht. In de SPAN-methode is dit beslist het geval. De methode steunt op het erkennen en beschrijven van determinerende elementen in het systeem dat men bestudeert.

Ten vierde is er de eis van de **volledigheid**. De SPAN-methode biedt de mogelijkheid om de rol van calculi bijna onbeperkt te verfijnen (vooral met

(31) GILL, S.P., *o.c.*, 329.

(32) SUSSKIND, R. E., *o.c.*, 114-116.

(33) Het postschema is een techniek om de beschrijving van de objecten aan te passen. GOOSSENAERTS, J., *o.c.*, 40-41.

(34) De inception transaction is een manier om de beschrijving te laten opstarten of wijzigen door bepaalde stages en actors. GOOSSENAERTS, J., *o.c.*, 41.

*expression refinements*³⁵ en extra *transactions* en *states*). De logische talen kampen vrij snel met onoverzichtelijkheid en contradicties.

Ten vijfde moet kennisoverdracht en -verwerking vergemakkelijkt worden in vergelijking met gangbare werkwijzen. Wanneer men zich in de SPAN-methode heeft verdiept, kan men willekeurige organisaties beschrijven en aan de hand van de beschrijving van anderen de werking van de organisatie begrijpen, veel beter dan dat zou mogelijk zijn met logische talen.

Tenslotte moet het geheel automatiseerbaar zijn en geen abstracte beschrijving blijven. Ook hier is er geen probleem. Er bestaan al *interpreters* of *compilers* voor vele andere object-georiënteerde talen en er zijn veel parallellen met talen die al "*computable*" zijn.

Blijft de vraag of men met het opzet van juridische kennisrepresentatie in het achterhoofd geen bijkomende eisen moet stellen. Er zijn in de bestaande voorbeelden voldoende redenen te vinden om te beweren dat de SPAN-methode voor juridische kennisrepresentatie als werkhypothese voldoet. Natuurlijk moeten er nog veel toepassingen worden gemaakt, die tegelijkertijd ook verificaties zijn voor de adequaatheid van de weergave van de werkelijkheid, m.a.w. voor de overeenstemming (*correspondence*) tussen de SPAN-beschrijving en de werkelijkheid. Zonder twijfel kunnen er nog verschillende kleine correcties aangebracht worden om de beschrijving te vergemakkelijken, maar de basisopties van TIE en de SPAN-methode hebben hun degelijkheid al afdoende bewezen.

(35) *Expression refinements* worden gebruikt om een uitdrukking, die men eerder al had gebruikt, maar die het programma niet kent, te omschrijven in begrippen die het programma wel kent. Een soort van *a posteriori* definitie dus. Hierdoor is het mogelijk om te programmeren met grote stappen om dan later de details uit te werken (*top-down programming*). GOOSSENAERTS, J., o.c., 56-57.