TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Ervaringen met specificaties voor kennissystemen

Citation for published version (APA):
vd Broek, G. M. C. M., Abbink Spaink, H., & Schuwer, R. V. (1990). Ervaringen met specificaties voor kennissystemen. *Informatie*, *32*(6), 531-536.

Document status and date:
Gepubliceerd: 01/01/1990

Document Version:
Uitgevers PDF, ook bekend als Version of Record

Please check the document version of this publication:

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

Link to publication

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

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

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

www.tue.nl/taverne

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

openaccess@tue.nl

providing details and we will investigate your claim.

Download date: 14. jan. 2020

Ervaringen met specificaties voor kennissystemen

G.M.C.M. van den Broek, H. Abbink Spaink en R.V. Schuwer

Het ontwikkelen van een kennissysteem in een bedrijfsomgeving met behulp van de prototyping-aanpak stuit op grote problemen. Evenals bij conventionele systeemontwikkeling lijkt een aantal van deze problemen door gebruik van een gestructureerde methode opgelost te kunnen worden. Hierbij kan veel van conventionele methoden overgenomen worden. Er wordt beschreven wat de inhoud kan zijn van het specificatiedocument tussen de fasen analyse en bouw. Hierbij wordt ook aandacht besteed aan te gebruiken modelleertechnieken.

0 Inleiding

Kennissystemen werden aanvankelijk nagenoeg alleen in een research-omgeving ontwikkeld. De daar gebruikte methode van ontwikkelen, *incremental prototyping*, voldeed goed. Bij het ontwikkelen van een kennissysteem in een bedrijfsomgeving worden aan een methode echter eisen gesteld, waaraan de prototyping-aanpak niet voldoet.

In dit artikel wordt verslag gedaan van ervaringen bij het gefaseerd ontwikkelen van kennissystemen in een bedrijfsomgeving. In het bijzonder is hierbij gekeken naar wat het document tussen de fasen analyse en bouw voor inhoud zou moeten hebben, opdat het enerzijds alle eisen en wensen bevat voor het te ontwikkelen kennissysteem en anderzijds kan dienen als communicatiemiddel tussen de kennisingenieur (de analist) en de softwareingenieur (de bouwer).

De opbouw van dit artikel is als volgt. In hoofdstuk 1 zal een inleiding op het probleemgebied worden gegeven en wordt de probleemstelling geformuleerd. Vervolgens geven we in hoofdstuk 2 onze visie op de wijze van ontwikkelen van een kennissysteem en de inhoud van een specificatiedocument. Deze visie wordt in hoofdstuk 3 geïllustreerd met een praktijkvoorbeeld. Hierbij hebben we ervaringen opgedaan met verschillende modelleertechnieken, die grotendeels vanuit de conventionele systeemontwikkeling werden overgenomen. In hoofdstuk 4 tenslotte staan de conclusies en worden enkele vraagstellingen geformuleerd, die door dit onderzoek zijn ontstaan.

1 Het probleemgebied

Het belangrijkste doel bij het ontwikkelen van kennissystemen in een research-omgeving was vaak het verkennen van de mogelijkheden van deze nieuwe technologie. Deze laboratoriumsystemen werden veelal ontwikkeld door middel van incrementele prototyping.

Omdat de bouwer van zo'n systeem meestal ook de (enige) gebruiker was, was de benodigde materiekennis enerzijds en de nodige automatiseringskennis anderzijds zo in één persoon aanwezig. De bouwer/gebruiker wist dus precies *wat* het systeem moest doen en ook *hoe* het systeem functioneerde. Er werden zo geen eisen gesteld aan gebruiksgemak, uitlegfaciliteiten en systeemdocumentatie. Onder deze omstandigheden voldoet incrementele prototyping uitstekend als ontwikkelmethode.

Bij de introductie van kennissystemen in een bedrijfsomgeving werd ook de, in de research-omgeving zo succesvolle, prototyping-methode meegenomen. Hoewel deze methode goed werkt voor relatief kleine problemen, waarbij de bouwer tevens de gebruiker is, zijn er door de aard van de omgeving enkele belangrijke nadelen te noemen van deze ontwikkel-aanpak.

Een belangrijk motief voor het ontwikkelen van kennissystemen binnen een bedrijfsomgeving is vaak het verspreiden van kennis. Expert, bouwer en gebruiker zijn dan veelal drie verschillende partijen. De materiekennis enerzijds en automatiseringskennis anderzijds is zo verspreid over meerdere personen. Er ontstaat een noodzaak tot communicatie tussen de diverse partijen. Hierbij staat de (informatie-)analist tussen de expert en de gebruiker enerzijds en de bouwer (de software-ingenieur) anderzijds. De ontwikkelmethode zal daarom moeten voorzien in project- en systeemdocumenten. Tevens worden eisen gesteld aan de user interface. Zo zal het systeem de oplossing moeten kunnen verklaren aan de gebruiker en zal het systeem ook eenvoudig in gebruik moeten zijn.

De ontwikkeling van een kennissysteem zal efficiënt moeten verlopen. In het algemeen is een snel verkregen 70%-oplossing te verkiezen boven een moeizaam verkregen 100%-oplossing. De beheersing van het ontwikkelproces is echter moeilijk door het ontbreken van concrete mijlpalen. Tevens is de haalbaarheid van het systeem moeilijk in te schatten, omdat er geen specificaties opgesteld zijn voordat aan de implementatie wordt begonnen.

De prototyping-methode kan worden gekarakteriseerd als een afwisseling van analyse en implementatie. Bij grotere en complexere systemen zal in de loop van het ontwikkelproces naar verhouding steeds meer aandacht moeten worden besteed aan de implementatie. Hierdoor groeit de tijd tussen twee analysesessies met de expert gestaag, met het risico dat de expert zijn/haar belangstelling voor het project verliest of zelfs helemaal niet meer beschikbaar is als gevolg van overplaatsing of pensionering.

Omdat bij de prototyping-methode het prototype wordt gebruikt als communicatiemiddel met de expert, is de expert geneigd zijn kennis op een voor hem onnatuurlijke manier te formuleren. Zo zal hij bij het gebruik van een

INFORMATIE JAARGANG 32 NR. 6
PAG. 513 T/M 596 1990

op regels gebaseerde kennissysteem-shell ook in IF-THEN-termen gaan praten.

Als deze nieuwe technologie binnen een bedrijf op brede schaal toegepast gaat worden, dan zal dus veel aandacht moeten worden besteed aan de methode van ontwikkelen en de documenten, die zo'n methode oplevert. Het gebruik van zo'n methode zal de bovengenoemde nadelen zoveel mogelijk moeten voorkomen. Welke inhoud de documenten moeten hebben, wordt in het volgende hoofdstuk behandeld.

2 Specificaties voor een kennissysteem

Hoewel kennissystemen zijn ontstaan binnen het terrein van de kunstmatige intelligentie, beschouwen wij een kennissysteem als een type informatiesysteem. Daarom kan naar onze mening voor de ontwikkeling van een kennissysteem veel uit de ontwikkelmethoden voor conventionele systemen worden gebruikt, maar er moet iets aan worden toegevoegd. Deze visie op kennissystemen is te meer noodzakelijk, daar kennissystemen binnen een organisatie niet los kunnen worden gezien van de reeds bestaande infrastructuur van software en hardware. Kennissystemen zijn het meest waardevol als ze goed integreren met die infrastructuur (onder andere gemeenschappelijke databases).

Veel van de ontwikkelproblemen die in het vorige hoofdstuk zijn genoemd, zijn ook al ervaren bij het ontwikkelen van conventionele systemen. Een gevolg toen was, dat het ontwikkelen van informatiesystemen op een gestructureerde wijze ging plaatsvinden. Er werden diverse methoden ontwikkeld, die bijna allemaal uitgingen van een of andere fasering. Uitgangspunt van al deze methoden was de idee van scheiding tussen ontwerp en bouw.
Het gebruik van een gestructureerde methode voor het ontwikkelen van een kennissysteem lijkt dan ook een logische wijze van werken.
Als kennissystemen echter ontwikkeld worden door middel van een meer gestructureerde aanpak, zullen er specificaties moeten worden gemaakt. In deze specificaties worden de inhoud van het systeem (de kennis) en de wensen van de gebruikers vastgelegd. Deze specificaties dienen als communicatiemiddel tussen de analist en enerzijds de expert en gebruikers, anderzijds de software-ingenieur.

Uit de conventionele methoden kan inspiratie worden opgedaan om te vinden wat een specificatie-document moet bevatten.
Bemelmans geeft aan (1987) dat de specificaties in twee klassen zijn te verdelen:

1 functionele specificaties;
2 niet-functionele specificaties.

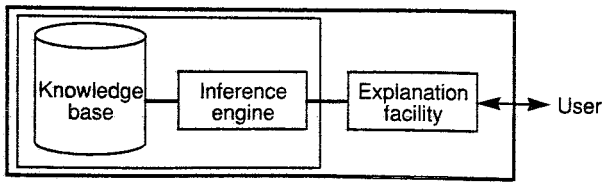
Functionele specificaties beschrijven welke gegevens verwerkt en verstrekt moeten worden. Meestal maken een *procesmodel* en een *datamodel* deel uit van de functionele specificaties. Het is ook raadzaam een beschrijving van de gebruikte definities (jargon!) te geven. Dit geeft aan de technisch ontwerper enig inzicht in de specificaties van het systeem.
Niet-functionele specificaties beschrijven de kwaliteit van de te verwerken en te verstrekken gegevens. Niet-functionele specificaties worden daarom ook wel *prestatie-eisen* of *kwaliteitseisen* genoemd. In dit onderzoek hebben we ons alleen beziggehouden met de functionele specificaties.

Specificaties voor een kennissysteem zullen op punten afwijken van die van een conventioneel systeem. Om te weten wat die afwijkingen zijn, moeten we de verschillen tussen een kennissysteem en een conventioneel systeem onderzoeken en nagaan welke gevolgen dit heeft voor de specificaties.

Als definitie voor een kennissysteem houden we aan (Mars, 1988):

> 'Een kennissysteem is een computerprogramma, waarin zo goed mogelijk een scheiding is aangebracht tussen toepassingsgebied-onafhankelijke afleidingsregels en toepassingsgebied-specifieke kennis.'

Uit deze definitie halen we het meest in het oog springende verschil van een kennissysteem met de hiervoor genoemde systemen, namelijk het aanwezig zijn van een *kennisbank*, waarin de toepassingsgebied-specifieke kennis ligt opgeslagen en een *inferentie-mechanisme*, waarmee de toepassingsgebied-onafhankelijke afleidingsregels worden gerealiseerd. Een ander verschil ligt in de aanwezigheid van een *uitlegfaciliteit*, waarmee het systeem aan de gebruiker zijn conclusies kan verklaren, respectievelijk aan de gebruiker de achtergrond van een gestelde vraag kan meedelen. Schematisch kan een en ander weergegeven worden zoals in figuur 1.



Figuur 1: Architectuur van een kennissysteem

In de specificaties zal een beschrijving moeten komen van zowel de toepassingsgebied-afhankelijke kennis als de wijze waarmee het inferentiemechanisme deze kennis bewerkt. Dit kan op een gestructureerde wijze gebeuren door naast een data- en een procesmodel ook een *con-*

ceptueel kennismodel te ontwikkelen.
Om aan de specifieke kennis te komen, zal vaak een expert moeten worden ondervraagd met het doel, zijn of haar (ervarings)kennis geformaliseerd te krijgen. Anders dan bij de hiervoor genoemde systemen speelt dit te formaliseren proces zich af in het hoofd van een persoon. Het is dit verschil dat het proces van kennisverkrijging vaak tot een moeizame affaire maakt. Het kennismodel kan als communicatiemiddel dienen tussen de analist en de expert. Eveneens geeft dit model voor de systeembouwer inzicht in de structuur van de gebruikte kennis. De te gebruiken wijze van weergeven van het kennismodel zal daarom zodanig moeten zijn, dat aan deze functies op een effectieve manier kan worden voldaan.

De meeste in de praktijk gerealiseerde systemen maken de expert niet overbodig. Deels vindt dit zijn oorzaak in de onvolledigheid van het kennismodel, deels in de beperkte mogelijkheden van de technologie en deels in het leggen van de verantwoordelijkheid voor de uiteindelijke beslissing bij de gebruiker. Een gevolg hiervan is, dat het systeem steeds in staat moet zijn, verantwoording af te leggen aan de gebruiker over de conclusies, waartoe het gekomen is. In de specificaties zal deze *uitlegfaciliteit* gedefinieerd moeten worden. Deze uitlegfaciliteit kan als onderdeel van het procesmodel worden beschreven.

Een specificatie voor een kennissysteem zal daarom tenminste de volgende punten moeten bevatten:
- datamodel;
- procesmodel (inclusief een specificatie van de uitlegfaciliteit);
- kennismodel;
- beschrijving van de gebruikte definities.

Het volgende hoofdstuk geeft een praktijkvoorbeeld. Hierbij werd in het bijzonder nagegaan, op welke wijze het kennismodel kan worden weergegeven.

3 Praktijkvoorbeeld

Het hier beschreven voorbeeld speelde zich af bij een industrieel bedrijf in Nederland. Hier werd medio 1987 besloten tot de ontwikkeling van een kennissysteem voor de oplossing van een technisch probleem. Dit probleem had te maken met materiaalselectie voor apparatuur, die blootgesteld wordt aan extreme temperaturen. Op dit gebied was reeds een uitgebreid voorschrift aanwezig, maar dit bleek in de praktijk veel problemen op te leveren, zodat dikwijls de hulp van een expert ingeroepen moest worden. Het kennissysteemproject werd slechts anderhalve maand voor de pensionering van de expert gestart, zodat een zeer efficiënte aanpak van de kennisacquisitie noodzakelijk was.

Bij het opstellen van de specificaties moest voor het weergeven van het data-, proces- en kennismodel een modelleertechniek worden gekozen.

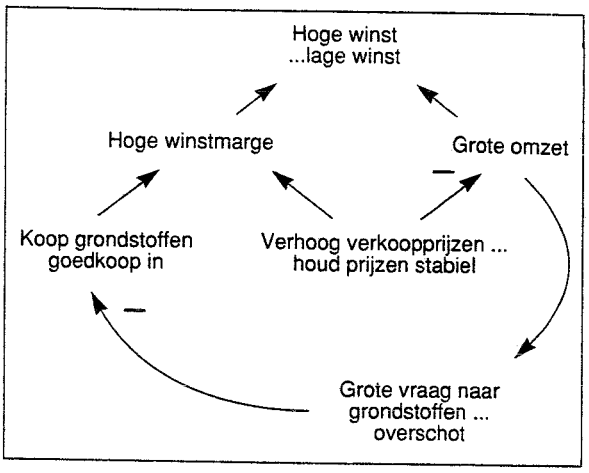
Het datamodel geeft de entiteiten en de relaties tussen de entiteiten die een rol spelen in het redeneerproces weer. Hiervoor zijn reeds verschillende diagramtechnieken beschikbaar vanuit de conventionele systeemontwikkeling (Bachman-diagram, Entity-Relationship-diagrammen (ER-diagram), Entity-Attribute-Relationship-diagrammen (EAR-diagram), enzovoorts). In dit onderzoek is er gekozen voor EAR-diagrammen. De keuze van de diagramtechniek is echter niet relevant, zolang alle entiteiten en relaties die van belang zijn er inzichtelijk in weergegeven kunnen worden.

Het procesmodel geeft weer hoe het proces die het systeem moet verrichten, opgedeeld kan worden in deelprocessen, hoe deze deelprocessen zich ten opzichte van elkaar verhouden, en welke input en output deze deelprocessen hebben. Ook hiervoor zijn reeds diverse diagramtechnieken beschikbaar vanuit de conventionele systeemontwikkeling (ISAC A-schema's, dataflow-diagrammen (DFD), enzovoorts).
In dit onderzoek is gekozen voor aangepaste ISAC A-schema's. De aanpassing ten opzichte van de oorspronkelijke diagramtechniek is, dat er onderscheid is gemaakt tussen trigger-informatie en input-informatie. Onder *trigger-informatie* (te vergelijken met de control-flow bij SADT-diagrammen) moet worden verstaan die informatie, die een proces aanzet tot het uitvoeren van activiteiten. *Input-informatie* is alle informatie die een proces nodig heeft bij het uitvoeren van activiteiten. De reden voor deze aanpassing is, dat we onderscheid wilden kunnen maken tussen beide soorten input. Hoewel SADT-diagrammen daarom een geschiktere keuze lijken, werd toch voor ISAC A-schema's gekozen om de eenvoudige reden, dat de analist hiermee de meeste ervaring had opgedaan.

Voor het weergeven van de verkregen kennis voor de communicatie met de expert werd gebruik gemaakt van een techniek uit de Management Consultancy: *cognitive mapping* (Eden, 1983). De cognitive mapping-techniek is ontwikkeld door prof. Colin Eden c.s. voor het analyseren en structureren van ideeën die binnen een groep leven. De techniek is gebaseerd op de uit de psychologie afkomstige 'personal construct theory'. Cognitive mapping is een manier om kennis, ideeën en opvattingen op een voor ieder begrijpelijke manier te modelleren. Figuur 2 geeft een voorbeeld van een cognitive map.

In essentie bestaat een cognitive map uit concepten en hun onderlinge samenhang. Bij ieder concept wordt ge-

INFORMATIE JAARGANG 32 NR. 6
PAG. 513 T/M 596 1990



Figuur 2: Voorbeeld van een cognitive map

tracht de psychologische tegenstelling aan te geven. Deze kan belangrijk afwijken van de logische tegenstelling. Afhankelijk van de context zijn bijvoorbeeld de volgende tegenstellingen van het concept 'hoge winst' denkbaar (met tussen haakjes de wijze, waarop 'hoge winst' dan moet worden opgevat in de context van de genoemde tegenstelling):

- lage winst ('in vergelijking tot voorgaande jaren heeft het bedrijf dit jaar een flinke winst geboekt');
- woekerwinst ('de winsten mogen dan hoog te noemen zijn, maar van woekerwinsten is geen sprake');
- grote verliezen ('de hoge winst van dit jaar is nodig om de grote verliezen ten gevolge van investeringen in voorgaande jaren te compenseren').

De tegenstelling geeft dus een verduidelijking van het concept.

De relaties die tussen concepten gedefinieerd worden, zijn hoofdzakelijk causale verbanden.

Hoewel de cognitive mapping-techniek primair ontwikkeld is voor Management Consultancy-achtige doeleinden (bijvoorbeeld het ontwikkelen van een strategie voor een bedrijf), zijn er duidelijke links met het modelleren van kennis ten behoeve van kennissystemen. De causale verbanden kunnen bijvoorbeeld nagenoeg direct vertaald worden in produktieregels en ook in de verdere analyse zijn er veel aspecten die goed aansluiten bij ideeën over kennisacquisitie, zoals het vinden van taakstructuren.

De cognitive maps worden opgesteld aan de hand van de interviews met een expert. Met behulp van speciaal daartoe ontwikkelde software ('COPE' (Eden, 1988)) worden vervolgens de maps geanalyseerd. Een belangrijke activiteit daarbij is het zoeken naar *clusters*: samenhangende stukken redenering die toewerken naar een bepaalde conclusie. Binnen een kennissysteem worden clusters vertaald naar taken. COPE bevat een speciaal algoritme voor het uitvoeren van cluster-analyse. De COPE-software bevat tevens vele andere faciliteiten, zoals het testen op volledigheid en consistentie van het kennismodel.

Het gebruik van de cognitive mapping-techniek als communicatiemiddel met de expert, in combinatie met COPE, heeft enige belangrijke voordelen:

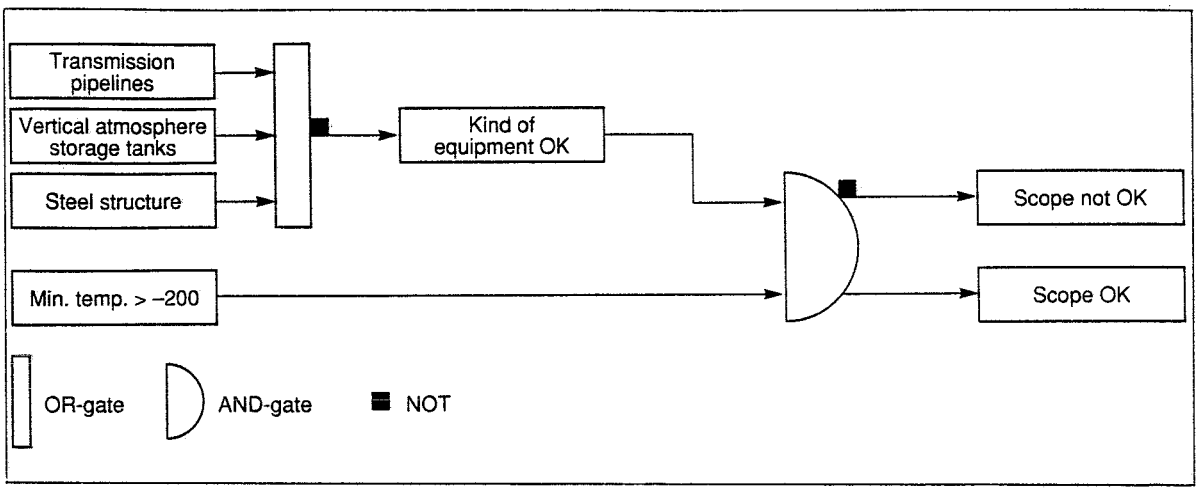
- de representatie is simpel en kan gemakkelijk door een expert begrepen worden. Controle op de redeneerstructuren is daardoor veel eenvoudiger geworden;
- de expert ziet vrij direct wat er met zijn/haar kennis gebeurt en voelt zich meer 'eigenaar' van de kennis. Dit 'ownership'-aspect leidt meestal tot een grotere betrokkenheid bij het project;
- de representatie is implementatie-onafhankelijk;
- de kennis kan gemakkelijker getoetst worden op consistentie en volledigheid.

Voor de implementatie van het kennissysteem bleken de cognitive maps echter niet voldoende als specificatie. Het is namelijk voor een niet in het probleem ingewijd persoon moeilijk om de benodigde informatie uit de diagrammen te achterhalen. Dit kwam mede door het feit dat de gebruikte modelleertechnieken niet eenduidig waren. Vandaar dat werd gezocht naar een andere wijze van weergeven van het kennismodel, die wel geschikt is voor de software-ingenieur. Hierbij moeten de redeneringen die de invulling vormen van de processen op het laagste hiërarchische niveau van het procesmodel worden weergegeven. Voor deze modellen vonden we geen geschikte diagramtechniek die overgenomen kon worden uit de conventionele systeemontwikkeling.

In dit geval is ervoor gekozen deze redeneringen weer te geven in logische AND/OR-diagrammen, zoals die ook gebruikt worden in de elektrotechniek voor het ontwerpen van logische schakelingen. Deze notatietechniek maakt het mogelijk complexe redeneringen toch visueel inzichtelijk te maken, en voorkomt de problemen van de cognitive mapping-notatie. Figuur 3 geeft een voorbeeld van de weergave van een deel van een kennismodel met behulp van deze diagramtechniek.

De vertaling van COPE-diagrammen naar AND/OR-diagrammen werd door de analist gedaan. Deze vertaling bleek goed uit te voeren.

Bij de uiteindelijke implementatie van het systeem werd gekozen voor de frame-based shell 'Acquaint'. Hierbij bleek dat de gebruikte AND/OR-diagrammen vrijwel letterlijk één op één vertaald konden worden naar regels in Acquaint. Zo ziet de Acquaint-vertaling van het AND/OR-diagram uit figuur 3 er als volgt uit:



Figuur 3: Voorbeeld van een deel van een kennismodel

```
(DefRule Check_scope
IF: $BACKWARD
    Equip_type_ok
    Min_temp_ok
THEN: 100 Scope_ok
ELSE: -100 Scope_ok ]
(DefRule Check_equip
IF: $BACKWARD $OR
    Equip_type = "Transmission pipelines"
    Equip_type = "Vert. atmospheric st. tank"
    Equip_type = "Steel structures"
THEN: -100 Equip_type_ok
ELSE: 100 Equip_type_ok ]
(DefRule Check_min_temp
IF: $BACKWARD $NOT Min_temp
THEN: 100 Min_temp_ok
ELSE: -100 Min_temp_ok ]
```

Bovendien vormden de ISAC A-schema's een goede basis voor de structurering van de rulebase; ieder (deel)proces van ISAC kon vertaald worden naar een (deel)context van Acquaint. De trigger-informatie leverde nuttige informatie voor de structuuropbouw van het uiteindelijke systeem. Tenslotte was het maken van de systeemdocumentatie erg gemakkelijk, omdat veel hiervan al in de logische specificaties was terug te vinden.

4 Conclusies

Het ontwikkelen van een kennissysteem binnen een bedrijfsomgeving stelt diverse eisen aan de methode van ontwikkelen en de wijze van documentatie. Door het gebruik van de incremental prototyping-methode wordt aan deze eisen niet voldaan.

Uit ervaringen die met een praktijk-project werden opgedaan, bleek, dat voor specificatie-methoden van een kennissysteem veel inspiratie uit conventionele methoden kan worden geput. De daar gebruikte technieken kunnen zonder meer ook voor kennissysteemontwikkeling worden gebruikt.

Het onderzoek heeft echter ook enkele vragen opgeroepen en onbeantwoord gelaten. Mogelijk dat die onderwerp kunnen zijn voor verder onderzoek.

- Bij het probleem was geen sprake van onvolledige of onzekere kennis. Het is voorstelbaar, dat dit extra eisen oplegt aan de specificaties. Vooralsnog is onduidelijk welke die extra eisen zijn en in welke vorm die gedocumenteerd kunnen worden.
- De in dit artikel beschreven resultaten van met name de gebruikte modelleringstechnieken zijn toepasbaar op de klasse van diagnostische problemen. Het is echter niet direct duidelijk of bij bijvoorbeeld een planningprobleem ook van soortgelijke technieken gebruik kan worden gemaakt.
- Als input voor de implementatiefase zou een prototype-kennissysteem gebruikt kunnen worden. Het prototype heeft dan slechts als communicatiemiddel tussen expert en kennisanalist gediend. Het is echter onzeker welke aspecten van een specificatie dan door zo'n prototype opgeleverd worden.
- Inbedding van het kennissysteem in een groter (conventioneel registratief) systeem zal een beschrijving van de interfaces tussen beide systemen vereisen.

Literatuur

Bemelmans, T.M.A. (1987). *Besturlijke informatiesystemen en automatisering*. Stenfert Kroese, Leiden, derde herziene druk.

Eden, C., S. Jones, D. Sims (1983). *Messing around in problems*. Pergamon Press, Oxford.

Eden, C., T. Smithin, M. Jones, J. Norris en P. Huggett (1988). *COPE version 10.8.3*. University of Strathclyde, Glasgow.

Mars, N. (1988). 'Onderzoek van niveau: Kennistechnologie in wording', *Informatie* 30, blz. 84-90.

Dr. H. Abbink Spaink studeerde fysische chemie en promoveerde aan de Universiteit van Amsterdam. Zijn promotie-onderzoek had betrekking op

de automatisering van een meetinstrument voor de bepaling van verontreinigingen in de atmosfeer. Vanaf 1986 houdt hij zich bezig met het ontwikkelen van kennissystemen voor gebruik in een zakelijke omgeving. Sinds augustus 1989 werkt hij bij BSO/Artificial Intelligence te Utrecht.

Ir. G.M.C.M. van den Broek studeerde af als informatica-ingenieur aan de TUE in 1988, bij de Vakgroep Bestuurlijke Informatiesystemen en Automatisering (BISA) van de afdeling Bedrijfskunde. Zij is sedert 1-2-1989 werkzaam als 'business information analyst' in een 'informatie- en organisatie-unit' bij Shell Nederland Raffinaderij. Zij houdt zich daar bezig met kantoorautomatisering.

Ir. drs. R.V. Schuwer studeerde wiskunde aan de KU te Nijmegen en informatica aan de TU te Eindhoven. Achtereenvolgens was hij werkzaam als docent wiskunde in het middelbaar onderwijs en als consultant van het informatiecentrum binnen Océ-van der Grinten te Venlo. Momenteel werkt hij als universitair docent aan de TU te Eindhoven binnen de vakgroep Bestuurlijke Informatiesystemen en Automatisering van de faculteit bedrijfskunde. Hij houdt zich daar onder andere bezig met (promotie-)onderzoek naar kennissystemen.