

OBJECT ORIËNTATIE BIJ KENNISMODELLERING

Ir. P.H.M. Spronck

Ir. K.O. Schilstra

Beide auteurs waren ten tijde van het schrijven van dit document werkzaam bij TNO
Bouw, afdeling Kennisgebaseerde Systemen (kennis@bouw.tno.nl)

Inleiding

Een bekende TNO slogan stelt dat kennis kracht is. Bedrijven die aan kennismanagement doen willen deze kracht vasthouden en vermenigvuldigen. Daarom zetten dit soort bedrijven kennissystemen in. Kennissystemen worden met name gebruikt om kennis die bij slechts een beperkt aantal experts bekend is, toegankelijk te maken voor een grotere groep gebruikers. Daarnaast kunnen ze worden gebruikt om nieuwe experts te trainen, of om de gezamenlijke kennis van verschillende experts op een bepaald gebied consistent te maken. Dit soort kennissystemen wordt ook wel "expertsystemen" genoemd.

Kenmerken van kennissystemen

Conventionele programmatuur bestaat over het algemeen uit een programma, dat in principe onveranderlijk is, en een database, waarin de gebruiker zijn eigen gegevens vastlegt. Kennissystemen karakteriseren zich door het feit dat naast een programma en een eventuele database, de programmatuur ook beschikt over een zogenaamde kennisbank. De kennisbank bevat redeneergegevens, zoals bijvoorbeeld kennisregels (zie [Fig1]). Het kennissysteem kan met behulp van deze redeneergegevens, en gegevens die door de gebruiker tijdens de consultatie van het systeem verstrekt worden, conclusies trekken over het kennisdomein waarvoor de kennisbank ontworpen is.

Twee zaken zijn met name van belang voor een kennissysteem, en dit zijn net de zaken waarmee een kennissysteem zich onderscheidt van een conventioneel stuk programmatuur. Ten eerste, de kennis vastgelegd in de kennisbank is expliciet. In de kennisbank staat eenduidig te lezen wat er moet gebeuren in specifieke situaties. Bijvoorbeeld, het (gefingeerde) regelgebaseerde kennissysteem weergegeven in figuur [Fig1] stelt vast of er in een bepaalde kamer een brandblusser geïnstalleerd moet worden. We zien hier expliciete kennis geformuleerd in regels als "Als een kamer open vuur bevat, is de kamer brand-onveilig". Veel van de kennis die mensen bezitten is echter impliciet. Mensen redeneren intuïtief, zonder dat ze de neiging hebben de impliciete kennis te vertalen in expliciete regels. Een kennissysteem kan helpen de kennis van mensen expliciet te maken. Het feit dat de kennis expliciet is, is van belang, omdat het experts op het kennisdomein in staat stelt om te controleren of de kennis correct en compleet is.

Het tweede kenmerkende punt van een kennissysteem, is het feit dat de kennis onderhoudbaar is zonder dat de programmatuur hoeft te wijzigen. Dit is belangrijk, omdat in de praktijk kennis regelmatig wijzigt en aangevuld moet worden. De scheiding stelt ons in staat om wijzigingen in de kennis te maken zonder dat de software aanpassing behoeft. Net zoals nieuwe gegevens in een database de werking van de programmatuur in

de ogen van een gebruiker niet veranderen, zal een gebruiker geen verschil hoeven op te merken in de werking van een kennissysteem als de kennis wijzigt.

IF kamer-diagnose = brand-veilig **THEN** brandblusser-installeren = nee **ELSE** brandblusser-installeren = ja

IF kamer-bevat-open-vuur **THEN** kamer-diagnose = brand-onveilig

IF kamer-muren = dik **THEN** kamer-diagnose = brand-veilig

IF aansluitende-kamer-diagnose = brand-onveilig **THEN** kamer-diagnose = brand-onveilig **ELSE** kamer-diagnose = brand-veilig

IF aansluitend-kamer-bevat-open-vuur **THEN** aansluitende-kamer-diagnose = brand-onveilig

IF aansluitende-kamer-type \diamond slaapkamer **THEN** aansluitende-kamer-diagnose = brand-veilig

IF bewoners-roken-in-bed **THEN** aansluitende-kamer-diagnose = brand-onveilig **ELSE** aansluitende-kamer-diagnose = brand-veilig

Figuur [Fig1]: Een regelgebaseerd kennissysteem dat volgens gefingeerde regels vaststelt of een kamer brand-onveilig is en er dus een brandblusser geïnstalleerd moet worden. De regels zijn, zoals gebruikelijk bij regelgebaseerde kennissystemen, van het "IF-THEN-ELSE" type. De regels worden van boven naar beneden geëvalueerd, en evaluatie stopt als een gezochte waarde gevonden is. De regels stellen dat er een brandblusser geïnstalleerd moet worden als de kamer brand-onveilig is. Een kamer is brand-onveilig als er open vuur is, of als de kamer verbonden is met een andere kamer via dunne muren, waarbij de aansluitende kamer brand-onveilig is. Een aansluitende kamer is brand-onveilig als er open vuur is, of als het een slaapkamer is waarbij de bewoners in bed roken.

Het feit dat de kennis expliciet is en onderhouden kan worden onafhankelijk van de programmatuur, betekent nog niet dat het voor de expert gemakkelijk is om de kennis te formuleren, te controleren en te wijzigen. In een ideale wereld wordt de expert ondersteund door een stuk gereedschap (een "tool"), die de kennisbank kan inlezen, aan de expert kan tonen op een manier die hij begrijpt, functionaliteiten biedt om te kennisbank te exploreren, en gebruikersvriendelijke manieren biedt om de kennis aan te passen en te testen. De praktijk wijst helaas uit dat de tools die heden ten dage op de markt zijn, niet aan deze eisen voldoen. Het onderhouden van een kennisbank door een expert is over het algemeen slechts mogelijk als de expert ondersteund wordt door een zogenaamde "kennistechnoloog".

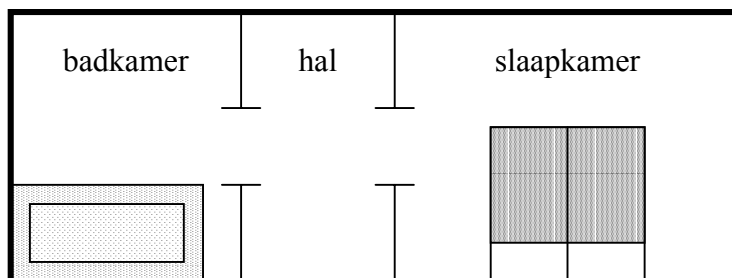
Een kennistechnoloog is een (veelal academisch geschoolde) informaticus die een speciale opleiding heeft genoten op het gebied van de kennistechnologie. Hij is niet alleen op de hoogte van de gebruikelijke technieken in de informatica, maar ook speciaal onderricht is om kennis te vergaren in gesprekken met experts. De kennistechnoloog

structureert de nieuw vergaarde kennis in een kennisrepresentatie voor de kennisbank. Bij elke wijziging in de kennis moet de kennistechnoloog aanwezig zijn om de vertaling naar de kennisbank te maken. Het is duidelijk dat hier een gevoelig punt zit, omdat kennis soms snel kan wijzigen en een kennissysteem net als een expert geen foutieve adviezen mag geven. Continue beschikbaarheid van een kennistechnoloog is een vereiste als de experts niet zelf hun kennisbanken kunnen onderhouden.

Het is wenselijk dat voor het onderhoud van de kennisbank de kennistechnoloog niet meer benodigd is. Immers, de kennistechnoloog kan niet dagelijks bezig zijn met het kennisdomein dat in de kennisbank is vastgelegd, en zal zich voor onderhoud op het systeem opnieuw in het vakgebied van de expert moeten verdiepen. Dit vergt tijd en is over het algemeen erg kostbaar. De expert zelf, die tenslotte ook degene is die constateert dat de kennisbank gewijzigd moet worden, is de aangewezen persoon om de wijzigingen door te voeren.

Visuele representatie van kennis

Een van de voornaamste redenen waardoor kennisbanken ontoegankelijk zijn voor experts, is dat de kennis zoals die door de tools gepresenteerd wordt niet aansluit bij de gedachtenwereld van de expert. Voor een expert bevindt zijn kennis zich in een bepaalde visie die hij op het kennisdomein heeft. Deze visie is niet terug te vinden in, bijvoorbeeld, een grote hoeveelheid regeltjes. Net zoals bij conventionele programmatuur, zal een kennistechnoloog die samen met een expert een kennisbank opzet, een model van het kennisdomein ontwerpen dat gebruikt wordt om met de expert te discussiëren. Dit model is in feite de visie van de expert op het kennisdomein. Dit is het model dat hij begrijpt. Het model is echter niet zichtbaar terug te vinden in de tools, althans niet direct verbonden met de kennisbank. Om wijzigingen te kunnen maken in de kennisbank heeft de expert twee dingen nodig: begrip van de kennisbank en een vocabulaire waarin hij zich uit kan drukken. Om dit mogelijk te maken moet worden gekeken naar visuele representatievormen van kennis die de structuur en inhoud van een kennisbank zichtbaar kunnen maken.



Figuur [Fig2]: Een verdieping van een gebouw zoals een expert zich die voorstelt: kamers die met elkaar verbonden zijn, waarbij iedere kamer een eigen functie heeft. De badkamer heeft geen open vuur, en is dus niet brand-onveilig van zichzelf, maar als de hal brand-onveilig is, moet de badkamer ook als brand-onveilig beschouwd worden omdat de twee kamers verbonden zijn via dunne muren.

Neem bijvoorbeeld het regelgebaseerde kennissysteem uit figuur [Fig1]. Een hypothetische expert op het gebied van brandbeveiliging beziet zijn kennisdomein niet

als de regeltjes die hier geformuleerd zijn. Hij stelt zich een gebouw voor als een serie kamers, die met elkaar verbonden zijn (zie figuur [Fig2]). Kamers hebben overeenkomsten (ze hebben allemaal wanden, ieder met hun eigen dikte), maar ook verschillen (woonkamers kunnen bijvoorbeeld een open haard hebben, in tegenstelling tot slaapkamers). Om vast te stellen of een brandblusser in een kamer geïnstalleerd moet worden, maakt de expert een diagnose van de toestand van de kamer in de context van het gebouw. Sommige kamers zijn brand-onveiliger dan andere. Een woonkamer met een open haard is brand-onveilig, en een slaapkamer waarin de bewoners van het pand in bed liggend roken ook. Kamers die slechts door een dunne muur gescheiden zijn van brand-onveilige kamers kunnen zelf ook brand-onveilig genoemd worden (nota bene en wellicht ten overvloede: dit is gefingeerde kennis die geen enkele relatie met de praktijk heeft).

Dit soort kennis is in figuur [Fig1] door een kennistechnoloog geformuleerd, maar niet op een voor de expert duidelijke manier. Dit is zeker niet het geval als het gaat om een iets groter model waarbij het aantal regels en feiten in de honderden kan lopen. Het onderhoud aan dergelijke kennisbanken wordt in de praktijk al snel onmogelijk. Wijzigingen op een kennisbank worden daarom vaak niet direct doorgevoerd maar verzameld, en als het aantal groot wordt, wordt de kennisbank geheel opnieuw opgezet met alle kosten en inspanningen vandien. Dit geeft aan dat er een aanpak nodig is die deze feiten inzichtelijk presenteert en groepeerd. Beslissingstabellen zijn een mogelijkheid om regels zodanig te presenteren en groeperen dat de kennis voor de expert leesbaar en onderhoudbaar wordt. In figuur [Fig3] staat dezelfde kennis afgebeeld als in figuur [Fig1], maar nu in de vorm van beslissingstabellen.

<i>brandblusser-installeren</i>		
<u>kamer-diagnose</u>	brand-veilig	brand-onveilig
brand-blusser-installeren	nee	ja

<i>kamer-diagnose</i>				
kamer-bevat-open-vuur	ja	nee		
kamer-muurdikte	-	dik	dun	
<u>aansluitende-kamer-diagnose</u>	-	-	brand-onveilig	brand-veilig
kamer-diagnose	brand-onveilig	brand-veilig	brand-onveilig	brand-veilig

<i>aansluitende-kamer-diagnose</i>				
aansluitende-kamer-bevat-open-vuur	ja	nee		
aansluitende-kamer-type	-	slaapkamer	ELSE	
bewoners-roken-in-bed	-	ja	nee	-
aansluitende-kamer-diagnose	brand-onveilig	brand-onveilig	brand-veilig	brand-veilig

Figuur [Fig3]: De kennisbank uit figuur [Fig1], nu in de vorm van beslissingstabellen. Een beslissingstabel dient als volgt gelezen te worden: Het "tabje" bovenaan de tabel bevat de tabel-titel. De linker-kolom, onder het tabje, op de laatste regel na, bevat zogenaamde "condities". Onderstreepte condities zijn verwijzingen naar andere beslissingstabellen. De waarden die de condities kunnen hebben staan rechts in dezelfde rij. Door van boven naar beneden door de condities te lopen en de kolom te volgen waarin de waarde voor de betreffende conditie staat, waarbij vakjes met een streepje kunnen worden overgeslagen, komt men uiteindelijk in één van de vakjes op de onderste rij uit. Dit vakje bevat de waarde die moet worden gegeven aan de zogenaamde "actie", die in het vakje linksonder in de tabel staat. Zo kunnen we in de tabel "kamer-diagnose" zien, dat als de conditie "kamer-bevat-open-vuur" de waarde "ja" heeft, de actie "kamer-diagnose" de waarde "brand-onveilig" krijgt.

Met het opzetten van een kennisbank als een verzameling beslissingstabellen is een deel van het onderhoudsprobleem opgelost, en kunnen experts zelf aan het werk gezet worden om een kennisbank te ontwerpen en vullen. Dit stelt wel enige eisen aan het analytisch vermogen van deze niet-informatici en er gaat ook hier een bovengrens spelen aan de grootte van de kennisbank. Daarnaast blijft een aantal andere problemen spelen. Zo is het lastig kennis zodanig vast te leggen dat hij algemeen toepasbaar is. De kennisbank in figuur [Fig3] gaat ervan uit dat de kamer die onderzocht wordt slechts één aansluitende kamer heeft. Als de te onderzoeken kamer twee aansluitende kamers zou hebben, moet de kennisbank gewijzigd worden zoals weergegeven in figuur [Fig4].

<i>brandblusser-installeren</i>		
<u>kamer-diagnose</u>	brand-veilig	brand-onveilig
brand-blusser-installeren	nee	ja

<i>kamer-diagnose</i>					
kamer-bevat-open-vuur	ja	nee			
kamer-muurdikte	-	dik	dun		
<u>eerste-aansluitende-kamer-diagnose</u>	-	-	brand-onveilig	brand-veilig	
<u>tweede-aansluitende-kamer-diagnose</u>	-	-	-	brand-onveilig	brand-veilig
kamer-diagnose	brand-onveilig	brand-veilig	brand-onveilig	brand-onveilig	brand-veilig

<i>eerste-aansluitende-kamer-diagnose</i>				
eerste-aansluitende-kamer-bevat-open-vuur	ja	nee		
eerste-aansluitende-kamer-type	-	slaapkamer		ELSE
eerste-aansluitende-kamer-bewoners-roken-in-bed	-	ja	nee	-
eerste-aansluitende-kamer-diagnose	brand-onveilig	brand-onveilig	brand-veilig	brand-veilig

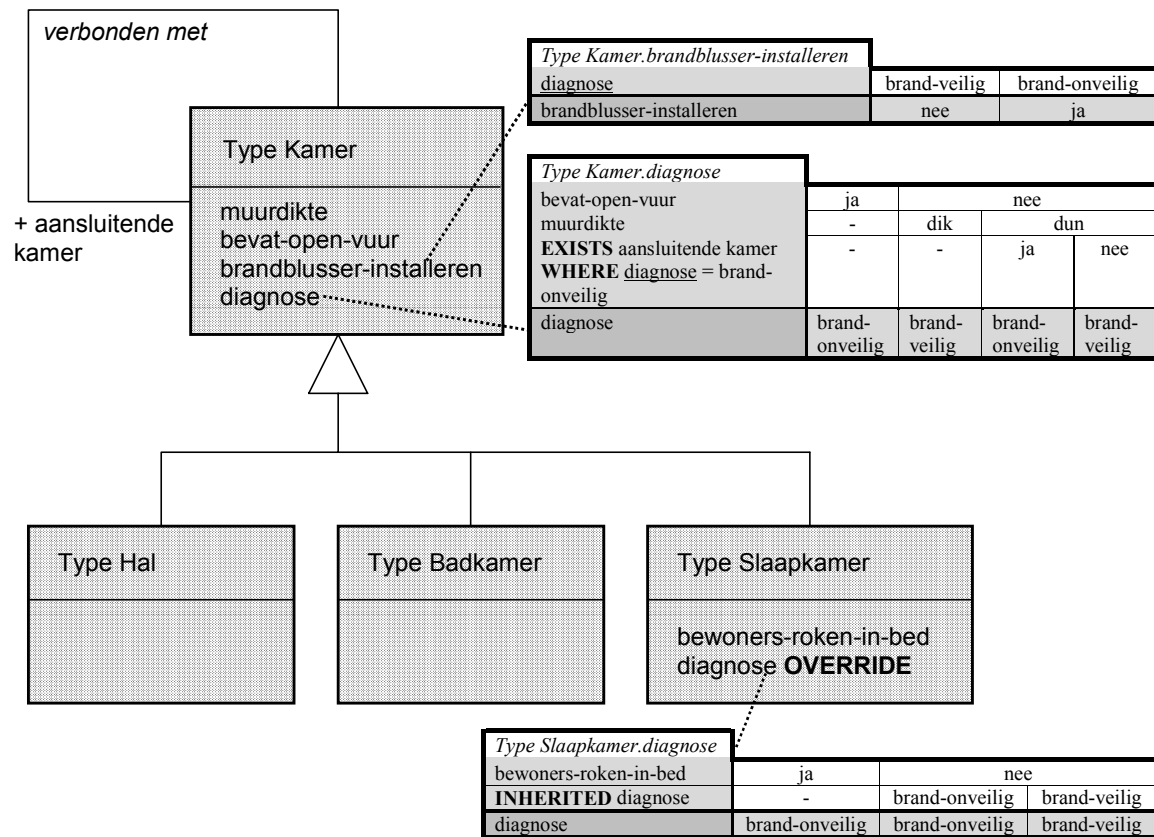
<i>tweede-aansluitende-kamer-diagnose</i>				
tweede-aansluitende-kamer-bevat-open-vuur	ja	nee		
tweede-aansluitende-kamer-type	-	slaapkamer		ELSE
tweede-aansluitende-kamer-bewoners-roken-in-bed	-	ja	nee	-
tweede-aansluitende-kamer-diagnose	brand-onveilig	brand-onveilig	brand-veilig	brand-veilig

Figuur [Fig4]: De kennisbank van figuur [Fig3], aangepast zodat er twee aansluitende kamers aan de te onderzoeken kamer ondersteund worden. We zien hier nogal drastische wijzigingen. De tabel "kamer-diagnose" is uitgebreid met een extra conditie, en de tabel "aansluitende-kamer-diagnose" is gekopieerd en komt nu twee keer voor, met dezelfde kennis, maar net iets andere benamingen.

Wat we moeten constateren, is dat door het toevoegen van de mogelijkheid om een kamer te laten aansluiten op twee kamers in plaats van één, de beslissingstabellen waarin de kennis is vastgelegd veranderen, terwijl de kennis zelf in feite niet veranderd is. En dan spreken we nog niet over de mogelijkheid dat de kamer grenst aan drie kamers, of aan kamers met speciale uitzonderingen. En het feit dat de "aansluitende kamer" zelf ook weer aansluitende kamers heeft, wordt geheel genegeerd. De reden dat deze problemen ontstaan, is dat beslissingstabellen, net zoals vrijwel alle andere technieken op het gebied van kennissystemen, kennis vastleggen in een zogenaamd "instantiemodel". Er kan geen kennis worden vastgelegd die geldig is voor alle kamers, kennis moet worden vastgelegd over specifieke instanties van een kamer. Het *concept* "kamer" is geen onderwerp van de kennisbank. We hebben daarom een nieuwe methode nodig voor de representatie van de kennis. Object oriëntatie, een geaccepteerde methode uit de software engineering die onder meer gebleken is nuttig te zijn bij communicatie met gebruikers en experts, biedt daarvoor perspectieven (Kaindl, 1994).

Object oriëntatie en kennissystemen

In een object georiënteerd systeem zou het model van de hypothetische brandveiligheidskennis uit het voorgaande er uit kunnen zien als weergegeven in figuur [Fig5]. We zien in dit model dat er kennis vastligt over het concept "kamer" in het algemeen. Deze kennis geldt in principe voor iedere kamer in het gebouw. We zien dat kamers met elkaar verbonden kunnen zijn, we zien dat kamers eigenschappen hebben, en we zien dat het al dan niet brand-onveilig zijn van een kamer bepaald kan worden volgens een algemeen geldende redenering, die de aansluitende ruimtes in acht neemt. Niet elke kamer kent echter dezelfde bepalingsmethode. Voor een slaapkamer gelden er bijvoorbeeld speciale regels. Van het abstracte type "kamer", kunnen diverse speciale types kamers afstammen. Zo is er het type "hal", dat in niets verschilt van de standaard kamer. Er is ook het type "slaapkamer", dat in zoverre verschilt dat het, behalve de eigenschappen die iedere kamer heeft, de extra eigenschap "bewoners-roken-in-bed" heeft, en een afwijkende manier om te bepalen of de slaapkamer brand-onveilig is.



Figuur [Fig5]: Het object georiënteerde model van de kennis weergegeven in figuur [Fig1], waarbij het nu mogelijk is een kamer te voorzien van een willekeurig aantal aansluitende kamers zonder dat de kennisbank hoeft te wijzigen. Dit is een zogenaamd "type model". Er is een abstract type "Kamer" gedefinieerd met vier eigenschappen ("muurdikte", "bevat-open-vuur", "brandblusser-installeren" en "diagnose"). Twee van deze eigenschappen hebben methodes gedefinieerd om hun waarde te bepalen ("brandblusser-installeren" en "diagnose"; de andere twee zouden bijvoorbeeld aan de gebruiker van het systeem gevraagd kunnen worden indien ze benodigd zijn). Van dit type zijn drie nieuwe types afgeleid. Al deze types "overerven" de eigenschappen en methodes van het type Kamer. Het type "Slaapkamer" voegt nog een extra eigenschap toe ("bewoners-roken-in-bed"), en vervangt de "diagnose"-methode van het type Kamer door een nieuwe methode.

Het getoonde model is een zogenaamd "type model". Het legt een visie op het kennisdomein vast in abstracte types, met redeneervormen (methodes) om lokaal conclusies te trekken over de eigenschappen van de abstracte types. Bij de opzet van een nieuw kennissysteem zal de kennistechnoloog in samenspraak met de expert meestal een dergelijk type model uittekenen, omdat het een duidelijke visie biedt op het kennisdomein. Een object georiënteerd kennissysteem laat het type model een expliciet onderdeel zijn van de kennisbank, en laat redeneringen vastleggen in en over het type model. Tijdens de consultatie van het systeem wordt, op aanwijzingen van de gebruiker, een zogenaamd "instantie model" gecreëerd. Het instantie model vertegenwoordigt het specifieke probleem van de gebruiker, bijvoorbeeld een hal met daaraan grenzend twee slaapkamers en een badkamer. De kennis zelf verandert niet als er meer instanties van een type gecreëerd worden.

Het voornaamste voordeel van een object georiënteerd kennissysteem ten opzichte van een conventioneel kennissysteem, is dat de structuur van de kennis, uitgedrukt in het type model, afzonderlijk vastligt van de kennis implementatie (de redeneerregels, formules, cases, etcetera). Omdat deze object georiënteerde structuur visueel wordt gepresenteerd, kan de expert duidelijke aansluiting vinden bij de kennis in de kennisbank en begrijpen hoe de kennisbank in elkaar zit. Hij heeft hiermee voldoende inzicht om de kennis te wijzigen en toevoegingen aan de kennisbank te doen, zoals bijvoorbeeld het definiëren van nieuwe soorten kamers. Dit vergroot niet alleen de onderhoudbaarheid van de kennis doordat kennis implementatie lokaal vastligt (dat wil zeggen, bij de types waartoe de kennis behoort), maar biedt ook de mogelijkheid willekeurige redeneervormen te combineren. Het maakt niet uit of we een beslissingstabel gebruiken om de aansluitende kamers te bepalen, of het rechtstreeks aan de gebruiker vragen, of een CAD-systeem uitlezen, of een case-base raadplegen. Omdat de structuur van de kennis niet verweven is met de redeneervormen, kan het raadplegen van de kennis, om een bepaalde conclusie te trekken, zonder problemen willekeurige combinaties van redeneervormen toepassen om lokale eigenschappen te bepalen.

Conclusies

Concluderend kan gesteld worden dat het toepassen van object oriëntatie bij de bouw van kennissystemen, gecombineerd met visuele representatie van kennis, de volgende voordelen oplevert ten opzichte van de conventionele methodes:

- Visuele representatie van de kennis vergroot de begrijpelijkheid van de kennis;
- Object oriëntatie maakt de kennisbank overzichtelijker en maakt dat deze beter aansluit bij de belevingswereld van de expert;
- Object oriëntatie versterkt de modularisering en encapsulatie van de kennis waardoor het duidelijk is waar en hoe de kennis verandert moet worden bij onderhoud;
- Object oriëntatie biedt meer en sterkere uitdrukkingmogelijkheden voor kennis, met name door de introductie van verschillende abstractieniveaus;
- Object oriëntatie maakt het mogelijk flexibel om te gaan met redeneervormen;
- Object oriëntatie maakt het mogelijk het type model, dat gebruikt wordt bij communicatie met de experts, een expliciet onderdeel te laten vormen van de kennisbank. Ofwel, het gebruik van object oriëntatie bij het bouwen van kennisbanken breidt het traditionele "representeren van kennis" uit met het "modelleren van kennis".

Door bovenstaande voordelen wordt de kennis onderhoudbaarder. Dit maakt het mogelijk dat de kennisbank in grote mate door de expert zelf onderhouden wordt. Dit drukt de kosten van het opzetten en beheren van een kennissysteem aanzienlijk, verbetert de betrokkenheid van de experts, en heeft een positief effect op de kwaliteit van de kennis.

De huidige stand van zaken is helaas zo dat tools die een visuele representatie van kennis koppelen aan een object georiënteerd model zich nog steeds in de research fase bevinden,

ondanks het feit dat in de literatuur al jaren lang geroepen wordt dat dit zeer belangrijke, welhaast onmisbare voordelen oplevert (Luger 1997, par. 6.5). Dit soort tools is echter momenteel wel in ontwikkeling, en de verwachting is dat de komende jaren één of meerdere van deze tools commercieel beschikbaar zullen komen.

Literatuur

Kaindl, H. (1994). *Object Oriented Approaches in Software Engineering and Artificial Intelligence*. In het "Journal of Object Oriented Programming", januari 1994, pp. 38-45.

Luger, George F. & Stubblefield, William A. (1997). *Artificial Intelligence, Structures and Strategies for Complex Problem Solving*, derde editie. Addison Wesley Longman, Inc., Reading, USA.

Document versie 2 juni 1999