

# Expertsystemen: Toepassingen - ontwikkeling - gevolgen voor de organisatie (deel II)

Prof. W. Hartman

## 1 Voorwoord

Onder de verzameltitel 'Expertsystemen: toepassingen - ontwikkeling en gevolgen voor de organisatie' worden achtereenvolgens in drie artikelen de volgende thema's besproken:

- 1 Inleiding tot expertsystemen en toepassingen van expertsystemen in de accountancy.
- 2 Het ontwikkelen en invoeren van expertsystemen.
- 3 De gevolgen van expertsystemen voor de organisatie: zowel management als individuele gebruikers. Dit derde artikel sluit af met de bespreking van enige nieuwe ontwikkelingen en met een literatuuroverzicht voor alle drie artikelen.

Deze artikelenserie is een vertaling en tevens bewerking van een voordracht, die in september 1988 in Amsterdam werd gehouden voor de International Conference 'Using Expert Systems by Accountants/-Auditors'. Ik dank Peter Groeneveld, student-assistent van de vakgroep ACAB (Erasmus Universiteit Rotterdam), die in eerste instantie voor de vertaling heeft zorggedragen.

Het eerste artikel werd opgenomen in het MAB januari/februari 1990; dit tweede artikel over het ontwikkelen en invoeren van expertsystemen is als volgt ingedeeld:

- 2 De mensen.
- 3 Projectplanning.
- 4 Ontwikkelingsmodellen.
- 5 Hulpmiddelen voor de ontwikkeling.
- 6 Nawoord.

## 2 De mensen

Bij het ontwikkelen van een expertsysteem kunnen verschillende rollen worden onderscheiden:

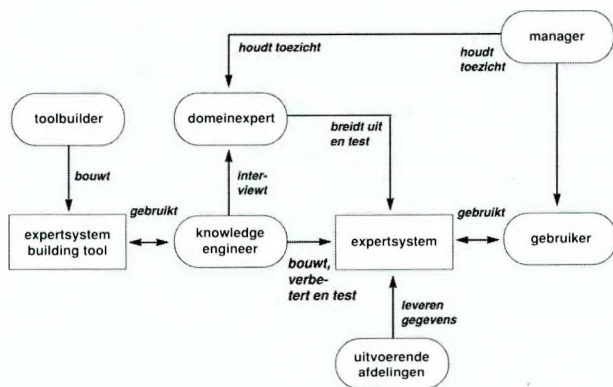
- a Elk accountantskantoor heeft minimaal een persoon nodig die zich toelegt op het ontwikkelen van expertsystemen. Deze *knowledge engineer* moet een gedegen kennis van de mogelijkheden van expertsystemen hebben en moet als tussenpersoon kunnen optreden tussen de expert en het te ontwikkelen expertsysteem. Een probleem ontstaat wanneer zo iemand niet aanwezig is in de organisatie. Inhuren van buiten de organisatie is kostbaar en heeft als nadeel dat de benodigde kennis niet binnen de organisatie zelf wordt opgebouwd.
- b Ten tweede heeft men een zogenaamde *domainexpert* nodig. Deze moet bereid zijn om zijn kennis met het expertsysteem te willen delen.
- c Ten derde is een *senior partner* of manager nodig om er op toe te zien dat de *knowledge engineer* niet te veel invloed gaat uitoefenen. Het expertsysteem moet in de behoeften van de organisatie voorzien.

Wij hebben tot nu toe 'knowledge engineer' onvertaald gelaten. Men kan als vertaling systeembouwer gebruiken. De *knowledge engineer* moet kunnen werken met de hoge mate van onduidelijkheid die inherent is aan het ontwikke-

W. Hartman is in de organisatie-adviespraktijk werkzaam als venoot van P&H Groep. Tevens is hij hoogleraar Accountancy aan de Erasmus Universiteit Rotterdam. Sinds 1962 publiceert hij regelmatig over Automatisering en Informatieverzorging.

len van expertsystemen. In de meeste gevallen is de Afdeling Informatiesystemen niet betrokken bij het ontwikkelen van expertsystemen. Expertsystemen ontstaan in specifieke domeinen binnen organisaties en in de adviesafdelingen van accountantsmaatschappen. Figuur 1 illustreert de rollen met betrekking tot expertsystemen. Het is een figuur van [Waterman85] die aangepast is volgens [Poelma87]. Minimaal vier van de zes rollen in deze figuur kunnen door accountants worden vervuld. De vierde is, naast de drie bovengenoemde, *de gebruiker*.

Figuur 1: Rollen bij het ontwikkelen van expertsystemen



### 3 Projectplanning

Enkele belangrijke regels voor de projectplanning voor de ontwikkeling van een expertstelsysteem zijn:

- Investeer eerst, verwacht geen resultaat op korte termijn.
- Selecteer een kleine en zo mogelijk eenvoudige toepassing: de betrokken partijen kunnen dan eerst ervaring opdoen.
- Het domein of probleemgebied moet relatief stabiel zijn. Het heeft geen zin om informatie over technische specificaties van personal computers in een expertstelsysteem op te nemen.
- Controleer of het probleem in woorden kan worden beschreven en of het systeem dit in korte tijd kan oplossen (in minuten of op zijn hoogst uren, maar niet in dagen). Zelfs vele technische problemen vereisen ook andere menselijke zintuigen zoals zicht en gevoel.

- Een overtuigende aanleiding voor het introduceren van expertsystemen in een organisatie is de noodzaak tot het documenteren van de expertise van een senior specialist (= domeinexpert) die zijn pensioendatum nadert. Aldus blijft zijn kennis behouden.
- In het algemeen heeft de organisatie een knowledge engineer nodig om de menselijke kennis te kunnen converteren naar de kennisbank van het expertstelsysteem.
- Houdt tijdens het ontwikkelproces de technische risico's en de noodzakelijke organisatorische aanpassingen goed in het oog. Creëer geen te hoge verwachtingen en zorg voor de medewerking van het management.

### 4 Ontwikkelingsmodellen

Ik ben twee verschillende ontwikkelingsmodellen voor expertsystemen tegengekomen:

- Prototyping, ofwel 'trial and error'-model.
- Het traditionele systeemanalysemodel.

#### a Prototyping

Als het ware voorbijgaand aan de complexiteit van het probleemgebied, past men bij dit model als regel weinig of geen zorgvuldige voorbereidingen toe. Zo blijft het definiëren van concrete, complete en gedetailleerde specificaties, dat bij een traditioneel ontwikkelingsproject voor een informatiesysteem een vereiste is, min of meer achterwege.

Prototyping, of stapsgewijze ontwikkeling, blijkt een natuurlijk model voor het ontwikkelen van expertsystemen te zijn. Het past bij het motto: 'Het belangrijkste doel is het voortschrijdend vergaren van kennis'.

Behalve de aan expertsystemen verbonden risico's die in paragraaf 4 van het derde artikel worden behandeld, wijs ik hier op twee aspecten die van belang zijn voor de systeemontwikkeling:

- De knowledge engineer-paradox [Waterman85].
- Het gebrek aan structuur.

De knowledge engineer-paradox wijst op het algemene verschijnsel dat veel domeinexperts zich er niet van bewust zijn hoe zij tot een beslis-

sing komen. Zij denken er niet over na. Martinez beschreef een soortgelijke waarneming als volgt: 'Schaakmeesters hebben waarschijnlijk geen groot inzicht in hun eigen denkproces' [Martinez88]. Dit verschijnsel veroorzaakt gedurende de analyse veel problemen.

'Gebrek aan structuur' wijst op het feit dat prototyping een bottom-up methode is: er is geen totaalstructuur beschikbaar om de elementen, die gedurende de analyse worden verzameld, in te passen.

De *fasering* van prototyping als ontwikkelingsmodel voor een expertsysteem kan als volgt worden gekarakteriseerd:

- initiëren: definiëren van het probleemgebied, vormen werkteam;
- creëren van een experimenteel proefsysteem, bevattende een beperkt aantal componenten;
- testen van het proefsysteem. Enkele testmethoden zijn:
  - gebruik van voorgeselecteerde testgevallen;
  - vergelijken met oplossingen van vroegere problemen;
  - analyseren van systeemlogica, evalueren van de regels;
  - beoordeling door andere experts ('peer review');
- één of meer iteraties voor het uitbreiden en verbeteren van het proefsysteem;
- nieuwe interimevaluatie;
- proefneming: praktijktest. De gebruiker moet vertrouwen krijgen in de resultaten van het systeem, anders zal het niet worden gebruikt. Het expertsysteem moet de zwaarste toetsen ('worst cases') ondergaan die de gebruiker kan bedenken;
- acceptatie van het voorlopige expertsysteem;
- regularisatie: verbeteren van het expertsysteem. Om de 'robuustheid' te vergroten worden beveiligingsmaatregelen in het systeem ingebouwd. Voorbeelden:
  - invoercontroles;
  - documenteren van de wijze waarop conclusies worden afgeleid;
  - vereist manueel bevestigen van beslissingen;

- vergelijken met achtergrondinformatie in de kennisbank van het expertsysteem;
  - implementeren van het 'uiteindelijke' expertsysteem;
  - periodiek evalueren van het systeem.
- Zie [Kuong88] voor een vergelijkbare aanpak.

## *b Traditionele analyse*

Het traditionele systeemanalysemodel voor het ontwikkelen van expertsystemen telt ook aanhangers. Evenals de systeemanalist die huidige en potentiële gebruikers interviewt, zal ook de knowledge engineer de domeinexpert(s) interviewen om hun kennis te converteren naar het expertsysteem. Martinez noemt een aantal technieken die als volgt in een top-down benadering kunnen worden toegepast:

- b1 Inhoudschema ('content diagram'): definieert het bereik van het expertsysteem. Het geeft de gegevensstroom aan tussen het systeem en de gebruikers. Deze stroom is bidirectioneel: gegevens van de gebruikers naar het systeem en antwoorden op vragen van het systeem.
- b2 Gegevensstroombiagram: geeft in voortschrijdende detaillering de uitwisseling van informatie tussen gebruiker en systeem, en op lagere niveau's, tussen processen en kennisbank weer.
- b3 Procesbeschrijving: drie verschillende technieken worden aanbevolen. De keuze is afhankelijk van de omstandigheden:
  - beslissingsbomen worden aanbevolen, ten eerste als het testen van een conditie afhankelijk is van de uitkomst van een vroegere toestand of conditie, en ten tweede als het aantal acties in verhouding tot het aantal condities relatief klein is;
  - beslissingstabellen worden aanbevolen als verschillende conditiecombinaties leiden tot eenzelfde actie of actiepatroon;
  - gestructureerd taalgebruik ('structured English'), als derde techniek voor procesbeschrijving, is bruikbaar voor relatief simpele If-Then-Else-stappen. Zogenaamde 'geneste If's' mogen voorkomen, maar meer dan drie geneste If-condities kunnen beter worden beschreven met behulp van een

beslissingsboom of beslissingstabel [Martinez88].

Zie voor een algemene beschouwing over vier modellen voor systeemontwikkeling [Hartman88].

## 5 Hulpmiddelen voor de ontwikkeling

Twee computertalen zijn vanaf het begin van belang voor expertsystemen: LISP in de Verenigde Staten en PROLOG in Europa. Deze talen zijn zeer geschikt voor het manipuleren van zowel symbolen als getallen. De mogelijkheid om symbolen te verwerken is belangrijk omdat kennis normaal gesproken in de vorm van if-then-conditions in expertsystemen wordt opgenomen. Bijvoorbeeld:

IF het kasstroomniveau van de cliënt is hoog,  
AND de cliënt aanvaardt een hoog risico,  
THEN beleggen in groeifondsen is sterk aanbevolen.

Hoe begint men met een expertstelsel? Met een commercieel softwarepakket dat een lege huls ('shell') bevat of met een speciaal voor een gebruiker ontworpen systeem? Dit laatste vergt veel meer tijd en een grondige kennis van programmeren met behulp van een specifieke taal, maar zal resulteren in een systeem dat voldoet aan specifieke wensen van de organisatie.

Het commerciële pakket biedt andere aspecten: geringere investering, ontwikkeling in relatief korte tijd en de mogelijkheid dat het ontwikkelen door de domeinexpert zelf kan worden gedaan. Een knowledge engineer en programmeurs zijn dan niet nodig.

Voorbeeld: ESIE of Expert System Inference Engine, een produkt van Lightwave Comp., Tampa Fl., een huls voor een expertstelsel voor MS-DOS microcomputers (US \$ 250,-).

Het aantal pakketten als hier bedoeld is groeiende [Schijff86].

## 6 Nawoord

Dit was het tweede van drie artikelen over expertsystemen. Zoals reeds gemeld, worden in de laatste bijdrage de consequenties voor de organisatie besproken. De literatuurverwijzingen voor alle drie de artikelen worden aan dat derde artikel toegevoegd.