

# Een expert systeem voor het gebruik van een software begrotingsmodel

**Citation for published version (APA):**

Cuelenaere, A. M. E., Genuchten, van, M. J. I. M., & Heemstra, F. J. (1987). Een expert systeem voor het gebruik van een software begrotingsmodel. *Informatie*, 29, 602-612.

**Document status and date:**

Gepubliceerd: 01/01/1987

**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](http://www.tue.nl/taverne)

**Take down policy**

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

[openaccess@tue.nl](mailto:openaccess@tue.nl)

providing details and we will investigate your claim.

# Een expert systeem voor het gebruik van een software begrotingsmodel

ir. A.M.E. Cuelenaere, ir. M.J.I.M. van Genuchten en ir. F.J. Heemstra

Het begroten van softwareprojecten blijkt een moeilijke opgave te zijn. Pas de laatste tien jaar worden er serieuze pogingen ondernomen het begroten te ondersteunen door het gebruik van begrotingsmodellen. De kwaliteit van de begrotingsresultaten, die met dergelijke modellen wordt gerealiseerd, wordt in belangrijke mate bepaald door de manier waarop het model voor gebruik gecalibreerd wordt op de omgeving waarin het model gebruikt gaat worden.

Alleen het calibreren blijkt in de praktijk moeilijk te zijn. Een ruime ervaring in het gebruik van het begrotingsmodel blijkt noodzakelijk. Ervaring die natuurlijk niet aanwezig is bij de beginnende modelgebruiker. In dit artikel wordt aangegeven hoe een expert systeem voor de onervaren gebruiker een hulpmiddel kan zijn bij het calibreren van een begrotingsmodel; in dit geval het model PRICE SP. Bovendien worden de ontwikkeling van en de ervaringen met een prototype van zo'n expert systeem beschreven.

## 1 Inleiding

Het begroten van softwareprojecten is moeilijk. Een eerste oorzaak hiervan is het grote aantal factoren dat van invloed is op een softwareproject. De invloed van veel van deze factoren is moeilijk te kwantificeren. Een tweede oorzaak is het gebrek aan gegevens van voltooide projecten. Het begroten komt immers altijd neer op het vergelijken van het te begroten project met voltooide projecten. Voor het begroten van softwareprojecten maakt men gebruik van begrotingsmodellen. Het is noodzakelijk een model aan te passen aan de omgeving waarin het gebruikt gaat worden: het model dient gecalibreerd te worden. Bij de calibratie worden aan één of meerdere modelvariabelen waarden toegekend. Die waarden worden afgeleid van projecten die gerealiseerd zijn in de omgeving waarin het model gebruikt gaat worden. Een onjuist uitgevoerde calibratie beïnvloedt de kwaliteit van de daarna volgende begrotingen met behulp van het model in negatieve zin.

Het calibreren van een model is een probleem: calibratie is de eerste kennismaking met het model, terwijl ervaring nodig is in het gebruik van het model om goed te kunnen calibreren. Een mogelijke oplossing voor dit probleem is de benodigde ervaring voor het calibreren van het model door middel van een expert systeem ter beschikking te stellen aan de onervaren gebruiker. Dit artikel beschrijft de ontwikkeling van zo'n expert systeem voor het binnen Philips gebruikte begrotingsmodel PRICE SP.

In paragraaf 2 wordt in grote lijnen het model PRICE SP beschreven en wordt aangegeven op welke problemen men stuit bij het calibreren van het model. Vervolgens wordt in paragraaf 3 uiteengezet welke rol een expert systeem kan spelen bij het oplossen van

deze problemen. In paragraaf 4 tenslotte wordt de ontwikkeling van het prototype van een dergelijk expert systeem beschreven en wordt toegelicht dat de eerste ervaringen met het prototype aanleiding vormen om op de ingeslagen weg verder te gaan. Het artikel wordt afgesloten met conclusies en een aantal aanbevelingen voor verder onderzoek (paragraaf 5)

Het onderzoek is uitgevoerd in een samenwerkingsverband tussen de Technische Universiteit Eindhoven (Faculteit Bedrijfskunde, vakgroep Bestuurlijke Informatiesystemen en Automatisering) en Philips (de afdeling EDP-Industriële Toepassingen en de afdeling Corporate Organization and Efficiency). We willen hier met name de heren Sierevelt, de Jonge en Koolen bedanken voor hun medewerking.

## 2 PRICE SP

### 2.1 Achtergrond PRICE modellen

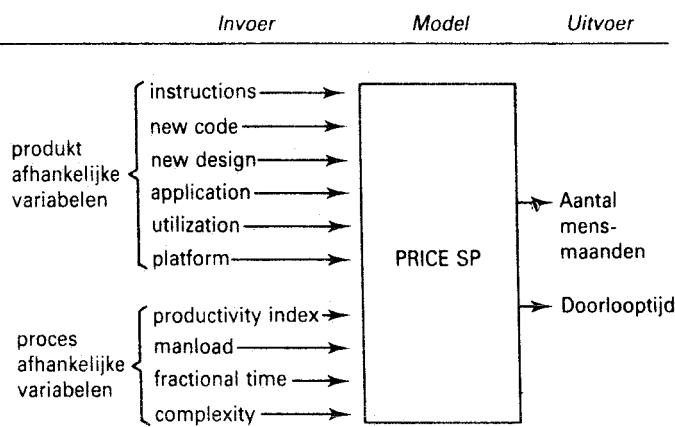
De PRICE modellen zijn modellen voor het begroten van hardware- en softwarekosten. PRICE staat voor 'Programmed Review of Information for Costing and Evaluation'. De modellen zijn ontwikkeld en worden ondersteund door RCA PRICE Systems, een onderdeel van Radio Corporation of America. RCA maakt op zijn beurt sinds korte tijd deel uit van General Electric. RCA is zo'n vijftien jaar geleden begonnen met het ontwikkelen van modellen voor het begroten van automatiseringskosten. Het eerste model dat RCA ontwikkeld heeft, is een model voor het begroten van hardwarekosten. Aangemoedigd door het succes van dit model heeft RCA een soortgelijk model voor het begroten van softwarekosten ontwikkeld. Later zijn ondermeer modellen ontwikkeld voor het begroten van onderhoudskosten en voor het begroten van het aantal coderegels van een softwareprodukt. Het model PRICE SP (Software Productivity) is het meest recente model voor het begroten van het aantal benodigde mensmaanden en de doorlooptijd van softwareprojecten. Het onderzoek dat in dit artikel besproken wordt, is gericht op dit model PRICE SP.

De inhoud van de PRICE softwaremodellen is geheim. Ook voor de huurders van het model blijft het model een black-box. De gebruiker van PRICE stuurt via een modem zijn invoergegevens naar een time sharing computer in Amerika, Engeland of Frankrijk en krijgt vrijwel direct de begroting terug. Ondanks deze beperking en de hoge huurprijs wordt het softwaremodel in de Verenigde Staten veel gebruikt. Enkele gebruikers zijn: Boeing, General Dynamics, Amerikaans ministerie van defensie, IBM, General Electric en

Texas Instruments. Het Amerikaanse ministerie van defensie eist zelfs bij offertes, die bij haar worden ingediend, een PRICE schatting. PRICE wordt in Europa minder gebruikt. Philips is in Nederland de enige grote PRICE gebruiker.

## 2.2 Het model PRICE SP

Een schema van de invoer en de uitvoer van het model is opgenomen in figuur 1.



Figuur 1: Schema PRICE SP

De uitvoer van het model bestaat in essentie uit een begroting van het aantal benodigde mensmaanden en de doorlooptijd van het project. Dit zijn de te verklaren variabelen. Zoals gemeld, is en blijft het model zelf een black-box. De invoer van het model is een karakteristiek van het te begroten project. De karakteristiek bestaat uit tien variabelen, de zogenaamde verklarende variabelen. Deze variabelen en hun definities zijn opgenomen in tabel 1. De definities zijn letterlijk overgenomen uit de Engelstalige PRICE ma-

nual [Pri85]. Omdat ze soms niet erg verhelderend zijn, is waar nodig een korte toelichting gegeven. In het verdere verloop van dit artikel worden steeds de PRICE namen van de variabelen genoemd. Dit is gedaan, omdat in sommige gevallen een Nederlands equivalent een andere betekenis suggereert en de lezer hierdoor op het verkeerde been zou kunnen worden gezet. Een voorbeeld hiervan is de PRICE variabele complexity. Een sluitende vertaling is moeilijk te geven, omdat deze variabele een veelheid aan factoren omvat (zie omschrijving in tabel 1).

## 2.3 Invoer van PRICE SP

De tien invoervariabelen zijn onder te verdelen in twee groepen (zie figuur 1). De eerste groep van zes variabelen beschrijft het te ontwikkelen produkt. Deze groep bestaat uit de variabelen instructions, new code, new design, application, utilization en platform. De tweede groep van vier variabelen beschrijft het ontwikkelingsproces, dat het gewenste produkt moet gaan opleveren. Deze groep bestaat uit de variabelen productivity index, manload, fractional time en complexity.

De variabele *instructions* is een maat voor de grootte van het programma. De gebruiker kan het aantal regels broncode of het aantal regels machinecode opgeven.

De variabelen *new code* en *new design* geven aan welk deel van het produkt geheel opnieuw ontwikkeld moet worden. Het is mogelijk, dat een deel van de code en van het ontwerp uit de literatuur of uit een vorig project gehaald kan worden. De waarde van de variabelen *new code* en *new design* zal in zo'n geval kleiner dan 1 zijn.

De variabele *application* beschrijft de soort toepassing. De gebruiker wordt geacht de waarde van de va-

Tabel 1: Invoervariabelen PRICE SP

Variabele	Omschrijving uit PRICE manual [Pri85]	Toelichting
Instructions	is the total number of deliverable, executable, machine level instructions.	grootte van het programma
New code	is the amount of new code.	waarde tussen 0 en 1
New design	is the amount of new design.	waarde tussen 0 en 1
Application	summarizes the application mix of instructions.	soort toepassing, waarde tussen 0.8 en 11 (zie tabel 2)
Utilization	is the fraction of available hardware cycle time or total memory capacity used.	waarde tussen 0 en 1
Platform	describes the planned operating environment for the software.	waarde tussen 0.6 en 2.5 (zie tabel 3)
Productivity index	is an empirically derived parameter that serves as a productivity, skill level, experience and efficiency index.	
Manload	is the average number of software personnel involved in the software project over the entire project.	
Fractional time	is the average fractional time dedicated to the software job.	de fractie van het aantal te werken uren, die men aan het project besteedt (zie tabel 4)
Complexity	describes the relative effect of complicating factors such as product familiarity, personnel software skills, hardware/software design interactions as they effect manpower costs.	

Tabel 2: Tabel voor de bepaling van de waarde van de variabele application

	WEIGHT	IDENTIFYING CHARACTERISTICS
Operating systems	10.95	Task management. Memory management. Heavy hardware interface. Many interactions. High reliability and strict timing requirements.
Interactive operations	10.95	Real time man/machine interfaces. Human engineering considerations and error protection very important.
Real time command and control	8.46	Machine to machine communications under tight timing constraints. Queuing not practicable. Heavy hardware interface. Strict protocol requirements.
On-line communications	6.16	Machine to machine communications with queuing allowed. Timing restrictions not as restrictive as with real time command and control.
Data storage and retrieval	4.10	Operation of data storage devices. Data base management. Secondary storage handling. Data blocking and deblocking. Hashing techniques. Hardware oriented.
String manipulation	2.31	Routine applications with no overriding constraints. Not oriented toward mathematics. Typified by language compilers, sorting, formatting, buffer manipulation, etc.
Mathematical operations	.86	Routine mathematical applications with no overriding constraints.

riabele application te bepalen door in tabel 2 de klasse te kiezen, die zijn project het best beschrijft. Dit is geen gemakkelijke opgave door de soms vage omschrijving van de klassen.

Eventuele hardware beperkingen worden beschreven in de variabele *utilization*. Men kan hierbij denken aan de beperkte geheugenruimte van de computer waarop het te ontwikkelen softwareprodukt moet gaan functioneren.

De laatste variabele die het project karakteriseert is *platform*. Platform beschrijft de omgeving waarin het te ontwikkelen softwareprodukt gebruikt gaat worden. In het algemeen zal voor een afdeling de waarde van de variabele platform voor de verschillende projecten gelijk zijn. De gebruiker wordt geacht de waarde van de variabele platform met behulp van tabel 3 te bepalen. Of deze tabel een algemeen bruikbaar hulpmiddel is, valt te betwijfelen. Wel is intuïtief duidelijk, dat de verschillende 'platforms' verschillende eisen aan de te ontwikkelen software stellen.

Tabel 3: Tabel voor de bepaling van de waarde van de variabele platform

Operating Environment	platform
Production Center Internally Developed s/w	0.6-0.8
Production Center Contracted s/w	1.0
MIL-Spec Ground	1.2
Military Mobile (Van or Shipboard)	1.4
Commercial Avionics	1.7
MIL-Spec Avionics	1.8
Unmanned Space	2.0
Manned Space	2.5

*Productivity index* is een variabele die door calibratie bepaald wordt op grond van een aantal voltooide projecten. De calibratie van het model en de variabele productivity index worden behandeld in paragraaf 2.4.

De variabele *manload* is in het model opgenomen, omdat mensen en tijd niet onderling uitwisselbaar zijn. Bijvoorbeeld: om hetzelfde produkt te realiseren, hebben vijf mensen (*manload* = 5) meer mensmaanden nodig dan twee mensen (*manload* = 2). Het verschil wordt onder andere veroorzaakt door de tijdrovende, onderlinge communicatie. Dit verschijnsel wordt in het bekende boek *The mythical man-month* van Brooks [Bro75] beschreven.

De variabele *fractional time* beschrijft het verschijnsel, dat versnippering van de aandacht leidt tot een lagere produktiviteit [Sie86]. Een in de softwarewereld veel voorkomend verschijnsel is, dat men zich deels met ontwikkeling van een nieuw produkt en deels met onderhoud aan vorige produkten bezig moet houden. Bijvoorbeeld: als iemand zich gedurende drie van de vijf dagen per week met een project bezig houdt, dan is de waarde van de variabele *fractional time* voor dit project 0.6.

De variabele *complexity* tot slot beschrijft projectkenmerken die vooral van invloed zijn op de doorlooptijd. De standaardwaarde van de variabele *complexity* is 1.0. Eventuele afwijkingen van die standaardwaarde moet de gebruiker bepalen aan de hand van tabel 4.

Voorbeeld: als eerder een soortgelijk project uitgevoerd is, maar de taal is nieuw voor de projectleden en de projectorganisatie is multinationalaal, dan wordt de waarde van de variabele *complexity* 1.3 (zie tabel 4). Zelfs in dit simpele voorbeeld is het overigens duidelijk, dat meerdere interpretaties van de tabel mogelijk zijn. Zo betekent een multinationalaal project (multinational project) veelal, dat het project op meer plaatsen uitgevoerd wordt (more than one location). Moet men nu de standaardwaarde van de variabele *complexity* aanpassen met 0.4 of 0.6?

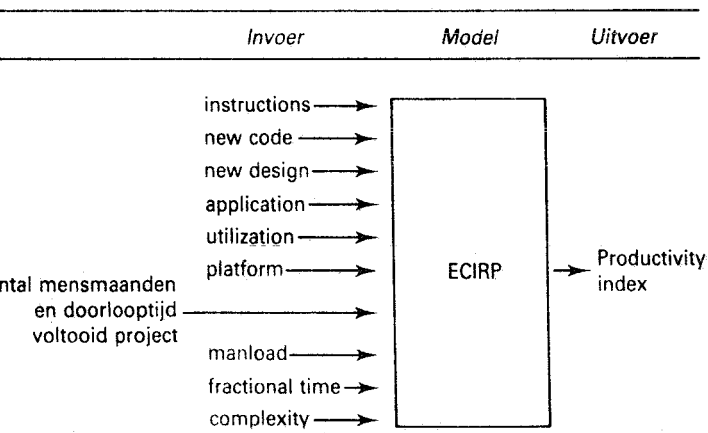
#### 2.4 Calibratie

De calibratie van het model gebeurt door een aantal

**Tabel 4: Tabel voor de bepaling van de waarde van de variabele complexity**

	<i>CPLX Adjustment</i>	<i>Voorbeeld</i>
<i>Personnel</i>		
Outstanding crew, among best in industry	-0.2	
Extensive experience, some top talent	-0.1	
Normal crew, experienced	0	
Mixed experience, some new hires	+0.1	
Relatively inexperienced, many new hires	+0.2	
<i>Product familiarity</i>		
Old hat, redo of previous work	-0.2	-0.2
Familiair type of project	-0.1	
Normal new project, normal line of business	0	
New line of business	+0.2	
<i>Complicating factors</i>		
First time with language	+0.1	+0.1
First time with processor	+0.1	
New language	+0.2 to +0.3	
New hardware	+0.2 to +0.3	
More than one location/organization	+0.2	
Multinational project	+0.4	+0.4
Hardware developed in parallel or many changing requirements	+0.2 to +0.3	
Assembly language	+0.2 to +0.3	
		+0.3

voltooide projecten met het model te beschrijven. De te verklaren variabelen zijn nu niet het aantal benodigde mensmaanden en de doorlooptijd. De te verklaren variabele is nu de variabele productivity index. Omdat het model als het ware omgekeerd gebruikt wordt, heeft RCA de calibratietoepassing van het model FCIRP genoemd. Een schema van de calibratietoepassing van het model is opgenomen in figuur 2 (zie ter vergelijking figuur 1).



**Figuur 2: Schema van de calibratietoepassing van PRICE SP**

De waarde van de variabele productivity index wordt bepaald voor zo'n vijf tot tien, door de betreffende afdeling uitgevoerde projecten. Omdat de projecten in dezelfde omgeving gerealiseerd zijn, mogen de met het model bepaalde waarden van de variabele productivity index niet te sterk van elkaar verschillen. Is aan deze voorwaarde voldaan, dan heeft men de waarde bepaald zoals die bij het begroten gebruikt zal gaan worden. Is aan deze voorwaarde niet voldaan, dan kan men nog niet met het begroten van nieuwe projecten met behulp van het model PRICE SP beginnen. Het is duidelijk, dat als er grote verschillen zijn tussen de afdelingen van een organisatie, het model voor deze afdelingen afzonderlijk gecalibreerd moet worden.

Het doel van de calibratie is het model te ijken op de omgeving waarin het model gebruikt gaat worden voor het begroten van nieuwe projecten. De noodzaak tot calibreren ontstaat onder andere door de verschillen tussen organisaties. Voorbeelden zijn verschillen in urenregistraties, in het aantal te werken uren per week, in gehanteerde overhead toeslagen en mogelijke efficiencyverschillen.

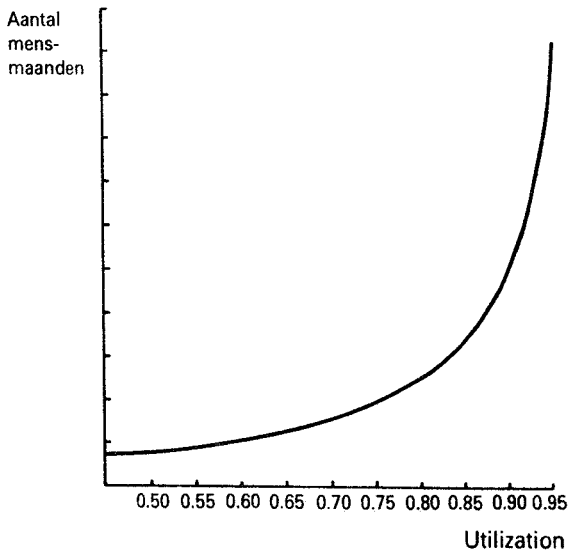
PRICE gaat ervan uit, dat men het model kan calibreren door de waarde van de variabele productivity index te bepalen uit een aantal oude projecten. De verschillen tussen de organisaties moeten tot uitdrukking komen in de waarde van die ene variabele productivity index. Door de modellering gaat informatie verloren. Wel geldt, dat in één variabele calibreren beter is dan niet calibreren. Als men bijvoorbeeld niet PRICE maar COCOMO [Boe81] gebruikt, dan zit men eigenlijk vast aan de veronderstelling dat men werkt als in een gemiddeld, Amerikaans bedrijf. COCOMO biedt namelijk geen mogelijkheid tot calibratie. De enige mogelijkheid om COCOMO te ijken op de eigen organisatie is de mogelijke waarden van de variabelen van het COCOMO model voor de eigen organisatie af te leiden. Gezien het aantal variabelen en de mogelijke verschillende waarden van de variabelen zijn hiervoor een grote hoeveelheid historische projectgegevens en een grote inspanning noodzakelijk.

## 2.5 Problemen bij calibratie

Het calibreren van het model blijkt in de praktijk problemen op te leveren. Drie oorzaken worden hier besproken. Ten eerste is de calibratie de eerste kennismaking met het model en men is dus niet gewend in de termen van het model te denken. Het juist interpreteren van de verschillende tabellen (bijvoorbeeld tabel 2, 3 en 4) is niet eenvoudig. We zijn van mening, dat de variabelen en de moeilijk te interpreteren tabellen een slechte interface vormen tussen het model en de beginnende gebruiker.

De tweede oorzaak van de problemen met de calibratie is het feit, dat de inhoud van de modellen geheim is. Daardoor is het moeilijk gevoeld te ontwikkelen voor de invloed van de verschillende waarden van de variabelen op de modeluitkomst. Als voorbeeld is een relatie tussen de variabele utilization en het aantal benodigde mensmaanden weergegeven in figuur 3. Het verschil tussen de waarden 0.8 en 0.9 heeft een veel

grotere invloed op het aantal mensmaanden dan het verschil tussen 0.6 en 0.7.



**Figuur 3: Relatie tussen de variabele utilization en het aantal benodigde mensmaanden**

De derde oorzaak van de problemen die optreden bij de calibratie is, dat de registratie van voltooide projecten soms onvoldoende gegevens bevat om de ECIRP invoervariabelen te kunnen bepalen. Het is overigens de vraag, hoe men de toekomst wil voorspellen als men het verleden niet eens kent. Het calibreren van een model mag dan problemen opleveren, het voorspellen met behulp van een model blijkt nog moeilijker. Het bepalen van het aantal instructies is bijvoorbeeld bij het calibreren een eenvoudige telling; het gaat immers om reeds voltooide projecten. Bij het begroten is het schatten van het aantal instructies daarentegen een lastig karwei.

### 2.6 Een mogelijke oplossing

De uitvoer van de calibratie is een waarde voor de va-

riabele productivity index. De invoer van de calibratie bestaat uit de waarden van elf variabelen: negen om het project te karakteriseren en twee om het gerealiseerde aantal mensmaanden en de doorlooptijd uit te drukken. De calibratie van het model kan alleen succesvol zijn, als de modelvariabelen op een consequente manier geïnterpreteerd en gewaardeerd worden. Wordt aan deze eis niet voldaan, dan loopt men het gevaar naar een schijnbaar nauwkeurige productivity index toe te rekenen. Met andere woorden: men sleutelt net zo lang aan de invoervariabelen tot het model voor verschillende voltooide projecten dezelfde waarde van de variabele productivity index vaststelt. Die index kan echter goed fout zijn met als gevolg een systematische fout in de begrotingen.

In paragraaf 2.5 is gemeld, dat de calibratie de eerste kennismaking van de gebruiker met het model is. Aan de andere kant vereist calibratie de nodige ervaring met het gebruik van het model. Het is daarom noodzakelijk de modelgebruiker bij de calibratie te ondersteunen. Een mogelijke manier om dit te doen is de gebruiker te ondersteunen met de expertise van ervaren gebruikers.

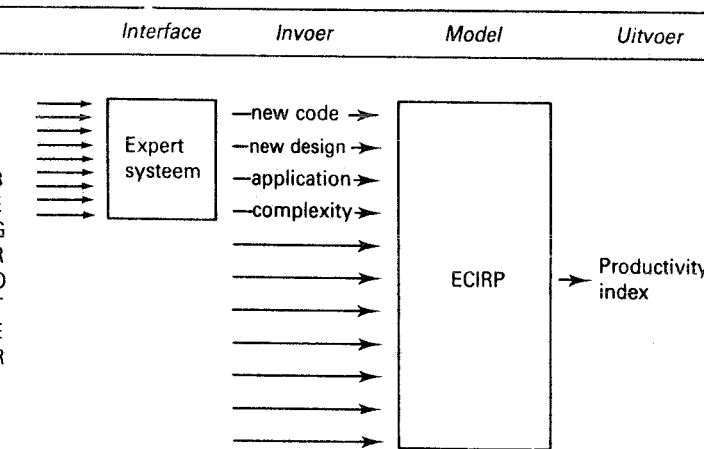
De ervaren gebruikers van het model blijken voor de bepaling van de waarde van de invoervariabelen een groot aantal heuristieken te gebruiken. Deze expertise is ontwikkeld in de jaren dat deze gebruikers met het model gewerkt hebben. Het overdragen van deze kennis en ervaring op nieuwe gebruikers blijkt echter een tijdrovende zaak te zijn. Als het nu mogelijk zou zijn de heuristieken op te slaan in een systeem, dan zou de beginnende gebruiker altijd een bruikbaar hulpmiddel bij de hand hebben. Een mogelijke vorm waarin zo'n systeem gerealiseerd kan worden, is een expert systeem.

## 3 Een expert systeem als hulpmiddel

### 3.1 De rol van het expert systeem bij calibratie

Een expert systeem is een systeem, dat in staat is om te redeneren over een bepaald kennisrijk gebied met als doel het oplossen van problemen en het geven van advies [Jac86]. Voor uitgebreide informatie over expert systemen verwijzen we naar de vele boeken en artikelen die de laatste tijd over dit onderwerp zijn verschenen (enkele voorbeelden zijn [Har85, Hay83]). Zoals gesteld in paragraaf 2.6, moet het expert systeem de beginnende gebruiker van PRICE SP de beschikking geven over expertise in het gebruik van het model. Expertise, die de expert in de loop der jaren opgebouwd heeft. Het expert systeem kan beschouwd worden als een interface tussen de begroter (ook degene die de calibratie uitvoert noemen we begroter) en de invoer van het model ECIRP. Het systeem zal niet voor alle invoervariabelen van ECIRP een waarde genereren, maar alleen voor die variabelen, waarvan het bepalen van de waarde veel expertise vraagt. De invoer van het expert systeem bestaat uit antwoorden, die de begroter geeft op vragen van het systeem. In figuur 4 is de rol van het expert systeem weergegeven.

Vanzelfsprekend zal het beantwoorden van de vragen



**Figuur 4: Het expert systeem als interface tussen begroter en ECIRP**

van het expert systeem eenvoudiger moeten zijn, dan het bepalen van de invoer van ECIRP. De kloof tussen de begroter en het model wordt op deze wijze verkleind. Gezien de noodzaak van de calibratie en de gevolgen van een onnauwkeurige calibratie, zal het expert systeem een zinnig hulpmiddel vormen bij het begroten van softwareprojecten.

Het toepassingsgebied voor het ontwikkelen van het bovengenoemd expert systeem wordt als volgt omschreven:

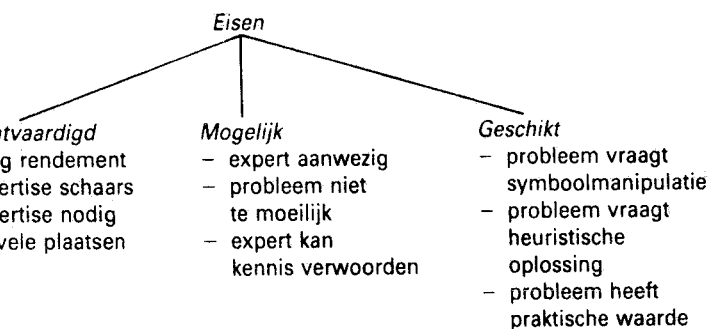
'het ondersteunen van het karakteriseren van voltooi-de projecten in de invoervariabelen van ECIRP'.

### 3.2 De toepasbaarheid van een expert systeem

In de literatuur worden verschillende eisen genoemd, waaraan een toepassingsgebied voor het gebruik van een expert systeem moet voldoen. Het door ons gedefinieerde toepassingsgebied is aan de eisen getoetst. Dit zal aan de hand van enkele belangrijke eisen worden toegelicht. Waterman [Wat86] heeft drie groepen van eisen opgesteld:

1. eisen, die de ontwikkeling van een expert systeem rechtvaardigen;
2. eisen, die de ontwikkeling mogelijk maken; en
3. eisen, die een expert systeem tot een geschikt hulpmiddel kunnen maken.

In tabel 5 zijn enkele eisen van Waterman opgenomen.



Tabel 5: Enkele eisen waaraan een toepassingsgebied moet voldoen

*Ad 1.* Het belang van calibratie is aangegeven. Het expert systeem kan leiden tot een beter gebruik van ECIRP en dus van PRICE SP. Expertise op het gebied van de invoerbepaling van ECIRP is schaars en is op meerdere plaatsen te gebruiken.

*Ad 2.* De belangrijkste eis is, dat er een expert is, die zijn kennis kan verwoorden. Voor het ontwikkelen van het door ons voorgestelde expert systeem is een expert beschikbaar, die al jaren ervaring heeft met PRICE en met calibratie in het bijzonder. Verder zijn er enkele zeer ervaren gebruikers van ECIRP. Hayes-Roth [Hay83] noemt verder als belangrijke eis, dat de expert het systeem als een nuttig hulpmiddel ziet. Ook aan deze eis is voldaan. Het probleem is niet al te moeilijk. Een intern Philips rapport noemt een taak geschikt als toepassingsgebied, als die tussen de tien minuten en de enkele uren in beslag neemt

[Ph186] Het vaststellen van de invoer van ECIRP blijkt in de praktijk ongeveer één tot twee uur te kosten.

*Ad 3.* Het oplossen van het probleem vraagt heuristische kennis en is uit te voeren met symboolmanipulatie (dit is het redeneren met symbolen die concepten uit de reële wereld voorstellen [Wat86]). Zoals al eerder genoemd, blijkt dit bij het vaststellen van de waarden van invoervariabelen het geval te zijn. De praktische waarde van het ondersteunen van het gebruik van ECIRP is dat het gebruik ervan wordt vereenvoudigd. Het begroten van softwareprojecten wordt zo ondersteund

Conclusie: het ondersteunen van het karakteriseren van voltooi-de projecten in de ECIRP invoervariabelen voldoet als toepassingsgebied aan de eisen van Waterman en aan andere eisen die in de literatuur gesteld worden. Het ontwikkelen van een expert systeem is derhalve het overwegen waard.

### 3.3 Een prototype

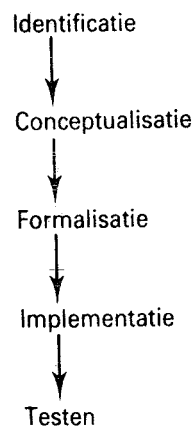
De volgende stap die gezet is, is het ontwikkelen van een prototype van het expert systeem. Zoals al vermeld, zal het expert systeem niet de invoerbepaling van alle variabelen van ECIRP ondersteunen. Zo is het niet zinvol om de bepaling van de waarde van de variabele instructions te ondersteunen, omdat dit bij calibratie alleen maar een telling is. Het expert systeem genereert numerieke waarden voor de invoervariabelen complexity, application, new design en new code (zie figuur 4).

## 4 De ontwikkeling van het prototype

### 4.1 Aanpak

Bij de ontwikkeling van het prototype is de fasering van Hayes-Roth [Haij83] gevolgd. Hij onderscheidt in het proces van kennisacquisitie de volgende fasen (zie figuur 5):

1. identificatie, het vertrouwd raken van de knowledge engineer met het probleemgebied en het karakteriseren van het probleem;
2. conceptualisatie, het ontdekken en structureren van de kennis door een nauwe samenwerking tussen expert en knowledge engineer;



Figuur 5: Fasen in het proces van kennisacquisitie

3. formalisatie, het herschrijven van de kennis in een representatievorm die gericht is op de gekozen implementatievorm;
4. implementatie, het invoeren van de kennis in het systeem;
5. testen, evalueren van het systeem.

Voor het weergeven van de structuur van de kennis in de verschillende fasen van de kennisacquisitie hebben we drie representatievormen gekozen. De eerste is onafhankelijk van de implementatie. Deze representatievorm zal verder aangeduid worden met de term beslissingsbomen. Deze representatie dient tevens als leidraad tijdens de gesprekken van de knowledge engineer met de expert. Een tweede representatievorm geeft in de vorm van een soort flow diagram de structuur van de opgeslagen kennis in het expert systeem weer. De tweede representatievorm is dus afhankelijk van de implementatie. Deze representatie is nodig om overzicht te houden over de opgeslagen kennis. Voor de implementatie zijn vele gereedschappen beschikbaar, de zogenaamde shells. Het gebruik van een shell maakt het mogelijk om op eenvoudige wijze gerepresenteerde kennis toe te voegen, te veranderen of te verwijderen. Het wordt hierdoor mogelijk om kennis stapsgewijs te implementeren. De derde representatievorm tenslotte zijn de rules zoals die in het systeem ingevoerd worden. Het prototype is namelijk ontwikkeld met een rule-based shell. Dat wil zeggen, dat de kennis in de vorm van produktieregels (if-then regels) opgeslagen wordt.

#### 4.2 Uitvoering

In het kennisdomein zijn drie gebieden te onderscheiden: kennis nodig voor de bepaling van de waarde van

- de variabele complexity;
- de variabele application;
- de variabelen new code/new design.

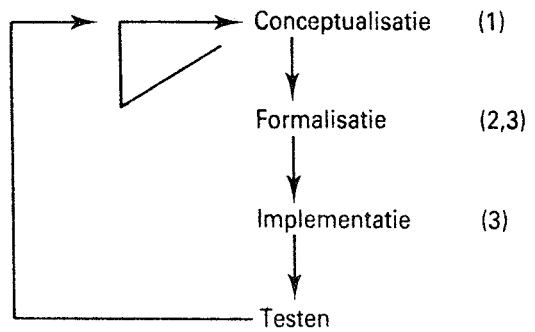
Per gebied zijn de fasen conceptualisatie, formalisatie, implementatie en testen (zie figuur 5) doorlopen.

##### Conceptualisatie

Er hebben zeventien gesprekken met de expert plaatsgevonden in een periode van drie maanden. De gesprekken zijn zorgvuldig voorbereid om de kostbare tijd van de expert efficiënt te gebruiken. De gesprekken zijn verlopen aan de hand van cases en aan de hand van door de knowledge engineer gestelde vragen. Na ieder gesprek heeft de knowledge engineer alle antwoorden en opmerkingen van de expert uitgewerkt en heeft hij getracht de opgedane kennis in een beslissingsboom te representeren. Aan de hand van deze beslissingsboom zijn weer nieuwe gesprekken gevoerd, waarbij de beslissingsboom is aangepast en uitgebreid (zie figuur 6).

##### Formalisatie en implementatie

De knowledge engineer heeft vervolgens de beslissingsboom herschreven in de tweede representatievorm. Daarna is de kennis weergegeven in rules. De rules zijn vervolgens geïmplementeerd.



Figuur 6: De doorlopen fasen bij de kennisacquisitie. De cijfers tussen haakjes verwijzen naar de representatievormen die bij de verschillende stappen gebruikt zijn

##### Testen

Om de nauwkeurigheid van de opgeslagen kennis te testen zijn we nagegaan, of het expert systeem dezelfde numerieke waarden genereert als de expert vaststelt. Hierbij moet niet alleen gelet worden op de uiteindelijke waarde, maar ook op de manier waarop deze waarde tot stand komt. De test is als volgt uitgevoerd: voor een aantal voltooidde projecten zijn de door het expert systeem gegenereerde waarden vergeleken met de waarden zoals de expert ze destijds bij calibratie heeft vastgesteld. Zo is bepaald, of de expert geen expertise gebruikt heeft, die nog niet in het systeem opgenomen is. Als de opgeslagen kennis aangepast moest worden, heeft er terugkoppeling plaatsgevonden naar de conceptualisatiefase en is de cyclus opnieuw doorlopen (zie figuur 6).

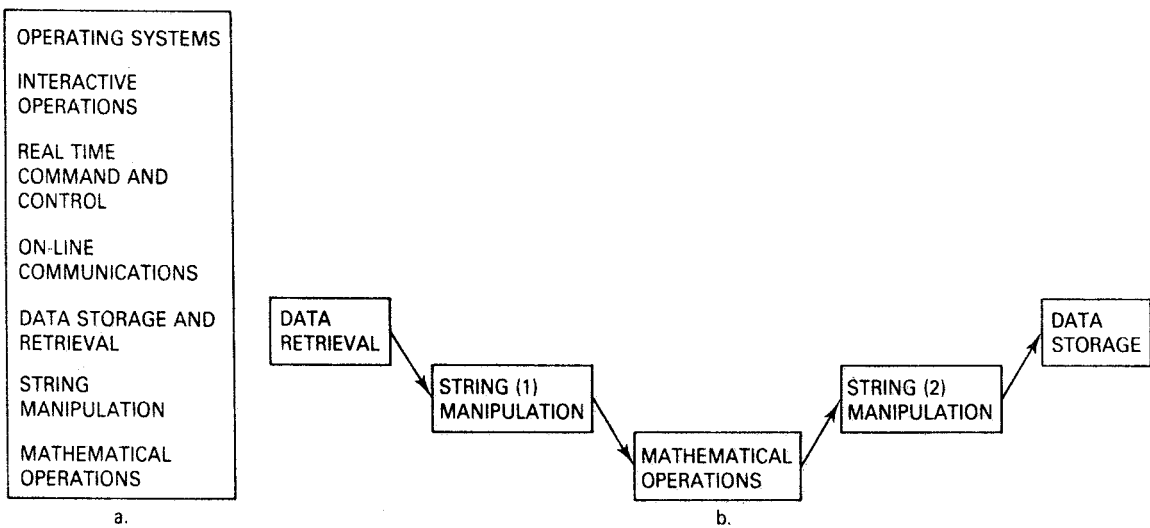
Liebowitz [Lie86] noemt naast het testen van de nauwkeurigheid van de opgeslagen kennis de gebruikerstest. Tot de gebruikerstest van het prototype is binnen dit onderzoek slechts een aanzet gegeven. Aan enkele potentiële gebruikers van PRICE SP is gevraagd om voor een voltooid project (waar ze nauw bij betrokken waren) de vragen van het systeem te beantwoorden. Hiermee is onder andere onderzocht, of de vragen juist en duidelijk zijn en of de antwoorden genuanceerd genoeg zijn. Het prototype is als gevolg hiervan een aantal malen aangepast.

#### 4.3 Resultaten

Het ontwikkelde prototype genereert waarden voor de vier invoervariabelen complexity, application, new code en new design. Het totale systeem omvat een kleine tweehonderd rules. Tijdens de gesprekken met de expert is gebleken, dat er factoren zijn die op meerdere variabelen invloed hebben. Bijvoorbeeld: als bij de bepaling van de waarde van de variabele complexity duidelijk is, dat er sprake is van ervaren ontwikkelaars die al soortgelijke systemen ontwikkeld hebben, dan is het onwaarschijnlijk, dat het ontwerp voor honderd procent nieuw is. Met andere woorden, het is onwaarschijnlijk, dat de variabele new design de waarde 1 krijgt. De deelgebieden complexity, application en new code/new design blijken dus niet te scheiden te zijn.

Twee onderdelen van het expert systeem worden na-





Figuur 7: De klassen volgens RCA en het basisproces

der toegelicht. Van de variabele application bespreken we een deel van de onderliggende expertise. Alledrie de representatievormen worden toegelicht aan de hand van een eenvoudig deel van de kennis, die nodig is bij het bepalen van de waarde van de variabele complexity. Tot slot worden de testresultaten van de nauwkeurigheid van de variabelen complexity en application besproken.

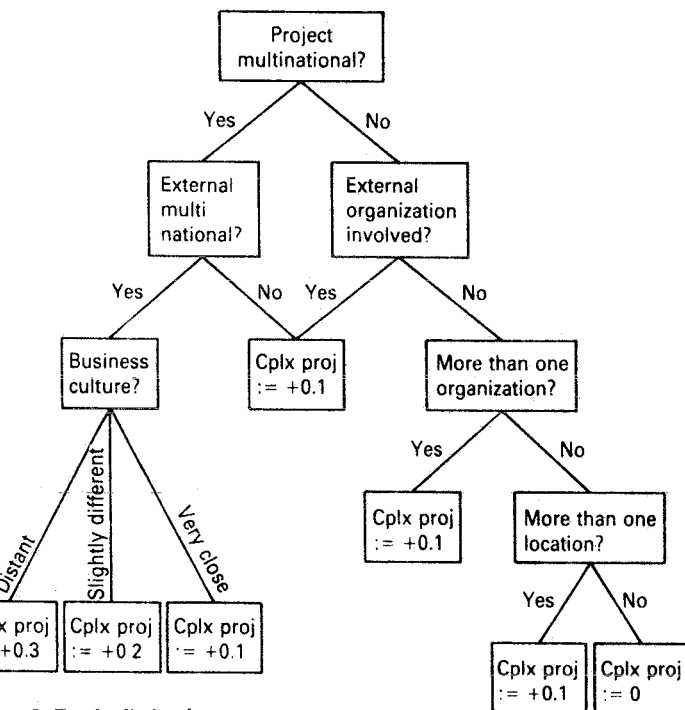
*Application*

We zullen hier één manier toelichten waarop de expert en dus het expert systeem de waarde van de variabele application bepaalt. Volgens de manual van PRICE is de bepaling van deze waarde het maken van een keuze uit zeven klassen van projecten (zie figuur 7.a). Dat het meer inhoudt dan alleen een keuze, blijkt uit de manier, waarop de expert de waarde be-

paalt. Bij het vaststellen van de waarde van de variabele application is het volgende basisproces te onderscheiden: in een programma worden gegevens opgehaald en eventueel geformatteerd om er een rekenproces mee uit te voeren. Na het rekenproces worden de gegevens weer geformatteerd en daarna ergens opgeslagen. Dit proces is weergegeven in figuur 7.b. De gebruiker wordt gevraagd aan te geven, welk percentage van zijn programma gerekend kan worden tot data retrieval, string manipulation (1), mathematical operations, string manipulation (2) en data storage. Deze vijf delen van het basisproces komen overeen met de onderste drie klassen van de application tabel van RCA (vergelijk de figuren 7.a en 7.b). Vervolgens wordt de gebruiker gevraagd, of er sprake is van communicatie met andere systemen. Zo ja, dan moet de gebruiker aangeven welke percentages van die vijf delen van het basisproces met communicatie te maken hebben. Zo is het mogelijk, dat een deel van de gegevens opgehaald wordt via communicatie met een ander systeem. Bijvoorbeeld: als de gebruiker twintig procent van het basisproces heeft geklassificeerd als data retrieval, dan kan nu de helft van die twintig procent 'opgewaarderd' worden tot de klasse communicatie. Hiermee wordt de waarde van de variabele application hoger. Daarna wordt op een zelfde manier onderzocht welk deel van het programma geklassificeerd moet worden als real time constraints en als interactive operations (zie tabel 2). Uiteindelijk resulteert het voorgaande in een opdeling van het programma in de zes klassen uit de application tabel (de klasse operating system wordt apart beschouwd). Het bepalen van de waarde van de variabele application is na deze opdeling slechts een berekening.

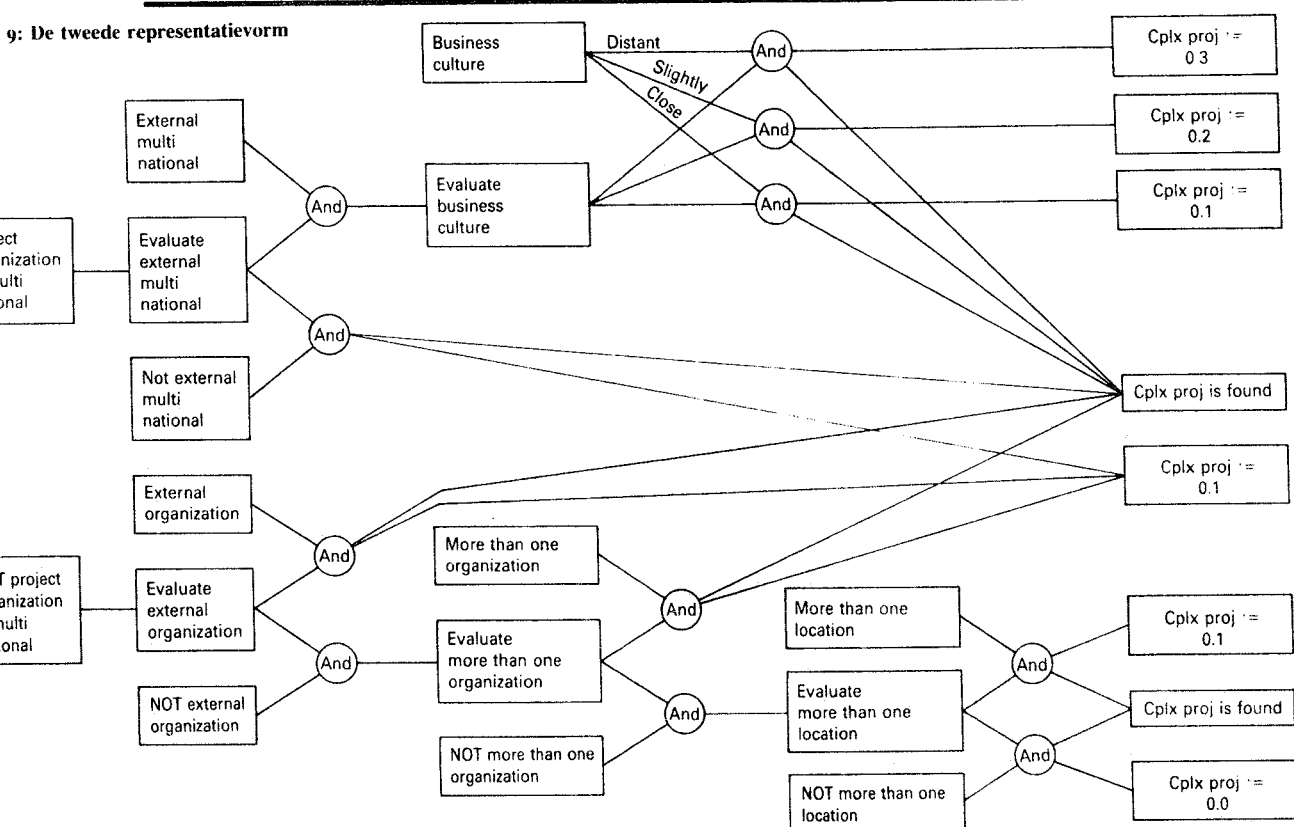
*Complexity*

Als tweede voorbeeld gaan we dieper in op een deel van de kennis, die nodig is voor de bepaling van de waarde van de variabele complexity. De drie representatievormen zullen voor dit deel worden getoond. De beslissingsboom staat in figuur 8 en spreekt voor zich.



Figuur 8: Een beslissingsboom

9: De tweede representatievorm



<p><b>RULE more than one location</b>          IF evaluate more than one location          AND more than one location          THEN cplx proj is found          AND cplx proj := 0.1</p> <p><b>RULE not more than one location</b>          IF evaluate more than one location          AND NOT more than one location          THEN cplx proj is found          AND cplx proj := 0</p> <p><b>RULE more than one organization</b>          IF evaluate more than one organization          AND more than one organization          THEN cplx proj is found          AND cplx proj := 0.1</p> <p><b>RULE not more than one organization</b>          IF evaluate more than one organization          AND NOT more than one organization          THEN evaluate more than one location</p> <p><b>RULE external organization involved</b>          IF evaluate external organization          AND external organization involved          THEN cplx proj is found          AND cplx proj := 0.1</p> <p><b>RULE not external organization involved</b>          IF evaluate external organization          AND NOT external organization involved          THEN evaluate more than one organization</p>	<p><b>RULE multi national</b>          IF project organization is multi national          THEN evaluate external multinational          ELSE evaluate external organization</p> <p><b>RULE external multinational</b>          IF evaluate external multinational          AND external multinational          THEN evaluate business culture</p> <p><b>RULE not external multinational</b>          IF evaluate external multinational          AND NOT external multinational          THEN cplx proj is found          AND cplx proj := 0.1</p> <p><b>RULE business culture distant</b>          IF evaluate business culture          AND business culture IS distant          THEN cplx proj is found          AND cplx proj := 0.3</p> <p><b>RULE business culture slightly different</b>          IF evaluate business culture          AND business culture IS slightly different          THEN cplx proj is found          AND cplx proj := 0.2</p> <p><b>RULE business culture very close</b>          IF evaluate business culture          AND business culture IS very close          THEN cplx proj is found          AND cplx proj := 0.1</p>
---	--

Figuur 10: De bijbehorende rules

Zoals gemeld, hebben de beslissingsbomen gediend als leidraad in de gesprekken met de expert. Tweederde van de inspanning van de totale ontwikkeling heeft bestaan uit het structureren en in deze vorm representeren van de kennis. De beslissingsboom is omgeschreven naar de tweede representatievorm als in figuur 9. De rules, zoals er een aantal opgenomen is in figuur 10, zijn hieruit eenvoudig af te leiden.

#### Testen complexity en application

De nauwkeurigheid van de opgeslagen kennis voor de bepaling van de waarde van de variabelen complexity en application is aan de hand van zes projecten bepaald. De testresultaten zijn opgenomen in tabel 6. In de kolommen A de waarden die de expert indertijd bepaald heeft. In de kolommen B de waarden zoals ze door het expert systeem gegenereerd zijn. De verschillen zijn aangegeven in de kolommen C. Bij de variabele complexity zijn alleen de afwijkingen van de standaardwaarde vermeld. Een verschil van minder dan 0.1 voor de waarde van de variabele complexity is goed te noemen, omdat zo'n verschil een bescheiden fout in de uitvoer van het model ECTRP geeft. Een verschil in waarde van 0.5 is acceptabel voor de variabele application (zie ook tabel 2).

Het expert systeem is op bescheiden schaal onderworpen aan een gebruikerstest. Bij enkele vragen is gebleken, dat het aantal antwoordmogelijkheden te beperkt was. Deze mogelijkheden zijn in overleg met de expert uitgebreid. Ook is uit de gebruikerstest gebleken, dat sommige heuristische plaats- of tijdgebonden zijn. Het is duidelijk, dat bij de ontwikkeling van een definitief systeem het testen van de nauwkeurigheid van de opgeslagen kennis en de gebruikerstest een belangrijke plaats zullen moeten innemen.

### 5 Conclusies

Begrotingsmodellen dienen gecalibreerd te worden. Het calibreren is moeilijk. Een belangrijke oorzaak hiervan is, dat de calibratie de eerste kennismaking van een gebruiker met het model is, terwijl calibratie toch ervaring in het gebruik van het model vraagt. Die ervaring is nodig om de invoervariabelen van het model op een consequente manier te kunnen interpreteren

en waarden. Wij zijn van mening, dat de gebruiker bij de calibratie ondersteund dient te worden met expertise van een ervaren modelgebruiker. Een middel om expertise te distribueren is een expert systeem. Wij hebben een prototype van een expert systeem ontwikkeld om het karakteriseren van projecten ten behoeve van calibratie in de PRICE SP invoervariabelen te ondersteunen. De resultaten tot nu toe en de positieve reacties van de gebruikers en de expert zijn aanleiding geweest om het onderzoek te gaan voortzetten.

Mogelijke vervolgstappen zijn:

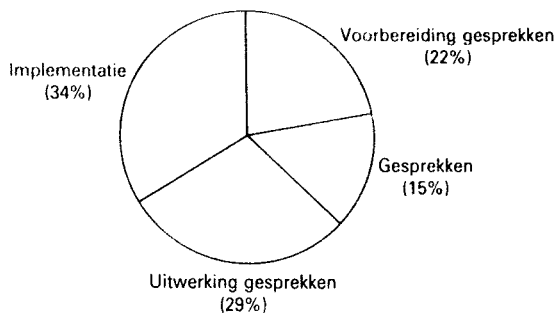
- het ontwikkelen van een *definitief expert systeem* met als basis het prototype;
- het ontwikkelen van een expert systeem als ondersteuning van begrotingsmodellen bij het begroten van *nieuwe projecten*. De expertise die nu verzameld is, kan deels bij het begroten van nieuwe projecten gebruikt worden;
- het op andere gebieden toepassen van een expert systeem als interface tussen een model of systeem en de gebruiker ervan. Op basis van de ervaringen opgedaan in dit onderzoek, lijkt het ons zinvol meer aandacht aan dit toepassingsgebied te besteden.

Het vastleggen van gegevens van voltooide projecten is de basis voor iedere begroting. Een goede registratie is belangrijker dan ieder model of iedere ondersteuning van een model. Een onderwerp, waar door de auteurs onderzoek naar gedaan wordt, is welke gegevens van historische projecten relevant zijn. Omdat we registratie belangrijk vinden, hebben we veel aandacht besteed aan het registreren van het ontwikkelen van het prototype. Uit de registratie blijkt onder meer hoeveel tijd besteed is aan de activiteiten: voorbereiding gesprekken, gesprekken met de expert, uitwerking gesprekken en implementatie van de kennis. Deze verdeling is weergegeven in figuur 11. De totale ontwikkeling van het prototype heeft ongeveer zesenzeventig mandagen gekost.

De registratie geeft ook een eerste indicatie van de verhouding in tijd die een knowledge engineer en een expert moeten besteden aan de ontwikkeling van een expert systeem. Bij het ontwikkelen van het prototype

Tabel 6: Testresultaten van de variabele complexity en de variabele application

COMPLEXITY				APPLICATION			
	A	B	C		A	B	C
PROJECT	EXPERT	EXPERT SYSTEEM	VERSCHIL	PROJECT	EXPERT	EXPERT SYSTEEM	VERSCHIL
1	0.09	0.10	0.01	1	8.04	7.50	0.54
2	0.23	0.21	0.02	2	7.28	7.24	0.04
3	0.25	0.30	0.05	3	5.41	4.93	0.48
4	-0.08	-0.10	0.02	4	5.00	4.68	0.32
5	0.58	0.50	0.08	5	7.09	6.89	0.20
6	0.90	0.70	0.20	6	4.73	4.33	0.40



N.B. De tijd besteed aan het testen is niet in deze verdeling opgenomen.

**Figuur 11: Tijdsverdeling over activiteiten**

was deze verhouding ongeveer zeven op één. De knowledge engineer heeft namelijk de vier genoemde activiteiten uitgevoerd; de expert was gesprekspartner.

Willen we iets weten over de invloed van het expert systeem op de kwaliteit van de begrotingen, dan moeten we weten hoe het met deze kwaliteit gesteld is. Deze gegevens ontbreken en vergelijken is dus niet mogelijk. Om dezelfde reden kan ook niet vastgesteld worden, wat de invloed van modellen en van registreren op de begroting is. Het voorgaande benadrukt de noodzaak tot empirische gegevens over het begroten van software-projecten.

We hebben met een expert systeem geprobeerd de invoerbepaling van de calibratie van PRICE SP te verbeteren. Een andere mogelijkheid om modelbegrotingen te verbeteren, is het aanpassen van het model zelf.

Hierbij moet men onder meer streven naar invoervariabelen, waarvan de waarden op een eenvoudige manier vastgesteld kunnen worden. Een model kan het

ontwikkelen van software nog zo goed beschrijven, als de waarden van de invoervariabelen moeilijk te bepalen zijn, hebben de modelbegrotingen en dus het model een beperkte waarde.

## 6 LITERATUUR

- [Boe81] Boehm, B.W., *Software engineering economics*, Prentice Hall, Englewood Cliffs, 1981.
- [Bro75] Brooks, F.P., *The mythical man-month, essays on software engineering*, Addison-Wesley Publ. Co., Reading, Massachusetts, 1975.
- [Har85] Harmon, P., King, D., *Expert systems, Artificial intelligence in business*, Wiley Press, New-York, 1985.
- [Hay83] Hayes-Roth, F., Waterman, D.A., Lenat, D.B., *Building expert systems*, Addison-Wesley Publ. Co., Reading, Massachusetts, 1983.
- [Jac86] Jackson, P., *Introduction to expert systems*, Addison-Wesley Publ. Co., Reading, Massachusetts, 1986.
- [Lic86] Liebowitz, J., 'Useful approach for evaluating expert systems', *Expert systems*, Vol. 3, No 2, April 1986.
- [Phi86] Philips rapport, *Guidelines for the introduction of expert systems technology*, 1986.
- [Pri85] RCA PRICE Systems, *PRICE SISP manual*, 1985.
- [Sie86] Siercvelt, H., 'Observations on software models', *Journal of Parametrics*, Vol. VI, No 4, December 1986.
- [Vli84] Vliet, J.C. van, *Software engineering*, Stenfert Kroese, Leiden/Antwerpen, 1984.
- [Wat86] Waterman, D.A., *A guide to expert systems*, Addison-Wesley Publ. Co., 1986.

*Ir. A.M.E. Cuelenaere heeft informatica gestudeerd aan de Technische Universiteit Eindhoven en is nu werkzaam bij Philips op de afdeling EDP-Industriële Toepassingen.*

*Ir. M.J.I.M. van Genuchten werkt bij de Philips afdeling Corporate Organization & Efficiency en de vakgroep Bestuurlijke Informatiesystemen en Automatisering van de faculteit Bedrijfskunde van de Technische Universiteit Eindhoven.*

*Ir. F.J. Heemstra is als universitaire docent verbonden aan de Technische Universiteit Eindhoven, faculteit Bedrijfskunde, vakgroep Bestuurlijke Informatiesystemen en Automatisering.*