

# Neue Verfahren zur Erstellung wissensbasierter Befundungssysteme mit der Expertensystemschale Pro.M.D.

## Recent methods for knowledge-based interpretation systems with the Pro.M.D. expert system shell

B. Pohl<sup>1</sup>, C. Beringer<sup>2</sup>, S. Walther<sup>3</sup>, J. Melzer<sup>3</sup>, F. Burow<sup>3</sup>, M. Schmidt-Schauß<sup>3</sup>, Chr. Trendelenburg<sup>4</sup>

<sup>1</sup>Institut für Hygiene und Mikrobiologie der Universität Würzburg

<sup>2</sup>Physikalisches Institut der Universität Heidelberg

<sup>3</sup>Fachbereich Informatik der Universität Frankfurt

<sup>4</sup>Institut für Laboratoriumsmedizin der Städt. Kliniken Frankfurt/M.-Höchst

### Zusammenfassung:

*Beim Aufbau eines wissensbasierten Befundungssystems ist die medizinische Wissensbasis die anspruchsvollste Komponente. Einige spezielle Problemgebiete sind hierbei die Behandlung zeitbezogener Relationen, die Behandlung unpräzisen oder mehrdeutigen Wissens und die Bewältigung der stark anwachsenden Komplexität größerer Wissensbasen. Im Rahmen einer Revision der Wissensrepräsentationssprache Pro.M.D. werden deshalb neue Konzepte entwickelt. Dies sind die flexible Inferenzstrategie mit Vorwärts- und Rückwärtsverkettung, die funktionale objektorientierte Wissensdarstellung, die Darstellung zeitbezogenen Wissens mit dynamischen mathematischen Modellen und die Einführung semibekannter Daten. Ziel dieser Maßnahmen ist es, den schwierigen Prozeß des Wissenserwerbs für medizinische Wissensbasen durch eine verbesserte Wissensrepräsentation und angepaßte Werkzeuge zu unterstützen.*

### Schlüsselwörter:

*Laboratoriumsmedizin – Spezialbefundung – wissensbasiertes System – Expertensysteme – Pro.M.D. – Mathematische Modelle*

### Summary:

*In constructing interpretation systems for laboratory test results, the most challenging part is the medical knowledge base. Some specific problems are handling temporal or fuzzy relations, and dealing with the phenomena of combinatoric explosion which stands for the rapidly increasing complexity of larger knowledge-based systems. Therefore, during a Pro.M.D. revision new concepts are being introduced. These are flexible inference strategies with forward and backward chaining, functional object-oriented knowledge representation, dynamic mathematical models for temporal reasoning, and semiknown data for possibilistic and probabilistic reasoning. These features will help to support the difficult task of knowledge acquisition by improved knowledge representation and adapted tools.*

### Keywords:

*Laboratory medicine – test result interpretation – knowledge-based system – expert system – Pro.M.D. – mathematical model*

## Einleitung

Das Pro.M.D.-System ist eine sogenannte Expertensystemschale (1, 2) und wurde zur Erstellung wissensbasierter Befundungssysteme für die labormedizinische Spezialbefundung konzipiert. Die Ziele des Pro.M.D.-Projektes sind die Definition einer Wissensrepräsentationssprache für die labormedizinische Spezialbefundung und die Implementierung einer entsprechenden Experten-

systemschale. Ein weiteres wichtiges Ziel ist die Entwicklung von Wissensbasen zur Interpretation von Befundkonstellationen in umschriebenen Teilgebieten der Laboratoriumsmedizin.

Wissensbasierte Befundungssysteme mit dem Pro.M.D.-System bestehen aus der Systemschale mit Wissenserwerbs-, Schlußfolgerungs-, Erklärungs- und Kommunikationsfunktion und aus den Wissensbasen, den Falldatenbanken und den Stammdatentabellen. Kritischer

Teil eines wissensbasierten Befundungssystems ist die medizinische Wissensbasis, die das erforderliche labormedizinische Fachwissen in Form von Schlußfolgerungsregeln enthält. Einige spezielle Problemgebiete sind hierbei:

- die Behandlung zeitbezogener Relationen, z.B. bei Berücksichtigung von Vorbefunden
- die Behandlung unpräzisen oder mehrdeutigen Wissens
- die Bewältigung der stark anwachsenden Komplexität größerer Wissensbasen

Die Neuentwicklung, viel mehr jedoch die Pfl egbarkeit, medizinischer Wissensbasen hängt sehr stark von einer geeigneten, problemnahen Darstellbarkeit des Fachwissens ab. Deshalb werden im Pro.M.D.-Projekt neue Verfahren entwickelt, um Pfl egbarkeit und Weiterentwicklung medizinischer Wissensbasen zu erleichtern.

Die hier beschriebenen Arbeiten wurden im Rahmen des Pro.M.D.-Projektes von Ärzten, Informatikern und Studenten verschiedener Fachrichtungen durchgeführt. Wir stellen diese Techniken hier dar, weil zunehmend weitere Kreise von Ärzten selbst Wissensbasen erstellen. Diese können mit Hilfe der Pro.M.D.-Wissensnotation ihr Wissen gleich selbst in vollständiger Form notieren.

## Methoden und Ziele

Pro.M.D. steht für Prolog-System zur Unterstützung medizinischer Diagnostik und beinhaltet unter anderem sowohl eine Wissensrepräsentationssprache als auch eine sogenannte Expertensystems chale.

Die Pro.M.D.-Schale ist ein Computerprogramm welches in der symbolischen Programmiersprache Prolog implementiert ist. Wir verwenden das Prolog-2 System der Firma Expert Systems International, Oxford, England, welches für Anwendungen auf IBM-kompatiblen Computern besonders geeignet ist.

Bei der Weiterentwicklung der Pro.M.D.-Sprache ließen wir uns von folgenden Zielen leiten:

1. Möglichst einfache Syntax und weitgehende Vermeidung von Sonderzeichen in den Regeln der Wissensbasen.
2. Orthogonales Sprachdesign. Die Mächtigkeit der Wissensrepräsentationssprache entsteht durch die Kombinierbarkeit weniger Grundstrukturen (Wissensmodule, Schnittstellen, Klassen, Methoden, Vererbung und Klassenhierarchie).
3. Klare Semantik auf der Basis allgemein akzeptierter mathematischer Verfahren wie z.B. Prädikatenkalkül, Bayes-Theorem und numerischer Lösung von Differentialgleichungen. Für die Wissensnotation sollen adhoc-Verfahren, Heuristiken und trickreiche Lösungswege vermieden werden. Nicht nur das Ergebnis ist wichtig, sondern auch der Weg.
4. Entlastung der Wissensnotation von rein informatikrelevanten Strukturen wie der Unterscheidung zwischen Programmierfehlern und Laufzeitfehlern oder der expliziten Definition von Datentypen. Das darzu-

stellende Modell kommt sehr gut ohne Bezug auf Programmierfehler und Datentypen aus. Die Implementierung soll im Gegensatz dazu inkrementelle Kompilation und verborgene Typenanalyse zur Fehlererkennung und Codeoptimierung verwenden. Die meisten Fehler können dann bereits bei der Erstellung der Wissensbasis erkannt werden.

5. Vermeidung überflüssiger Codierungen, wie z.B. die Numerisierung aller Stammdaten. Anstatt dessen Bevorzugung frei wählbarer symbolischer Datenrepräsentationen, d.h. Namen anstatt Nummern.
6. Bevorzugung problembezogener Strukturen, wie semibekannte Daten und dynamische Modelle, die eine korrekte Darstellung labormedizinischen Wissens erleichtern.
7. Unterstützung der Änderbarkeit und Wiederverwendbarkeit von Wissen durch Einführung von Schnittstellen und Begrenzung der Namensräume.
8. Anbindbarkeit externer Spezialsysteme wie Statistikprogramme (z.B. SPSS), Mathematikprogramme (z.B. Mathematica) oder von Programmbibliotheken (z.B. mathematische Fortranbibliothek) zur Anwendung hochentwickelter Rechenverfahren.
9. Zugreifbarkeit auf externe Datenbanken, z.B. Gen-datenbanken, Literaturdatenbanken oder Rote Liste über lokale oder globale Netzwerke, z.B. ISDN oder Internet.
10. Integrierbarkeit der Pro.M.D.-Systeme in Labor-EDV-Systeme oder Krankenhausinformationssysteme unter verschiedenen Betriebssystemen.

## Ergebnisse

Flexible Inferenzstrategien, funktionale Wissensdarstellung, semibekannte Daten, dynamische Modelle und vor allem die objektorientierten Strukturen des revidierten Pro.M.D.-Systems sollen den Einsatz wissensbasierter Systeme für die Laboratoriumsmedizin erheblich verbessern.

### Inferenzstrategien

Die Inferenzstrategie bestimmt, wie aus einem bestimmten Wissensmodul Schlußfolgerungen gezogen werden und insbesondere in welcher Reihenfolge die Schlußfolgerungsregeln des Moduls angewandt werden. Der bevorzugte deklarative Ansatz verlangt dabei, daß die Ergebnisse der Schlußfolgerungen von der Reihenfolge der Regelanordnung in der Wissensbasis unabhängig sind.

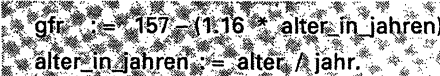

$$\begin{aligned} \text{gfr} &:= 157 - (1.16 * \text{alter in Jahren}) \\ \text{alter in Jahren} &:= \text{alter} / \text{Jahr} \end{aligned}$$

Abb. 1: Beispiel von zwei Schlußfolgerungsregeln zur Schätzung der glomerulären Filtrationsrate (gfr) nach Watkins und Shock

Konventionelle Programmiersprachen arbeiten diese Regeln als Anweisungen in der Reihe ihres Auftretens ab. So wird z.B. in der Regel  $y := x$ . der Wert des auf der

rechten Seite stehenden Ausdrucks  $x$  an die auf der linken Seite stehende Variable  $y$  zugewiesen. Im vorliegenden Beispiel könnte jedoch dann die glomeruläre Filtrationsrate nicht berechnet werden, da der Wert von *alter\_in\_jahren* erst danach berechnet wird. Zur Bewältigung dieses Reihenfolgenproblems kann in der Pro.M.D.-Schale zwischen verschiedenen Strategien gewählt werden:

Abarbeitung der Schlußfolgerungsregeln mit:

- ❶ **Einfache Vorwärtsverkettung:** Kann eine geeignete Sortierung der Regeln innerhalb einer Wissensbasis angenommen werden, so reicht ein linearer Durchlauf. Dieses Verfahren arbeitet zumeist am schnellsten. Falls die richtige Sortierung der Regeln in der Wissensbasis jedoch nicht sichergestellt ist, so wird dies als sogenannter Sequenzkonflikt gemeldet.
- ❷ **Vorwärtsverkettung mit Phasenwiederholung:** Hier wird eine Regel nur dann angewendet, wenn alle von ihr benötigten Daten bereits vorhanden sind. Gegebenenfalls werden mehrere Durchläufe über die Wissensbasis notwendig. Dieses Verfahren war ursprünglich in Pro.M.D. eingesetzt worden.
- ❸ **Vorwärtsverkettung nach Regelumsortierung:** Man kann jedoch auch die Schlußfolgerungsregeln durch einen Compiler vorsortieren lassen und dann das erste Verfahren einsetzen. Diese Technik steht in Pro.M.D. seit kurzem zur Verfügung.
- ❹ **Vorwärtsverkettung mit verzögerter Termabarbeitung (geplant):** Hier werden noch nicht anwendbare Regeln in einer Resteliste gespeichert und nach Anwendung anderer Regeln später erneut untersucht.
- ❺ **Rückwärtsverkettung:** Bei der Rückwärtsverkettung werden die Wissensbasen nicht linear abgearbeitet, sondern ausgehend von zu berechnenden Werten werden geeignete Regeln ausgesucht, die diese Werte berechnen können. Bei Anwendung dieser Regeln kann sich weitere Rückwärtsverkettung ergeben. Dieses Verfahren wird in Pro.M.D. bei den neuen objekt-orientierten Wissensbasen (s.u.) erstmals angewandt.

Diese verschiedenen Strategien führen alle zum gleichen Ergebnis, jedoch je nach Art der vorliegenden Wissensbasis mit unterschiedlicher Effizienz. Das Ziel der geeigneten Inferenzstrategie ist, den Steuerungsaufwand bei der Berechnung von Schlußfolgerungen für ein Wissensbasissegment zu minimieren.

### Funktionale Wissensdarstellung

Schlußfolgerungen aus Fachwissen verändern im Pro.M.D.-System das zugrundeliegende Wissen selbst nicht. Während der Schlußfolgerungsberechnung konvergieren die zu berechnenden Variablen vom Zustand des unbekannten Wertes hin zu dem exakten zu berechnenden Wert. Programmiersprachen, die ohne Seiteneffekte auf globalen Variablen arbeiten und Werte nur durch Funktionsanwendung auf Parameter berechnet werden, bezeichnet man als funktionale Programmiersprachen (3, 4, 5). Hierdurch entsteht eine gute Kontrolle über den Informationsfluß. Im Gegensatz zu imperativen Programmiersprachen, wie z.B. Basic, C, Fortran oder Pascal, kann bei Programmen in funktionalen Sprachen der ge-

samte Ableitungsbaum der Schlußfolgerungen statisch nachvollzogen werden.

Ziel der funktionalen Wissensdarstellung ist die bessere Verständlichkeit der Wissensbasen und in Zukunft eine „semantische“ Fehleranalyse zur Compilezeit.

### Kompetenzrechnung

Kompetenzrechnung betrifft mehr oder weniger vages Wissen. Vages Wissen ist Wissen mit ungenauen Daten. In Pro.M.D.-Terminologie werden ungenaue Daten als semibekannt bezeichnet (6). In Bezug auf die Aussagekraft von Daten unterscheiden wir bekannt, semibekannt und unbekannt (s. Tab. 1). Semibekannte Daten können weiter unterteilt werden in possibilistische Daten und probabilistische Daten.

Tab. 1: Stufen der Bekanntheit von Labordaten

● bekannter Wert	= eindeutige und präzise Information
● semibekannter Wert	= mehrdeutige oder unpräzise Information
● unbekannter Wert	= keine Information

In Abb. 2 enthält die erste Regel des Beispiels den Ausdruck *normal bis erhöht*, der keinen genauen Wert sondern einen semibekannten Wert liefert, der die möglichen (= possibilistischen) Alternativen Ausdruck *normal* und *erhöht* umfaßt. In der zweiten Regel wird der Variablen *crp* der unpräzise Wert  $6.2 \pm 0.15$  (im Mittelwert 6.2 und Standardabweichung 0.15) zugeordnet.

wenn glucose = (normal bis erhöht)  
dann keine\_hypoglykämie  
crp := (6.2 ± 0.15) mg/dl

Abb. 2: Beispiel von zwei Schlußfolgerungsregeln unter Verwendung von semibekannten Daten, Erläuterungen siehe Text

Aus solchen unpräzisen Daten können bei Kenntnis der Art der Streuung (normal, lognormal, ...) und ihrer statistischen Parameter dann Wahrscheinlichkeitsaussagen (= probabilistisch) abgeleitet werden. Werden probabilistische Daten aus Stichproben abgeleitet, so können in ihnen zusätzlich zu statistischen Parametern wie z.B. Mittelwert und Standardabweichung auch weitere Informationen wie Stichprobengröße enthalten sein. Zu ihrer Auswertung sind dann statistische Testverfahren erforderlich.

### Zeitbezogene Relationen

Die Verarbeitung von Wissen über zeitliche Beziehungen von Laborwerten erfordert eigene Instrumente. Die technischen Voraussetzungen sind, daß auf dieses Wissen (z.B. Daten aus Voruntersuchungen) zugegriffen werden kann und sowohl Zeitpunkt als auch Zeiträume in Wissensbasen dargestellt und bearbeitet werden können. Darüber hinaus wird im Rahmen des Pro.M.D.-Projekts die Darstellung von zeitbezogenen Relationen mit dynami-

schen Modellen erprobt. Dynamische Modelle werden durch ein System von Differentialgleichungen definiert (7, 8).

```

konz (FIIa) := +
+ spaltung (FII, FXaFVaKomplex, k27, k28)
- bindung (HepATIII, FIIa, TAT, k39, k40).

spaltung (A, E, Kcat, Km) :=
Kcat * konz (E) * konz (A) / (Km + konz (A)).

bindung (A, B, P, K1, K2) :=
K1 * konz (A) * konz (B) - K2 * konz (P).

k27 := 32 / sec.
k28 := 0.0058 uM.
k39 := 6.8 / uM sek.
k40 := 0 / sec.

```

Abb. 3: Beispiel eines dynamischen mathematischen Modells zur Thrombinaktivierung und -inaktivierung

Das Beispiel in Abb. 3 entstammt eigenen Vorarbeiten (9) und zeigt, wie die Bildung von Thrombin (FIIa) mit Hilfe eines dynamischen Modelles dargestellt werden kann. Dieses vereinfachte Beispiel besagt, daß Thrombin (FIIa) durch eine Spaltungsreaktion mit Michaelis-Menten-Kinetik ( $K_m = 5.8 \text{ nM}$  und  $K_{cat} = 32 \text{ sek}^{-1}$ ) entsteht und in einer Bindungsreaktion zweiter Ordnung ( $K_{on} = 6.8 \mu\text{M}^{-1} \cdot \text{sek}^{-1}$  ohne Rückreaktion) weiterreagiert. Das Symbol := weist in diesem Beispiel auf eine algebraische Gleichung hin, während das Symbol -= eine Differentialgleichung charakterisiert. Schlußfolgerungen können aus einem dynamischen Modell entweder mit Verfahren zur kontinuierlicher Modellsimulation oder mit Verfahren zur Parameteranpassung gezogen werden.

## Objektorientiertes Wissen.

Für Pro.M.D.-Wissensbasen wird ein Konzept der objektorientierten Strukturierung entwickelt, in welchem funktionale Objekte als Informationsträger dienen. Information setzt sich hierbei aus den das Objekt charakterisierenden Daten, den Objektparametern, sowie den zur Interpretation der Daten erforderlichen Methoden, dem Objektkontext, zusammen.

### Information = Daten + Kontext

Schlußfolgerungen werden gezogen, wenn nach Eingabe von Labordaten in eine Datenbanktabelle von der Pro.M.D.-Schale durch eine entsprechende Nachricht an ein Objekt der Wissensbasis, z.B. *labor(...)*, der dazugehörende Befundbericht, abgerufen wird.

Die objektorientierte Pro.M.D.-Sprache wird im folgenden als Pro.M.D.\*\* bezeichnet. Pro.M.D.\*\*-Wissensbasen sind aus Klassendefinitionen aufgebaut. Objekte sind durch einen Klassenbezeichner und aktuelle Parameter bestimmt. Spezielle Objekte sind Texte und Zahlen, wie z.B. „Institut für Klinische Chemie“ oder 3.14, die in der Wissensbasis direkt durch die entsprechende Zeichenkette oder Ziffernfolge angegeben werden. Das kleine Wissensbasissegment in Abb. 4 definiert eine Klasse *labor*, die zwei Methoden *befund\_brief* und *beurteilung*

und eine Unterklasse *fachwissen* enthält. Die Unterklasse *fachwissen* enthält unter anderem eine Methode zur Interpretation der Falldaten und zur Ermittlung einer Therapieempfehlung. Eine Klasse ist ein Abschnitt einer Wissensbasis. Jede Klasse definiert zwischen den Regeln *klasse ...* und *ende ...* einen lokalen Namensraum, in welchem die Bezeichner Methoden und Unterklassen erkannt werden. Enthaltene Unterklassen definieren eigene Namensräume. Der Namensraum einer Klasse kann durch Vererbung oder über die Klasse eines Nachrichtenparameters erweitert werden. So bewirkt z.B. die Regel *beerbe Aktueller\_Fall* in Abb. 4, daß zusätzlich zu den in der Klasse *labor* definierten Bezeichnern auch auf solche einer weiteren Klasse zugegriffen werden kann, die durch die Nachricht *Aktueller\_Fall* festgelegt ist. Der Zugriff auf die in einer Datenbanktabelle gespeicherten Falldaten erfolgt somit per Vererbung. Der Operator : bewirkt, daß der rechts vom Doppelpunkt stehende Bezeichner in dem links davon angegebenen Namensraum zu suchen ist. Der Operator # hängt die entsprechenden Textteile zu einem Gesamttext zusammen. Das Beispiel zeigt die Modularisierung der Wissensbasis, bei welcher die Klasse *labor* die Globalstruktur des Befundbriefes (Labor-kopf, Verwaltungskopf, Befunddarstellung und Beurteilung) festlegt, Detailinformationen jedoch von der separaten Unterklasse *fachwissen* bezieht.

```

klasse labor (AKTUELLER_FALL)
  beerbe AKTUELLER_FALL.

  methode befund_brief :=
  laborkopf #
  verwaltungskopf (patient, einsender) #
  befunddarstellung (AKTUELLER_FALL) #
  beurteilung (AKTUELLER_FALL).

  methode laborkopf :=
  „Institut für Klinische Chemie“.

  methode beurteilung (FALL) :=
  fachwissen (FALL) : interpretation #
  fachwissen (FALL) : therapieempfehlung.

  klasse fachwissen (FALL)
    methode interpretation := ...
    methode therapieempfehlung := ...
  ...
  ende fachwissen (FALL)
ende labor (AKTUELLER_FALL).

```

Abb. 4: Beispiel in objektorientierter Wissensnotation mit Klassen, Methoden und Vererbung, Erläuterungen im Text

Die interne Arbeitsweise der Pro.M.D.-Schale wird durch die Einführung der objektorientierten Wissensbasen wesentlich verändert. So wurde beispielsweise die bisherige Inferenzstrategie der Vorwärtsverkettung nun durch eine Rückwärtsverkettung abgelöst. Das neue Konzept ist jedoch mächtig genug, um bisher erstellte Wissensbasen mit nur relativ wenigen Änderungen auf die neue Sprache Pro.M.D.\*\* umstellen zu können. Diese Umstellung betrifft vor allem das Einfügen von Schnittstellen in den Wissensabschnitten und die Umwandlung von globalen Bezeichnern (Begriffen) in lokale Parameter, die mit einem Großbuchstaben beginnen. Die zu ändernden Schlüsselworte (abschnitt → klasse, betrachte → beerbe,

fakt → methode) können durch die Pro.M.D.-Schale selbst umgesetzt werden. Die verwendeten Pro.M.D.-Operatoren und die interne Logik der bisherigen Pro.M.D.-Wissensbasen brauchen für die Umstellung nach Pro.M.D.+ nicht geändert zu werden. Ziel der objektorientierten Darstellung ist es, Wissensbasen zu strukturieren und die Wiederverwendbarkeit von Teilen der Wissensbasis zu erleichtern.

## Diskussion

Die Alternative zum wissensbasierten Ansatz ist die direkte Programmierung von labormedizinischen Befundungssystemen in einer algorithmischen Programmiersprache, z.B. C++. Der Vorteil dieser Vorgehensweise liegt in der schnelleren Programmausführung bei der Berechnung von Schlußfolgerungen. Wie Aufwandsabschätzungen zeigen (10), sind diese algorithmischen Systeme jedoch schon bei Berücksichtigung weniger Parameter kaum noch pflegbar. Werden mehrere Labormethoden mit klinischen Angaben kombiniert, so ist deshalb meist nur noch ein wissensbasierter Ansatz sinnvoll.

Bei algorithmischen Programmiersprachen bestimmt die Anordnung der Anweisungen im Programm die Ausführungsreihenfolge der Befehle. Durch die Inferenzstrategie wird der Übergang von der algorithmischen zur deklarativen Wissensnotation ermöglicht, bei welcher die Schlußfolgerungsfunktion selbst die Reihenfolge der Regelarbeitung bestimmt.

Unüberschaubare Seiteneffekte können die Komplexität von imperativen Computerprogrammen so stark erhö-

hen, daß Fehler nur sehr schwer identifiziert und behoben werden können. Die im Pro.M.D.-System bevorzugte funktionale Wissensdarstellung vermeidet solche Seiteneffekte und erleichtert dadurch auch die nachträgliche Erklärung der gefolgerten Ergebnisse.

Heutige Computer sind dafür ausgelegt, mit präzisen und eindeutigen Daten zu arbeiten. In der Laboratoriumsmedizin ist diese Bedingung jedoch nicht erfüllt. Durch inhärente methodenbedingte Meßfehler ist kein Laborwert wirklich präzise bekannt. Mit der im Pro.M.D.-System nun eingeführten Technik der semibekannten Daten kann dieser Mangel an korrekter Information nicht ausgeglichen werden. Er kann jedoch soweit kontrolliert werden, daß die Überinterpretation von Labordaten und klinischen Informationen vermindert werden kann. Semibekannte Daten beleuchten auch statistische Methoden von einer anderen Seite. Im Gegensatz zur konventionellen Statistik ist bei semibekannten Daten die Art der Wahrscheinlichkeitsverteilung Bestandteil der Daten und nicht des Problems. Bei semibekannten Daten aus Stichproben genügt die Angabe der Fragestellung und der geeignete statistische Test kann dann durch die Schlußfolgerungsfunktion selbst ausgesucht werden.

Anwendungsgebiete für dynamische Modelle sind Kompartimentsysteme zur Beschreibung von Pharmakokinetik und -dynamik. Modellierung von hormonellen Regulationssystemen (z.B. Insulinregulation oder Schilddrüsenhormonregulation) und labordiagnostischen Testsystemen. Ziel des Einsatzes dynamischer Modelle ist vor allem die kompakte Darstellung physiologischen und biochemischen Wissens in Pro.M.D.-Wissensbasen. In labormedizinischen Wissensbasen können dynamische Modelle dazu dienen, Untersuchungsergebnisse verschie-

**BESITZEN SIE** ... eine Software, die die Einnahmen und Kosten Ihrer Verfahren errechnet und übersichtlich darstellt ?

Wir schon:

- Bestellwesen mit Lieferanten-Preisvergleich
- Kosten-Nutzen-Controlling einzelner Verfahren / Hitliste
- Artikel- und Lieferantenstatistiken
- Datentransfer vom Labor-Großrechner
- Finanzbuchhaltung oder Einnahme-Überschuß-Rechnung
- DELAB-Artikel- und Lieferantenkatalog

Fordern Sie den aktuellen  
Prospekt an. Wir beraten  
Sie gern !

**ACS** Software-GmbH  
Rheinstr. 24 41836 Hückelhoven  
Telefon: 02433/41037 Telefax 02433/41038

**MED-GIGANT**

dener Zeitpunkte zueinander in Beziehung zu setzen, z.B. zur Abschätzung des Zeitpunktes des Beginns eines wenige Stunden zurückliegenden Herzinfarktes aus der Verlaufskurve von Creatin-Kinase-Aktivitäten.

Von objektorientierten Verfahren versprechen sich die Informatiker eine gute Strukturierung vor allem umfangreicherer Programmsysteme. Da bei bestehenden objektorientierten Systemen, z.B. in den Programmiersprachen Smalltalk oder C++ die Klassenhierarchie nachträglich nur schwer verändert werden kann, haben wir für die Pro.M.D.-Wissensnotation einen flexibleren Vererbungsmechanismus gewählt, der neben multipler Vererbung über der Klassenhierarchie auch eine freiere Einbindung von Methodenbibliotheken erlaubt.

Ziel der vorgestellten Methoden zur Wissensdarstellung ist, daß medizinische Experten ohne Zwischenschaltung sogenannter Wissensingenieure ihr Fachwissen in einer formalen Sprache selbst darstellen können. Durch geeignete syntaktische Verpackung soll so auch der mathematisch und informatisch weniger Geübte Zugang zu den dargestellten Schlußfolgerungstechniken erhalten.

Anschrift für die Verfasser:

Dr. med. Bernhard Pohl  
Institut für Hygiene und Mikrobiologie der Universität Würzburg  
Josef-Schneider-Straße 2  
97080 Würzburg

## Literatur:

1. Pohl, B.; Trendelenburg, Chr. (1988): Pro.M.D. – A diagnostic expert system shell for clinical chemistry test result interpretation. *Meth. Inform. Med.* 27, 3; 111-117
2. Trendelenburg, Chr.; Pohl, B. (1990): Pro.M.D. – Medizinische Diagnostik mit Expertensystemen. Stuttgart: Thieme-Verlag, 3. Auflage
3. Plasmeijer, R.; van Eekelen, M. (1993): Functional programming and parallel graph rewriting. Bonn: Addison-Wesley
4. Bird, R.; Wadler, Ph. (1988): Introduction to functional programming. New York: Prentis Hall
5. Davie, A.J.T. (1992): An introduction to functional programming systems using haskell. Cambridge: Cambridge University Press
6. Pohl, B. (1993): Competence reasoning – handling ambiguous and imprecise data in the Pro.M.D. expert system shell. *Clinical Chimica Acta* 222; 85-100
7. Pearl, J. (1988): Probabilistic reasoning in intelligent systems: Networks of plausible inference. San Mateo: Morgan Kaufmann Publishers
8. Albert, A.; Harris, E.K. (1987): Multivariate interpretation of clinical laboratory data. New York: Marcel Dekker
9. Pohl, B.; Beringer, C.; Bomhard, M.; Keller, F. (1994): The quick machine – a mathematical model for the extrinsic activation of coagulation. *Haemostasis* 24: 325-337
10. Pohl, B.; Bepperling, C. (1994): Wissensbasierte Systeme in der Laboratoriumsmedizin. *mta zeitung* 9: 178-183
11. Carson, E.R.; Cobelli, C.; Finkelstein, L. (1983): The mathematical modeling of metabolic and endocrine systems. New York: Wiley
12. Dickinson, C.J. (1977): A computer model of human respiration. Lancaster, England: MTP Press Ltd.
13. Walther, S. (1992): Ein Konzept zur Verarbeitung eines Gewißheitsmaßes in einem regelbasierten Diagnosesystem. Diplomarbeit, Johann Wolfgang Goethe-Universität, Frankfurt am Main
14. Melzer, J. (1993): Analyse der Abhängigkeitsstrukturen von Variablen in Pro.M.D.-Wissensbasen zur Fehlererkennung und Codegenerierung. Diplomarbeit, Johann Wolfgang Goethe-Universität, Frankfurt am Main
15. Moos, M. (1994): Diplomarbeit, Johann Wolfgang Goethe-Universität, Frankfurt am Main
16. Burow F. (1994): Ein objektorientierter Ansatz für die Behandlung von Wissensbasen in der Expertensystemschale Pro.M.D.; Diplomarbeit, Johann Wolfgang Goethe-Universität, Frankfurt am Main
17. McCabe, F.G. (1992): Logic and objects. New York: Prentis Hall