



Intelligente tutorielle Systeme

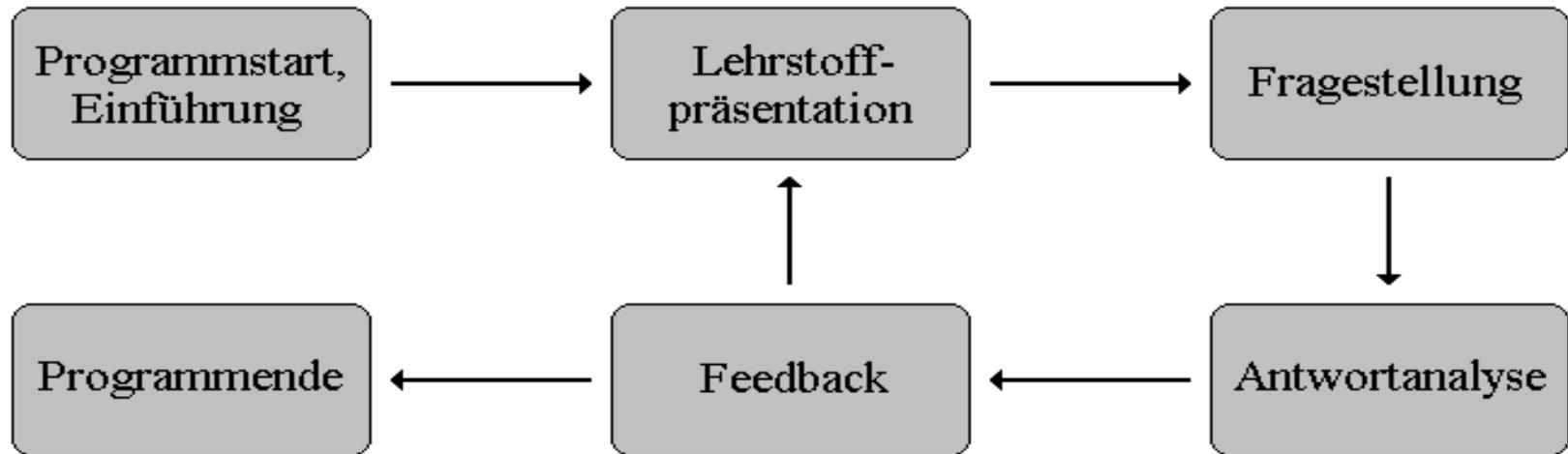
Hauptseminar Medieninformatik
Multimediale Lehr- und Lernsysteme

Carmen Eisendle
Wintersemester 2003/04

Überblick

- (Traditionelle) Tutorielle Systeme
- Definition eines ITS
- Hauptmerkmale
- weitere Eigenschaften
- Anwendungsbereiche von ITS
- Aufbau eines ITS
 - Das Wissensmodell
 - Das Lernermodell
 - Das Tutorenmodell
 - Die Benutzerschnittstelle
- Auflistung einiger ITS
- Beispiele
- Mängel und Kritik
- Ausblick

(Traditionelle) tutorielle Systeme



- Stoffvermittlung → Fragen → Verzweigung in andere Programmteile
- Kursablauf wird von der Software gesteuert und ist von den Antworten des Anwenders abhängig
- Verzweigung fest vorgegeben
- komplexe Zusammenhänge können nur schlecht dargestellt werden
- meist „naive“ Programme

Definition eines ITS

„Intelligente tutorielle Systeme (ITS) sind adaptive Mediensysteme, die sich ähnlich einem menschlichen Tutor an die kognitiven Prozesse des Lernenden anpassen sollen, indem sie die Lernfortschritte und –defizite analysieren und dementsprechend das Lernangebot generativ modifizieren sollen.“

Ludwig J. Issing, Paul Klimsa: Information und Lernen mit Multimedia (1995) S. 555

Hauptmerkmale

- **Adaptivität**

selbständige Anpassung an den jeweiligen Benutzer durch Auswertung einer Kombination von Informationen über Fachinhalte, pädagogischen Strategien und den Lernenden

- **Flexibilität**

Fähigkeit zur Änderung der Darstellung der Lerninhalte; durch getrennte Implementierung der Wissensbasis und der tutoriellen Komponente

- **Diagnosefähigkeit**

analysieren und Rückschlüsse über die Kompetenz des Lernalers führen

weitere Eigenschaften

- Ablauf **benutzergesteuert**, nicht von Kontrollstrukturen
- **wissensbasiert**
- verwendet Methoden der **Künstlichen Intelligenz (KI)**
- die Intelligenz liegt darin, einen **flexiblen und adaptiven Dialog** mit dem Lernenden führen zu können → **Individualisierung**
- zugrundeliegende Softwaretechnik meist regel- oder fallbasierte Expertensysteme, objektorientierte Wissensbasen oder Programmiersprachen (z.B. LISP)
- Versuche der Realisierung in den USA bereits seit etwa 1970

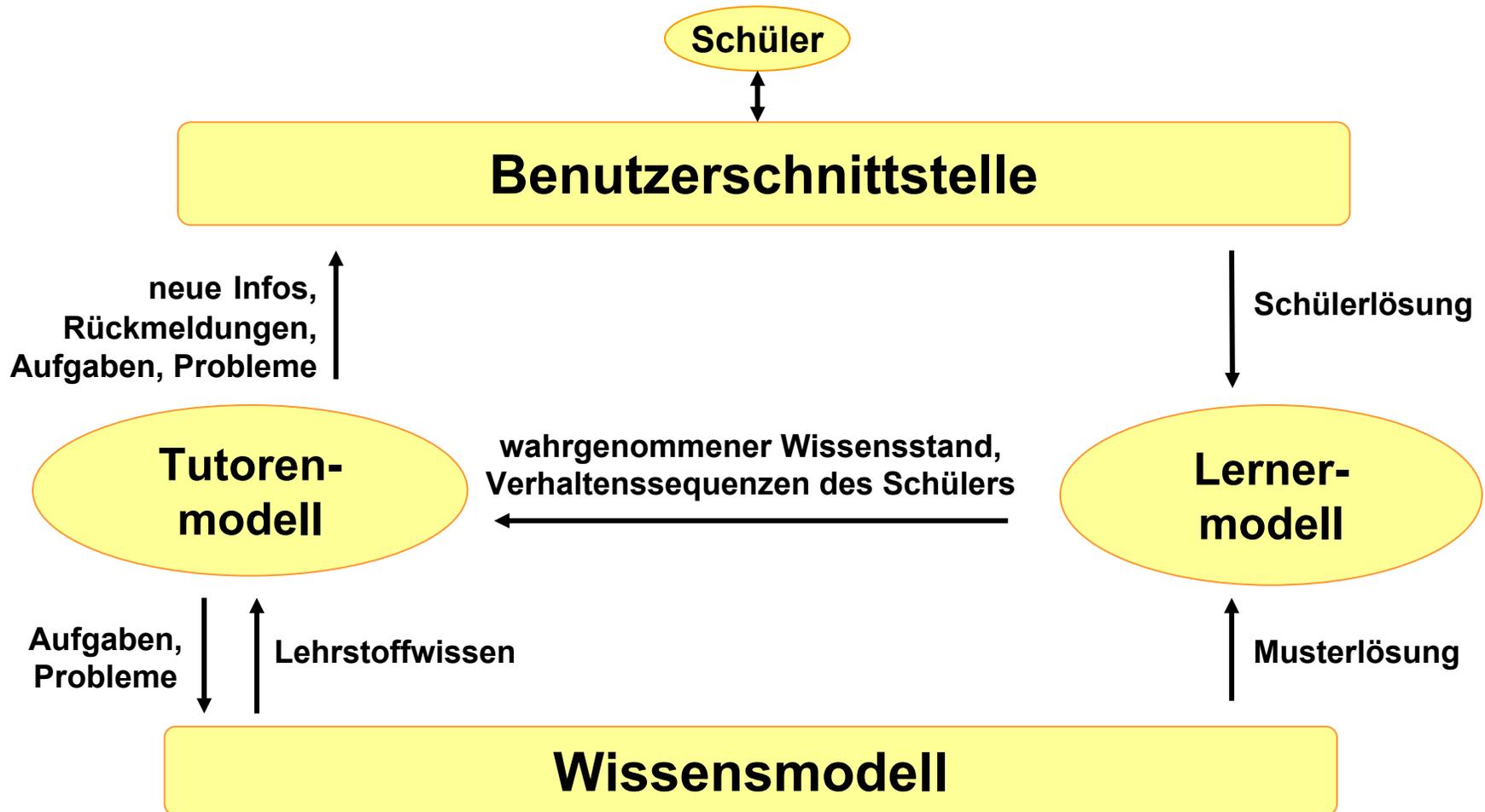
Anwendungsbereiche von ITS

Bereiche, in denen der Lerner Probleme lösen muss, die ein ITS prinzipiell selbst vorbildlich lösen kann, z.B.:

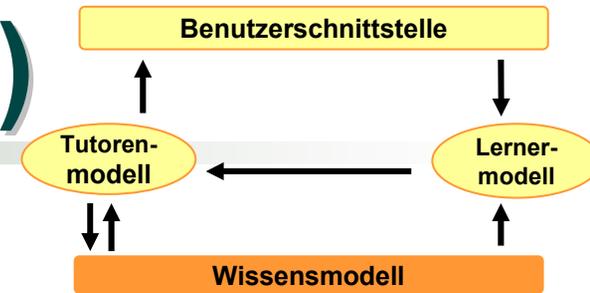
- Flugzeugsteuerung
- Anlagensteuerung
- Technische Reparaturdiagnostik
- Medizinische Entscheidungsfindung
- Juristische Entscheidungsfindung
- Mathematische Probleme (Arithmetik, Geometrie, Beweisen)
- Programmieren

(Frank Puppe, Uni Würzburg 2003)

Aufbau eines ITS

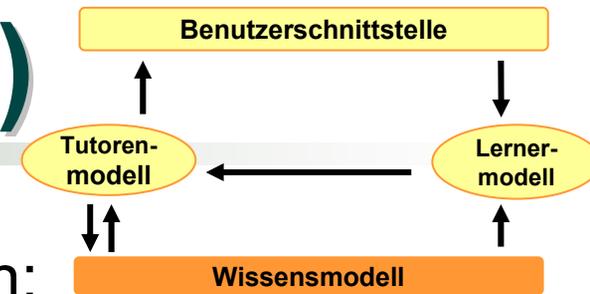


Das Wissensmodell (1)



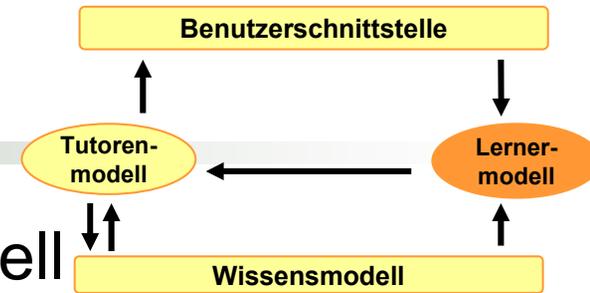
- auch *knowledge domain* oder Expertenmodell
- Ansammlung von Kenntnissen, Erfahrungen, Problemlösungsmethoden und Allgemeinwissen
- bildet die Wissensbasis (meist als Expertensystem):
 - **deklaratives Wissen** (Faktenwissen),
 - **prozedurales Wissen** (praktisches Wissen)
 - **heuristisches Wissen** (Erfahrungs- und Problemlösungswissen)
- dient als Bewertungsmaßstab

Das Wissensmodell (2)



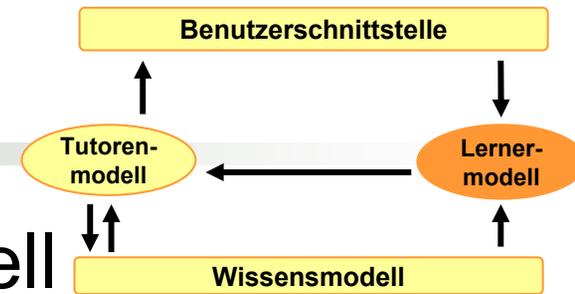
- verschiedene Modelle der Repräsentation:
 - „**black-box**“-Modell (nur die Ergebnisse der Problemlösungen können eingesehen werden)
 - „**glass-box**“-Modell (jeder Einzelschritt ist einsehbar)
- Organisation des Wissens in Form von Listen, Wissenstrukturdiagrammen oder Regeln
- meist in Form von semantischen Netzen modelliert: Knoten als Informationseinheiten und Kanten als Verweise zwischen ihnen
- Generierung von Musterlösungen durch Vorwärtsverkettung, Rückwärtsverkettung oder probabilistisches Ableiten

Das Lernermodell (1)



- auch *student model* oder Diagnosemodell
- enthält zu jedem Zeitpunkt aktuellen Wissensstand des Lernenden
 - Was weiß der Student? (Know that)
 - Was kann der Student? (Know how)
 - Was hat er schon gemacht? (Historie)
 - Was für ein Typ ist er? (individuelle Aussagen)
- beobachtet, interpretiert und diagnostiziert das Verhalten des Lernenden
- erkennt Lernfortschritte und -defizite und zieht daraus Schlussfolgerungen
- Wichtiger Bestandteil: Fehlerdarstellung
- große Probleme bei der Erkennung von Fehlerursachen in der Praxis

Das Lernermodell (2)



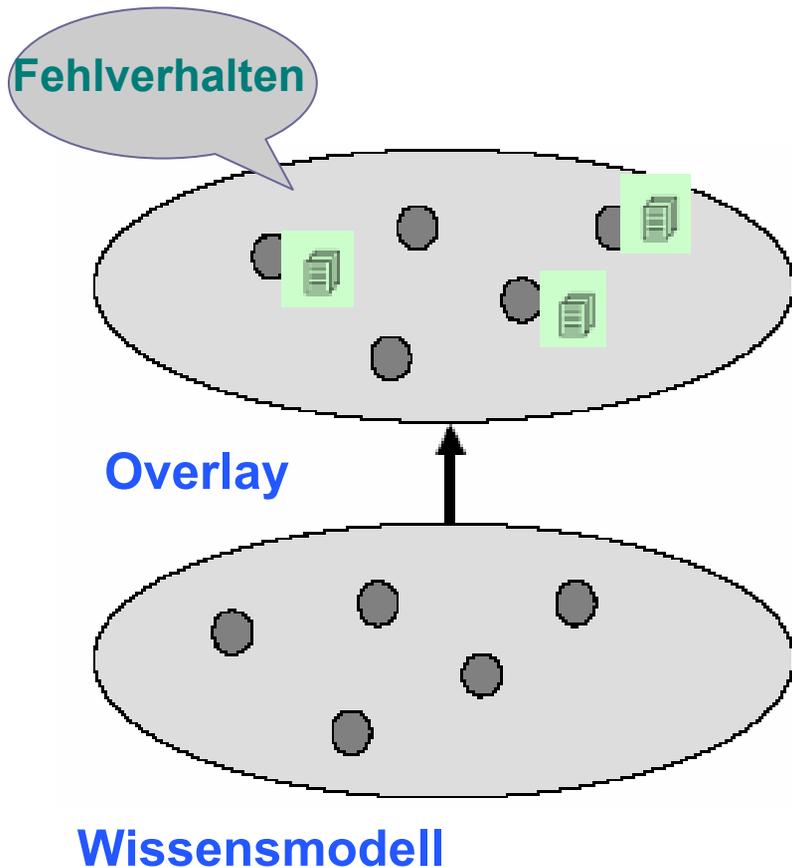
■ Anforderungen an das Lernermodell

- korrigieren
- vervollständigen
- anpassen
- diagnostizieren
- simulieren
- rekonstruieren

■ Arten von Lernermodellen:

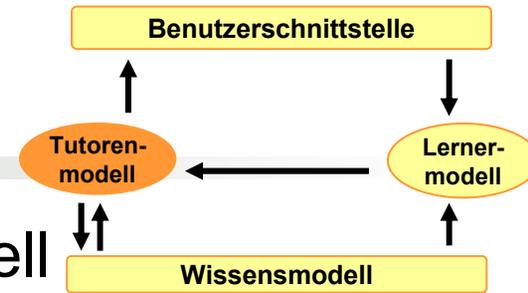
- Deviation-Modell (Abweichungsmodell)
 - aktueller Wissensstand = Performanz bzw. Abweichung vom Expertenwissen
- Subset oder Overlay-Modell (Untermengenmodell)
 - aktueller Wissensstand = Teilmenge des Expertenmodells

Overlay-Modell



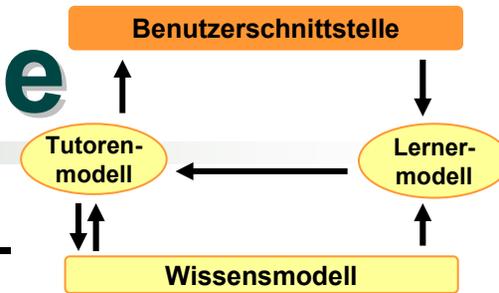
- zu jedem Wissensselement wird notiert, wie gut der Student es weiß und anwenden kann.
- Einfach und detailliert, aber unflexibel
- Annahme: Fehler stammen aus unvollständigem Wissen
- erweitertes Modell: berücksichtigt auch typisches Fehlverhalten

Das Tutorenmodell



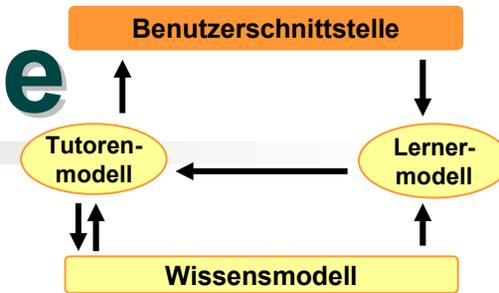
- auch *tutor model* oder pädagogisches Modell
- simuliert Entscheidungsverhalten eines Lehrers
→ *was soll wann wie* präsentiert werden
- enthält Wissensbasis für Lehrstrategien
- Aufgaben:
 - Auswahl der Lernziele
 - Überwachung und Bewertung des Problemlösungsverhaltens
 - Bestimmung des Zeitpunktes einer Unterbrechung und einer Hilfestellung
 - Präsentation einer Hilfestellung auf Anfrage
 - Auswahl von Hinweisen und Ratschlägen
 - Steuerung der Kommunikation
- erstellt psychologische Diagnosen
- vorherrschende Methoden: sokratischer Dialog und Coaching

Die Benutzerschnittstelle



- auch *interface* oder Kommunikationskomponente
- Übersetzung der internen Repräsentation
- eigentliche Form in der das System dem Anwender gegenübertritt
- Kommunikation soll der natürlichen Sprache möglichst nahe kommen → Problem der Realisation
- Anforderungen: robust, effizient, leicht erlern- und benutzbar
- soll die Adaptivität und Flexibilität des Systems ermöglichen

Die Benutzerschnittstelle



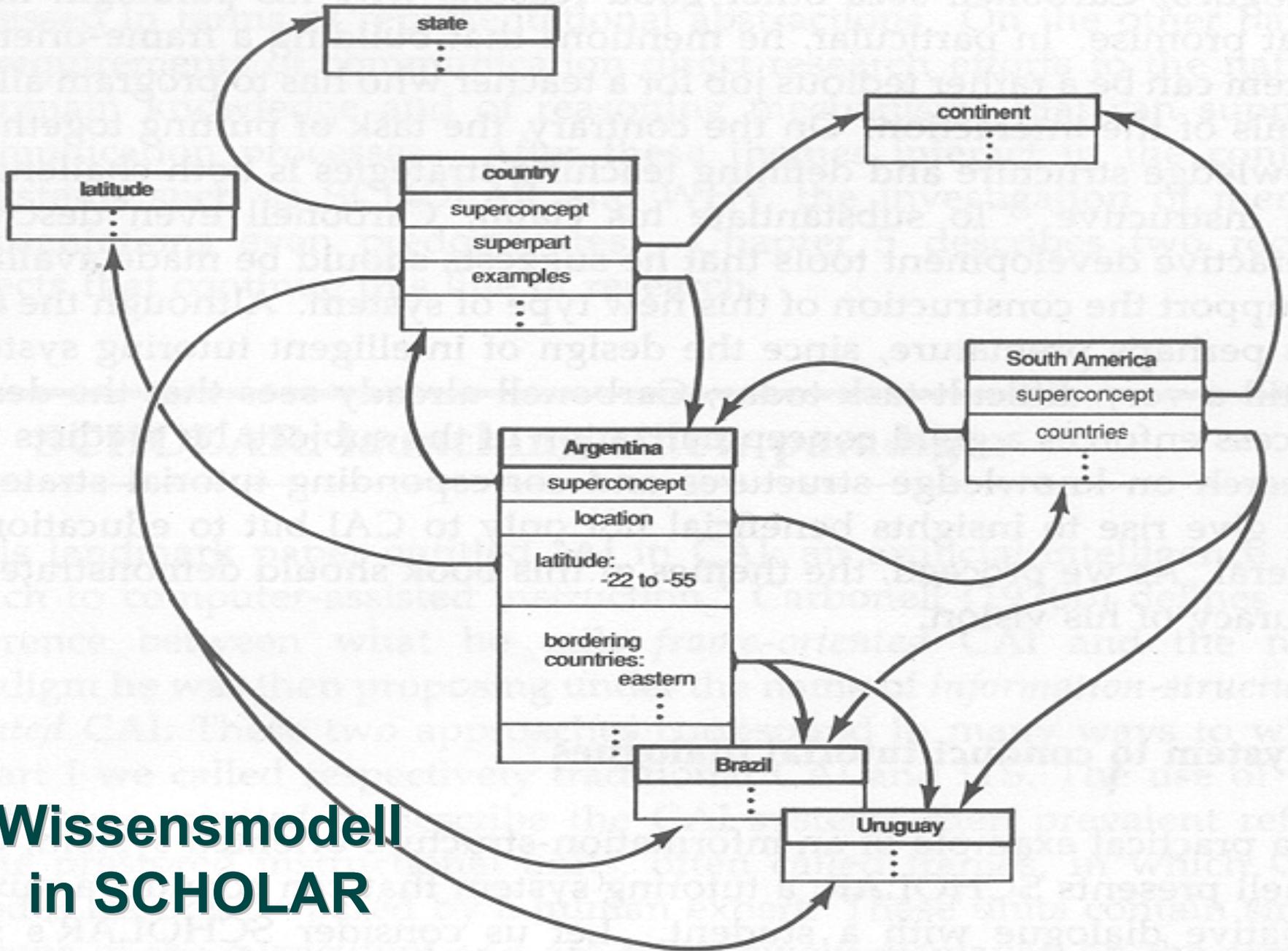
- Realisierung über
 - Sprache (Kommando-, Pseudo-natürliche oder natürliche Sprache)
 - Modellweltschnittstelle (Icons, Menüs, Grafische Darstellung)
- bietet Orientierungshilfen
 - Manual, Historie, Technische Bedienhilfen (z.B. Tooltips)
- Navigation möglichst flexibel (ereignisgesteuert vs. geführt)
- verschiedene Arten der Interaktion zwischen Lerner und System:
 - sokratischer Dialog
 - Coaching
 - learning by doing
 - learning while doing

Auflistung einiger ITS

Name	Literatur	Fachbereich
➤ SCHOLAR	Carbonell (1970)	Geographie
MYCIN	Schortliffe (1976)	Medizin
GUIDON	Clancey (1982/83)	
SOPHIE	Brown/Burton (1974), Burton (1982)	Physik (Schaltkreise)
PROUST	Johnson/Soloway (1987)	Informatik
WEST	Burton/Brown (1979)	Mathematik
➤ Algebraland	Brown (1985)	Mathematik
STEAMER	Williams/Hollan (1983) Hollan/Hutchins (1987)	Maschine
DiBi	Spada/Opwis (1988) Spada/Stumpf (1989)	Physik
➤ BRIDGE	Bonar/Cunningham (1988)	Informatik

Beispiel 1: SCHOLAR

- zum Lernen der Geographie Südamerikas
- verwendetes Lernermodell: Untermengenmodell
- Didaktik: (sokratischer) Dialog mit gemischter Initiative; Themenauswahl mit Interessanzheitszahlen, Dialoggeschichte und Zufallsgenerator
- Schwäche: Semantische Netze erlauben keine Darstellung von Prozessen, sondern nur von Faktenwissen
- Benutzeroberfläche: Pseudo-Natürliche Sprache
 - Textverstehen durch Muster- bzw. Schlüsselwortvergleich
 - Textgenerierung mit Satz- und Fragemuster; Auffüllen durch semantische Netze



Wissensmodell in SCHOLAR

Beispieldialog aus SCHOLAR

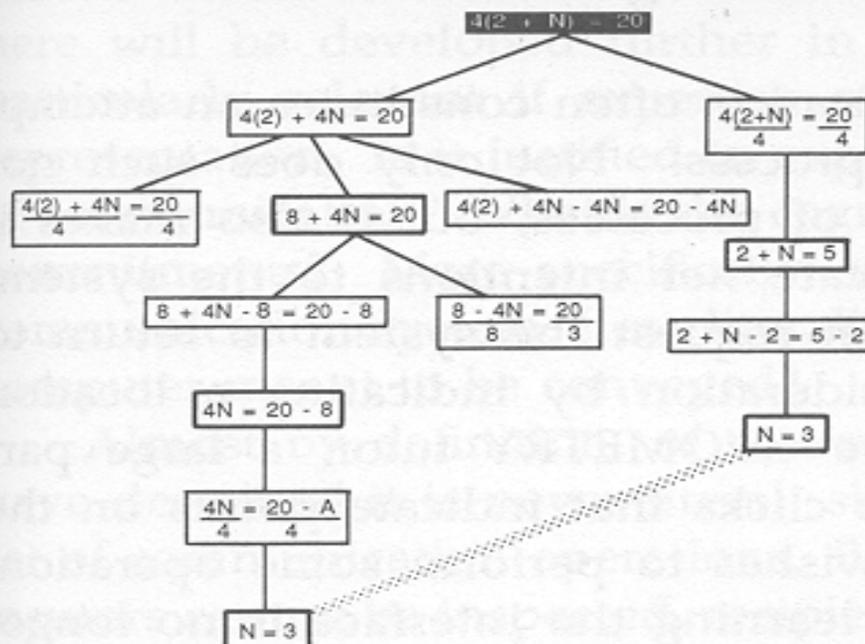
Scholar	Approx what is the area for Brazil?
Student	2 500 000 square miles
Scholar	Wrong. Please indicate if the following statement is correct: the area of Paraguay is approx 47 432 square miles.
Student	What is the area of Brazil?
Scholar	The area of Brazil is approx 3 600 000 square miles. Now answer the question you didn't answer before.
Student	False
Scholar	Rephrase your statement. I can't understand it.
Student	Incorrect
Scholar	Very good. The area of Paraguay is approx. 157 000 square miles. The language in Brazil is French. Right or Wrong?
Student	Wrong
Scholar	You are correct. The language of Brazil is Portuguese. Is it true that the capital in Brazil is Brazilia?
Student	Yes
Scholar	That's fine. Approx what is the population in Brazil?
Student	60 000 000
Scholar	Appox true, but a more exact value is 70000000. What is Brazil?
Student	Tell me about Peru. [...]

Beispiel 2: AlgebraLand

- zum Lösen von Gleichungen mit einer Unbekannten
- Wissensmodell: Regel- und Kontrollstrategien
- Didaktik: „Laut denken“. Student wählt erst abstrakte Schritte aus, bevor er sie umsetzt
 - Planebene (z.B. „Variablen isolieren“ oder „gleichartige Terme zusammenfassen“)
 - Operatorebene (z.B. auf beiden Seiten teilen, subtrahieren usw.)
 - Arithmetische Ebene (Umsetzung der Operatoren; kann auch an System delegiert werden)
- Benutzeroberfläche: Benutzer markiert Teilterm und Operation; zusätzlich verschiedene Übersichtsfenster

Prompt Window

SEARCH SPACE WINDOW



PLAN MENU

ISOLATE the variable.
COLLECT like terms into a single expression.
GROUP together like terms (transpose terms).
SPLIT apart expressions containing the variable.
SIMPLIFY the expression.

RECORD WINDOW

- (4) $4(2 + N) = 20$ (PROBLEM)
(9) $1(2 + N) = 20$ (DIVIDE)
(10) $1 + 3 = 4$ (Do-arithmetic)
(11) $R + N - 2 = 5 - 2$ (SUBTRACT)
(12) $4 =$ (Do-arithmetic)
(13) $N = 3$ (SOLVED)

PROBLEMS
(done)

Problem Mer

$$58 + 17 = 7 - 38$$

$$X(17 + 2) = 41$$

$$\frac{N}{8} = \frac{1}{4}$$

$$\frac{1}{U} + \frac{1}{V} = \frac{1}{F}$$

$$17(A-Z) = 10 (A+$$

$$A = P \left(1 + \frac{R}{100} \right)$$

Solve for N

$$4(2 + N) = 20$$

BASIC OPERATIONS

To Both Sides

ADD

SUBTRACT

MULTIPLY

DIVIDE

Do-Arithmetic

Distribute

Expand

Combine-Terms

Cancel-Terms

(undo)

(next problem)

Programmier-Tutorsysteme

Problem: Beim Programmieren gibt es sehr viele Freiheitsgrade. Hierzu gibt es verschiedene Ansätze

- Nur das Ergebnis testen (syntaktisch; semantisch)
 - Kein Hinweis auf Art des Fehlers
 - Keine Unterstützung bei Vorgehensweise
- Schrittweise Vorgehensweise (z.B bei SPADE, BRIDGE)
 - Meist Gängelung des Studenten
 - Fehler frühzeitig erkennbar
- Vergleich des Studentenprogramms mit Musterlösung (PROUST):
 - Viele Varianten der Musterlösung erforderlich (wegen Freiheitsgrade des Programmierens)
 - u.U. hilfreiche Kommentare

Beispiel 3: BRIDGE

- zum Lernen des Programmierens in Pascal
- Beobachtung: Viele Anfänger programmieren analog zu sprachlichen Äußerungen
- Idee: Programmierer zunächst in einer Art Pseudocode, die denn zu echtem Code verfeinert wird.
- Umsetzung: Vorgabe abstrakter sprachlicher Basiskonstrukte im Menü
 - nichtprozedural: *Durchschnitt berechnen, Menge von Zahlen*
 - Aggregierende Beschreibung: *Summe der Zahlen, Zähle sie*
 - Operationalisierung: *Addiere nächste Zahl zu der Summe bis 99999*
 - Pseudocode: *Wiederhole bis, lese eine Zahl ein*
- informellen Ideen → strukturierter Programmierplan
→ Programmstrukturen
- Lernender kann Hilfe anfordern, aber Coaching nur bei Problemen
- außerdem kann er Problemlösungsprozess verfolgen

Sprachlicher Plan

Ending Value Averaging Problem

Write a Program which repeatedly reads in integers until it reads the integer 99999. After seeing 99999, it should print out the CORRECT AVERAGE without counting the final 99999.

Select Here for Instructions

Natural Language Plans

Sum up . . . the integers
Compute . . . the average

English Phrase Menu

- Compute . . .
- Print . . .
- Output . . .
- . . . And So On . . .
- Continue steps . . .
- Keep doing steps . . .
- Get . . .
- Read in . . .
- Count . . .
- Add . . .
- Sum . . .

Sage Advice from Gwinky (sm)



What operator do you need to compute an average?

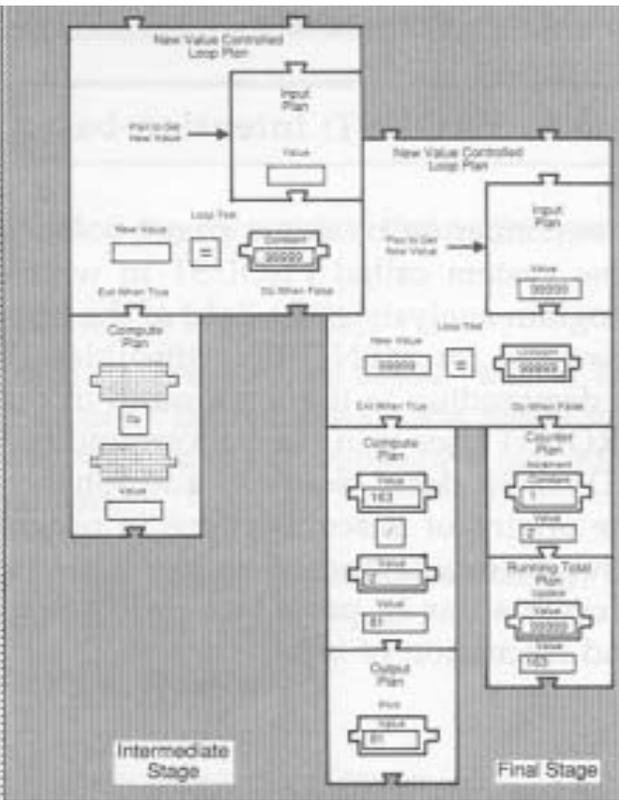
English step-by-step Solution

Continue steps . . .

- Read in . . . each integer
- Count . . . each integer
- Add . . . Integer to running total

Until 99999 is seen

- Compute . . . the average
- Print . . . the average



Sprachlicher Zwischen- & grafischer Endplan

Wie geht es weiter?

„Das Gebiet der intelligenten tutoriellen Systeme ist offenbar in der Phase der Programmierung steckengeblieben: Selbst wenn man nur die Aufsätze einer einzigen Zeitschrift zu dem Thema ITS betrachtet, so wiederholen sich ständig dieselben Aussagen, Systematisierungen und Rückgriffe auf die AI-Literatur. Ich habe selten so viel Redundanz auf dem Haufen gesehen, und selten eine so kleine Gemeinschaft von Forschern, die ständig dasselbe auf dem selben Entwicklungsstand veröffentlichen.“

Rolf Schulmeister, Grundlagen hypermedialer Lernsysteme (1996), S. 188

Mängel und Kritik (1)

- kaum Einsatz in der Praxis
 - nur einzelne Projekte und Prototypen (meist zu Forschungszwecken) sind bisher realisiert; können nicht für den realen Unterricht dienen
- kommerzieller Misserfolg
 - Komplexität → zu hoher Entwicklungsaufwand
- nur für begrenzte Themengebiete umsetzbar:
 - es eignen sich nur Themenbereiche, deren Wissen leicht in Regeln zu fassen und gut abzugrenzen ist (z.B. Mathematik, Informatik)
 - deshalb ist man auch von einer Entwicklung eines allgemeinen (domänenunspezifisches) Systems weit entfernt
- Primitivität des Lernermodells:
 - zu vereinfachende Auffassung vom Lernen, zu wenig Merkmale des Lerners werden erfasst → Fehldiagnosen
 - Bedarf an kognitiven Theorien des Lernens, die in der Lage sind, die gewünschten intelligenten Prozesse abzubilden

Mängel und Kritik (2)

- zu lineare und sequentielle Strategien, zu wenig Wahlmöglichkeiten
- Effizienz solcher Systeme gegenüber herkömmlichen interaktiven Lernprogrammen ist nicht gesichert größer
- Schwierigkeiten der Anpassung nicht nur in der Umsetzung am Computer sondern vor allem darin, den Menschen zu verstehen: jeder hat eigene Lernstile und andere Vorkenntnisse
- Wissenserwerb setzt aktive Rolle des Lernenden voraus → Gefahr der Reduktion der Eigenaktivitäten
- ITS-Systeme wissen, was der Schüler lernen soll und determinieren was er lernen kann → kein Raum für Entwicklung der Kreativität
- Einbindung des Lernens in einen sozialen Kontext wird nicht unterstützt
- Lehrstrategien: gibt es das überhaupt? Pädagogisches Geschick beruht großteils auf Einfühlungsvermögen

Ausblick

- teilweise Nutzung verschiedener Komponenten und Prinzipien der ITS:
 - in neueren Autorenwerkzeugen und Entwicklungsumgebungen (v.a. im Softwaretechnischen Bereich: Trennung der Teilkomponenten)
 - in Hypemediasystemen (als Wissens-Datenbanken, Mikrowelten und Diagnoseinstrumente)
- die Verbreitung graphischer Benutzeroberflächen zeigt: nicht nur *was*, sondern *wie* etwas dargestellt wird, ist entscheidend
 - dem Lernenden mehr Freiheiten in der Auswahl der Präsentation geben: wünscht er einfachere oder komplexere Darstellung?
 - Adaptivität dann Frage der Gestaltung, nicht mehr der Diagnosefähigkeit

Ausblick

- Adaptivität weiterhin erwünscht, jedoch ohne ein „Verständnis“ des Lernenden vorzutäuschen.
- Potentiale des Computereinsatzes auf anderem Niveau sehen:
 - Möglichkeit, reichhaltige Umgebungen zu schaffen, die Exploration und individuelle Konstruktion im Lernprozess betonen.
- Neuere Ansätze gehen in Richtung des Einsatzes intelligenter Agenten, die spezialisierte Aufgaben (z.B. Unterstützung bei der Suche in komplexen Datenbeständen) erledigen können.

**Vielen Dank für die
Aufmerksamkeit**

Quellen

- EULER, Dieter u.a.: Computerunterstützter Unterricht. Vieweg, Wiesbaden und Braunschweig 1987, S.73 ff
- GÖTZ, Klaus; HÄFNER, Peter: Computergestütztes Lernen in der Aus- und Weiterbildung. Dt. Studien-Verlag, Weinheim 1991
- ISSING, J. Ludwig; KLIMSA, Paul: Information und Lernen mit Multimedia. Psychologische Verlagsunion, Weinheim 1997
- KAMMERL, Rudolf: Computerunterstütztes Lernen. Oldenburg Verlag, München Wien 2000
- KERRES, Michael: Multimediale und telemediale Lernumgebungen. Konzeption und Entwicklung. Oldenburg Verlag, München Wien 2001, S.69 ff.
- KÜFFNER, Helmuth; SEIDEL, Christoph: Computerlernen und Autorensysteme. Verlag für Angewandte Psychologie, Stuttgart 1989, S. 127 und 145 ff.
- SCHULMEISTER, Rolf: Grundlagen hypermedialer Lernsysteme
- Vorlesungsfolien von Prof. Puppe (Uni Würzburg, SS 2003):
<http://ki.informatik.uni-wuerzburg.de/teach/ss-2003/its/uebungen/>