

ITS

intelligente tutorielle Systeme

von Kirstin Peters

zur Vorlesung: „Grundlagen hypermedialer Lernsysteme

13.12.2002

0. Einleitung:

1. Was ist ein „intelligentes tutorielles System“?:

2. Aufbau eines ITS:

2.1. Das Expertenmodul:

2.1.1. Das enthaltene Wissen:

2.1.2. Modelle für Expertenmodule:

2.1.2.1. Semantische Netze:

2.1.2.2. Frames:

2.1.2.3. Adaptive Control of Thought (ACT):

2.1.3. Generierung der Musterlösungen:

2.2. Das Lerner-Modul:

2.2.1. Darstellung des aktuellen Wissensstandes:

2.2.2. Fehlerdarstellung:

2.2.3. Diagnose des Lernverhaltens:

2.2.4. Kritikpunkte:

2.3. Das Pädagogische Modul:

2.4. Das Kommunikations-Modul:

2.5. Graphische Darstellung eines ITS:

3. Expertensysteme und ITS:

4. Was kann ein ITS leisten und was nicht?:

4.1. Adaptivität:

4.2. Themengebiete:

4.3. Lehrstrategien:

4.4. Anwendungsbereich:

4.5. Einordnung:

Quellen:

0. Einleitung:

„**Wissen ist Macht!**“ Wer musste die Richtigkeit dieses allseits bekannten Sprichwortes nicht schon am eigenen Leib erfahren? Gerade in der heutigen Zeit lernen wir niemals aus, sondern sind gezwungen ständig dazu zu lernen, wenn wir im täglichen Leben besonders im Berufsleben bestehen wollen. Doch wer hat schon die Zeit neben dem Job auch noch eine Schule, Universität oder Fortbildung zu besuchen? Dennoch ist Weiterbildung im heutigen Berufsleben unabdingbar, was die hohen Ausgaben der Firmen für die Weiterbildung ihrer Angestellten beweisen. Daher wurde der Ruf nach zeit- und geldsparenden Unterrichtsmethoden laut, wie den *intelligenten tutoriellen Systemen*, einem computergestützten Lernwerkzeug, das auf die in der Künstlichen Intelligenz gemachten Erfahrungen zurückgreift und ein schnelles und selbstbestimmtes Lernen ermöglichen soll.

1. Was ist ein „intelligentes tutorielles System“?:

Intelligente tutorielle Systeme, kurz **ITS** genannt, sind nicht der erste Versuch den Computer als Medium zum Unterrichten zu benutzen. Traditionellere Lernsoftware basierte oft auf dem Konzept des „Programmierten Unterrichts“. Hierbei wurde ein starrer Unterrichtsablauf entworfen und jegliche Reaktion des Programms auf den Lernenden musste im Vorfeld programmiert sein. Mit anderen Worten: Verhielt sich der Lernende in einer bei der Programmierung unvorhergesehenen Weise, so konnte das Programm nicht darauf reagieren.

Um dieses Problem zu beseitigen, erhält ein ITS bei der Programmierung neben dem zu vermittelnden Wissen auch noch gewisse Kenntnisse über das Unterrichten an sich. Das System soll mit Hilfe dieser pädagogischen Fähigkeiten in der Lage sein, sich flexibel an den Lernenden und dessen Wissensstand anzupassen und ihm so ein selbstbestimmtes - eben nicht nach einem starren Muster ablaufendes - individuelles Lernen ermöglichen.

Um das zu erreichen, muss das Programm in der Lage sein sich das aktuelle Wissen und Können des Lernenden vorzustellen, sich quasi ein „*Modell des Lernenden*“ zu erstellen.

2. Aufbau eines ITS:

Wie schon angedeutet besteht ein ITS meist aus mehreren Teilen, wie dem zu vermittelnden Wissen und einer modellhaften Vorstellung der aktuellen Kenntnisse des Lernenden. Genauer gesagt, unterscheidet man zwischen vier Teilen oder auch Modulen.

Der erste Teil wird „**Expertenmodul**“ genannt, denn er enthält den Lehrstoff und das dazugehörige Fachwissen. Er stellt also eine Wissensbasis dar, die dem eines Experten in diesem speziellen Gebiet entsprechen soll.

Der zweite Teil ist das schon erwähnte „**Lerner-Modul**“. Seine Aufgabe ist es, den aktuellen Wissensstand des Lernenden zu modellieren und außerdem eine Diagnose seines Verhaltens vorzunehmen.

Das Wissen über Lehrmethoden und -strategien ist im dritten Teil, dem „**Pädagogischen Modul**“ gespeichert.

Und die Aufgabe des letzten Moduls ist schließlich die Kommunikation zwischen dem Lernenden und dem System, deshalb auch der Name „**Kommunikations-Modul**“ oder Interface.

2.1. Das Expertenmodul:

2.1.1. Das enthaltene Wissen:

Die Wissensbasis („knowledge domain“) des Expertenmoduls beschreibt das Wissen eines Experten - daher auch der Name - über das zu lehrende Gebiet. Es umfasst sowohl **deklaratives**, als auch **prozedurales** und manchmal sogar **heuristisches Wissen**. Mit deklarativem Wissen meint man das reine Faktenwissen, das über die Objekte des Fachgebietes bekannt ist. Hier werden also die nötigen Begriffe definiert, wie etwa „Wurzel“ oder „Potenz“ zum Beispiel in einem Mathematik-Tutor. Prozedurales Wissen hingegen besteht aus Argumenten oder Regeln, mit deren Hilfe sich Beziehungen zwischen den Objekten darstellen und Probleme lösen lassen. Heuristisches Wissen schließlich sind im wesentlichen Handlungsempfehlungen der Experten, die dem Lernenden helfen können, die richtige Herangehensweise an ein Problem zu finden. „Löse zuerst alle Klammern auf“ könnte zum Beispiel ein solcher Rat für die richtige Herangehensweise an ein hier mathematisches Problem sein. Diese Empfehlungen

stammen oft aus dem Erfahrungswissen des Experten und werden benötigt, um den Lernenden in seinem Problemlöseverhalten anzuleiten.

Mit Hilfe von diesem gespeicherten Wissen muss das Expertenmodul in der Lage sein, Strategien zum Lösen der auftretenden Probleme zu generieren, quasi Musterlösungen zu liefern, damit die Lösungsvorschläge des Lernenden überprüft bzw. verglichen werden können. Auch die Zusammenstellung von Fragen, Antworten, Aufgaben und Erklärungen sollte aus dieser Wissensbasis möglich sein.

2.1.2. Modelle für Expertenmodule:

Nun gibt es unterschiedliche Modelle, mit denen man das Expertenmodul erzeugen kann. Zum einem wäre da die Repräsentation durch ein strenges „**black-box**“-Modell. Hierbei können nur die Ergebnisse der Problemlösungen eingesehen werden. Was das Programm macht, um auf diese Lösung zu kommen, bleibt im Dunkeln. Wogegen bei strengem Vorgehen nach dem sogenannten „**glass-box**“-Modell jeder einzelne Schritt einsehbar ist. Das Programm ist also durchsichtig.

Der Vorteil der letzteren Methode ist klar, fragt der Lernende nach Hilfe, so kann einfach der Lösungsweg des Expertenmoduls ausgegeben werden. Andererseits erhebt das „glass-box“-Modell an sich den Anspruch vergleichbare Methoden zum Problemlösen wie der Mensch anzuwenden, jedoch ist das nicht immer umsetzbar und auch nicht immer effektiv. In einem „black-box“-Modell können auch komplizierte Formeln zur Berechnung herangezogen werden, die für den Lernenden oft unlösbar aber zumindest verwirrend wären, für den Computer aber kein Problem darstellen.

So kann das Programm zum Beispiel zur Berechnung einer Potenz verkettete Funktionen nutzen, was für ihn viel schneller zu einem Ergebnis führt, während der Lernende eher versuchen wird dieses über Potenzgesetze zu tun. Außerdem braucht das Expertenmodul in diesem Fall auch nur einen Lösungsweg zu ermitteln und nicht alle möglichen, da es nicht mit dem Lösungsweg des Lernenden vergleichen muss.

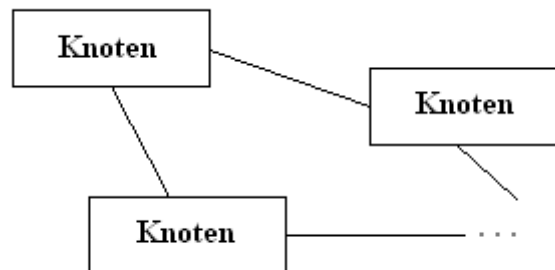
Oft ist aber gerade dieser Vergleich wichtig, so dass es erstrebenswert ist in einem Programm beide Modelle zu realisieren und sich genau zu überlegen, wann man den Lösungsweg vermitteln will, also eher das „glass-box“-Modell verwendet und wann nur eine schnelle Lösung wichtig ist, man also das „black-box“-Modell nimmt.

Allgemein wird das Wissen in Form von Listen, Wissensstrukturdiagrammen oder Regeln organisiert. Wichtig ist dabei nur, dass die Implementation des Wissens das Bilden von Musterlösungen erlaubt. Auch ist es von Vorteil, wenn Aussagen über eine möglichst gute Lernreihenfolge des Stoffes an für den Lernenden gemacht werden können.

Im folgenden werden einige mögliche Modelle zur Wissensrepräsentation kurz vorgestellt.

2.1.2.1. Semantische Netze:

Diese graphischen Notationen wurden Ende der sechziger Jahre von QUILLIAN eingeführt. „Ein semantisches Netz ist ein Graph, dessen Knoten Informationseinheiten und dessen Kanten Verweise zwischen ihnen sind.“ (Andreas Liening, „Intelligente tutorielle Systeme (ITS)“, Lit Verlag 1992, S. 130) Sie lassen sich leicht über Listen - zum Beispiel mit Hilfe von LISP - generieren.



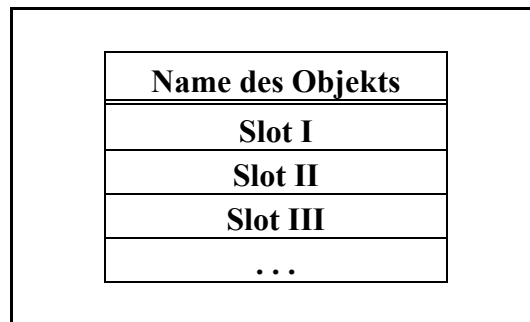
2.1.2.2. Frames:

MINSKY entwickelte diese semantischen Netze zu Frames weiter, um das Wissen zu begrifflichen Einheiten zusammenzufassen. Das Wissen wird in zusammenhängenden, aber differenzierbaren Einheiten strukturiert.

Innerhalb eines Frames wird ein Objekt mit seinen Eigenschaften, genannt Slots, dargestellt. Die Eigenschaften können mittels Facets näher beschrieben werden, hier kann zum Beispiel ein Wertebereich angegeben werden. Die Facets können aber auch Prozeduren zur Verwaltung des Objektes enthalten.

Als mögliche Sprache zur Realisierung von Frames kann die Wissensrepräsentationssprache CRL genannt werden.

Frame:



2.1.2.3. Adaptive Control of Thought (ACT):

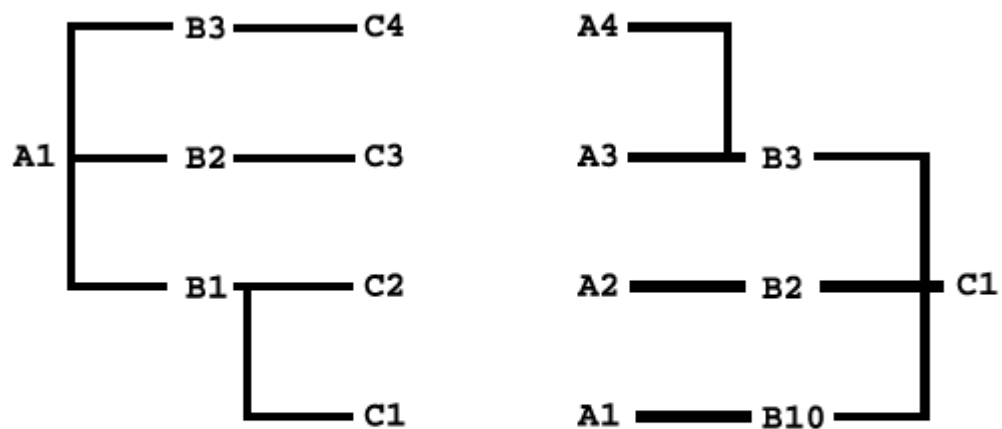
Diese Theorie wurde von ANDERSON, als Versuch höhere kognitive Funktionen des Menschen zu modellieren, entwickelt. Er unterteilt das Gedächtnis ins deklarative Gedächtnis (Wissensformen), prozedurale Gedächtnis (Regelwissen) und ins Arbeitsgedächtnis (aktuelle Informationen). ACT soll in gegebenen Situationen genauso wie ein Mensch reagieren, also menschliches Verhalten, auch Lernverhalten, simulieren. Allerdings gibt es aufgrund der Komplexität des menschlichen Denkapparates bisher noch keine vollständige praktische Umsetzung eines solchen Modells. Trotzdem könnte es für ein stark abgegrenztes Wissensgebiet eingesetzt werden.

2.1.3. Generierung der Musterlösungen:

Wie genau generiert das Programm nun eigentlich die Lösungswege aus seinem Fakten- und Regelwissen? Nun auch dafür gibt es wieder verschiedene Modellansätze.

Bei der ***Vorwärtsverkettung*** zum Beispiel gelangt man, ausgehend von einer Aussage, durch anwenden von Regeln zu einer neuen Aussage, auf die man dann wieder eine Regel anwenden kann, bis man schließlich zu einem Ziel kommt. Allerdings muss dieses Ziel nicht notwendigerweise dem entsprechen, das man erreichen will. In diesem Fall müsste man noch einmal von vorn beginnen und an mindestens einer Stelle eine andere Regel anwenden, falls das möglich ist, um zu einem neuen Ziel zu gelangen. Erst wenn man alle möglichen Wege gegangen ist und alle möglichen Ziele ermittelt hat, kann man im Vergleich sicher sein, dass das gegebene Ziel, wenn es mit keinem der Ermittelten übereinstimmt, mit der gegebenen Wissensbasis nicht zu erschließen ist.

Genau umgekehrt funktioniert die Methode der **Rückwärtsverkettung**. Hierbei werden ausgehend von einem Ziel - in diesem Fall auch Hypothese genannt - die Regeln rückwärtsverfolgt. Welche Methode die geeignetere ist, hängt stark von den einzelnen Aussagen mit ihren Beziehungen zu einander ab. Soll in der folgenden Abbildung links von A1 das Ziel C1 abgeleitet werden, so sollte dies über die Rückwärtsverkettung geschehen, wogegen rechts die Vorwärtsverkettung bevorzugt werden sollte. Im Allgemeinen eignet sich die Vorwärtsverkettung besser zum Lösen von Aufgaben, während mit der Rückwärtsverkettung die Lösungen leichter begründet (da zurückverfolgt) werden können.



"Grundlagen hypermedialer Lernsysteme", R. Schulmeister, 1997, S.164f.

Enthält die Wissensbasis vages Wissen, also Regeln, die nicht immer angewendet werden können, von denen man aber mit einer gewissen Wahrscheinlichkeit sagen kann, dass sie in der speziellen Situation zum richtigen Ergebnis führen, so kann man **probabilistisches Ableiten** einsetzen. In diesem Fall wird den Regeln eben jene Wahrscheinlichkeit, mit der sie in der Situation gelten, zugeordnet. Mittels statistischer Verfahren wird dann die „richtige“ Regel ausgesucht und angewandt.

2.2. Das Lerner-Modul:

Die Fähigkeit sich flexibel an den Lernenden anzupassen, wird in diesem Modul implementiert. Es enthält den aktuellen Wissensstand des Lernenden zu jedem Zeitpunkt des Programmablaufs. Dieser Wissensstand sollte auch Falschverstandenes und Gründe für die Schwierigkeiten mit einschließen.

2.2.1. Darstellung des aktuellen Wissensstandes:

Eine Möglichkeit den aktuellen Wissensstand des Lernenden aufzuzeichnen, ist sein Wissen als Teilmenge des Expertenwissens aufzufassen. In diesem „*Untermengenmodell*“ oder auch „subset model“ kann ein Lösungsweg allerdings nur dann als richtig erkannt werden, wenn er mit der Lösung des Experten übereinstimmt. Dieses Modell sollte also nur dann zum Einsatz kommen, wenn die Menge an möglichen Lösungswegen überschaubar gering ist.

Will oder kann man nicht im Vorfeld alle möglichen Lösungen programmieren, so sollte man wohl besser das „*Abweichungsmodell*“ oder auch „deviation model“ verwenden. Hierbei wird im wesentlichen nur überprüft, ob die Lösungen mit denen des Expertenwissens übereinstimmen. Das hat den Vorteil, das die Lösungswege völlig von dem des Expertenmoduls abweichen können. Andererseits ist es manchmal von Nöten den Lösungsweg zum Beispiel auf Effizienz zu überprüfen. Auch wenn es bei diesem Modell einfacher sein mag den aktuellen Wissensstand des Lernenden zu erfassen, ist es doch sehr schwierig den Lernenden zu beurteilen.

2.2.2. Fehlerdarstellung:

Wichtig neben dem aktuellen Wissensstand des Lernenden ist auch zu wissen, was der Lernende nicht kann und welche Fehler er macht, denn nur dann kann ihm spezifisch geholfen werden, diese Fehler zu überwinden. Hierzu ist es hilfreich eine Sprache zu definieren, die sowohl konkretes als auch fehlerhaftes Wissen über geeignete Regeln repräsentieren kann. Außerdem muss die Sprache in der Lage sein, neue Fehlerregeln definieren zu können, um auch jene Fehler zu erkennen, die nicht den vorhandenen Regeln entsprechen. Bei dieser sogenannten „*generativen Modellierung*“ entsteht

allerdings ein Konflikt zwischen der Notwendigkeit alle Fehler korrekt zu erkennen und den Sprachumfang so gering zu halten, dass sich die Laufzeit nicht zu sehr erhöht.

Eine andere Möglichkeit ist das sammeln von Fehlern in Fehlerbibliotheken. Um bei diesem „*replikativen Ansatz*“ die Bibliotheken nicht allzu groß werden zu lassen, kann man Fehlerregeln mit aufnehmen.

Das Problem bei beiden Ansätzen ist, dass es immer Aussagen geben wird, die weder unter dem bekannten Wissen noch in der Fehlerbibliothek zu finden sind, quasi Aussagen, die über das bekannte Wissen hinausgehen. Damit solche Antworten verarbeitet werden können, weißt man ihnen entweder ein Prädikat der Art „unbekannt“ zu oder versucht über *implizite Regeln* ihre Richtigkeit zu überprüfen. Mit impliziten Regeln meint man hierbei Regeln wie „wenn die Eigenschaft nicht explizit angegeben wird, existiert sie nicht“ oder „wenn diese Eigenschaft für die meisten ähnlichen Objekte gilt, dann gilt sie auch für dieses“.

2.2.3. Diagnose des Lernverhaltens:

Um wirklich individuell auf den Lernenden zu reagieren, ist es notwendig sein Verhalten zu interpretieren und zum Beispiel den Grund für seine Fehler zu ermitteln. Hierbei sollte man wohl anmerken, dass das Lerner-Modul selbst nur inhaltliche Diagnosen vornehmen kann. Psychologische Diagnosen sind Aufgabe des pädagogischen Moduls. Zur Beurteilung der Leistungen des Lernenden kann es von Vorteil sein zuerst einmal Vorhersagen über sein Verhalten zu erzeugen, da diese den Suchraum nach möglichen Interpretationen einschränken. Das Modul muss in der Lage sein, Ursachen für Fehler zu finden und die verwandten Lösungswege nachzuvollziehen, um dem nächsten Modul dem „Pädagogischen Modul“ die Arbeit zu ermöglichen.

Gerade diese Ursachensuche stellt in der Praxis ein großes Problem dar, gerade wenn es sich um psychologische Ursachen wie mangelnde Motivation oder auch Glück handelt. Da eine Maschine solche Ursachen nicht ermitteln kann, ist es zweckmäßig diese als statistische Größen mit in die Berechnung des aktuellen Wissenstandes aufzunehmen. So sollte ein Sachverhalt erst dann als verstanden gelten, wenn der Lernende ihn mehrfach in verschiedenen Situationen richtig angewandt hat und nicht schon nachdem er zufällig bei einer Aufgabe die richtige Lösung erraten hat.

2.2.4. Kritikpunkte:

So ungenau das Erkennen der Gründe für die gemachten Fehler ist, so ungenau müssen auch die Versuche des Programms bleiben, dem Lernenden über diese Fehler hinwegzuhelfen. So ist hier eine eindeutige Einschränkung momentaner maschineller Tutoren gegenüber ihren menschlichen Vorbildern gegeben.

Auch die Tatsache, dass die individuellen Lernstile gar nicht oder nur kaum in den Entwicklungsprozess einbezogen werden, weckt Kritik an den bestehenden tutoriellen Systemen.

Schließlich muss noch erwähnt werden, dass der Aufwand der betrieben wird, um die Module der Systeme zu erstellen, oft so hoch ist, dass sich kaum ein kommerzieller Nutzen erzielen lässt.

2.3. Das Pädagogische Modul:

Im pädagogischen Modul wird eine Wissensbasis für Lehrstrategien benötigt, mit deren Hilfe entschieden wird, ob und wie in die Lernsituation eingegriffen wird. Ob es gerade angebracht ist, Hilfe zu leisten, neue Aufgaben zu stellen oder etwa den Lernenden in Ruhe weiter arbeiten zu lassen. Auch die Motivierung des Lernenden ist Aufgabe dieses Moduls. Im wesentlichen soll also hier der Entscheidungsprozess eines Lehrers simuliert werden.

Eine dieser Lehrstrategien ist der *sokratische Dialog*, der im wesentlichen versucht, den Lernenden – hauptsächlich durch das Stellen von Fragen – dazu zu bringen, generelle Grundsätze aus individuellen Situationen heraus zu erkennen, also selbstständig zu denken. Mittels bestimmter Heuristiken wird versucht, den Lernenden zur Überprüfung seiner Hypothesen zu bringen. Er soll die Widersprüche in seinen Lösungsansätzen erkennen und daraus die richtigen Schlussfolgerungen ziehen.

Beim *Coaching* steht dagegen das *entdeckende Lernen* im Vordergrund. Das heißt das System greift nur selten aktiv - zum Beispiel durch Ratschläge, Musterlösungen, etc. - ins Lerngeschehen ein, sondern lässt den Lernenden selbst herausfinden, was er zu tun hat.

2.4. Das Kommunikations-Modul:

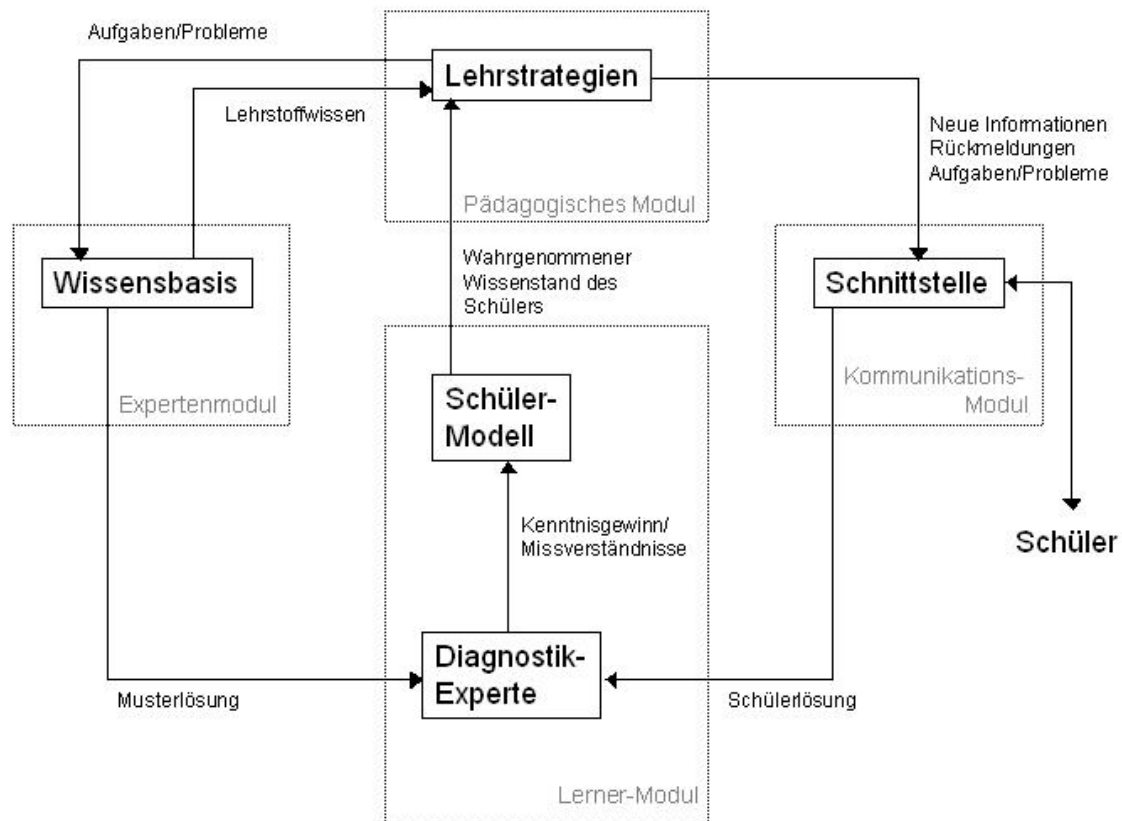
Das Kommunikations-Modul sorgt für die eigentliche Form in der das System gegenüber dem Anwender auftritt, außerdem ermöglicht es ihm, sich zu äußern und leitet die Eingaben gegebenenfalls nach einer Aufarbeitung an die jeweiligen Module weiter.

Eben jene Fähigkeit Eingaben des Lernenden anzunehmen und in einer für die einzelnen Module verständlichen Form weiterzuleiten, stellt in der praktischen Umsetzung ein großes Problem dar. Will man dem Lernenden erlauben in seiner „*natürlichen Sprache*“ zu antworten oder Fragen zu stellen, so muss das System in der Lage sein, diese zu verstehen. Viele haben sich schon mit der Realisation natürlichsprachlicher Dialoge zwischen Mensch und Maschine beschäftigt, allerdings gibt es bisher noch keine zufriedenstellende Lösung. Aber es ist möglich zumindest ein gewisses Verständnis der natürlichen Sprache zu ermöglichen, indem man auf die Interpretation über Schlüsselwörter zurückgreift, doch auch diese Methode ist in ihrem Sprachumfang und in ihrer Genauigkeit begrenzt.

Es stellt sich die Frage, ob es überhaupt erstrebenswert ist Natürlichsprachlichkeit umzusetzen, wenn man bedenkt, dass zum Beispiel *menügesteuerte Systeme* eine viel schnellere Eingabe ermöglichen, da nicht erst lange Sätze ausformuliert werden müssen. Andererseits sollte das System dem Lernenden ermöglichen unerwartete Fragen zu stellen, für die eine Menüsteuerung nicht in Frage kommt. Daher bietet sich auch hier wohl wieder eine Kombination verschiedener Ansätze an.

Darüber hinaus sollte ein gutes ITS auch mit Grafiken und Animationen arbeiten, da mit diesen zum Beispiel statistische Daten viel besser veranschaulicht werden können und sich das Lernen auch interessanter gestalten lässt.

2.5. Graphische Darstellung eines ITS:



Aufbau eines ITS (frei nach LEUTNER, 1992, S.61)

3. Expertensysteme und ITS:

Ein sogenanntes *Expertensystem* ist ein intelligentes Programm, das aus einer Wissensbasis und einer Inferenzmaschine besteht. Es kann also mit Hilfe von Schlussfolgerungsregeln und –prozeduren Probleme lösen. Mit einem Expertensystem wird versucht, das Wissen von Experten zu sammeln. Über Vorwärts- und Rückwärtsverkettung werden dann die jeweiligen Schlussfolgerungen gezogen. Im Gegensatz zu ITS versuchen Expertensysteme aber nicht menschliches Denken nachzuahmen. Außerdem ist es auch nicht in der Lage zu lernen, da es seine Wissensbasis allein durch Befragen der Experten bildet.

Darin liegt auch ihr Schwachpunkt, denn oft ist es für Experten schwer zu erklären wie genau sie etwas tun. Man kann zum Beispiel nicht in einfachen Verhaltensregeln ausdrücken, wie man Fahrrad fährt. Man lernt es einfach im Laufe der Zeit und kann es dann, doch in Formeln zu fassen was genau man dabei tut, ist sehr schwierig und kann von einem Computer nicht nachvollzogen werden.

Expertensysteme können einem ITS als Wissensbasis im Expertenmodul dienen. Sie können aber auch das Wissen über didaktische Strukturen, also die Lehrstrategien, abbilden.

4. Was kann ein ITS leisten und was nicht?:

4.1. Adaptivität:

Wie eingangs erläutert, soll die individuelle Anpassung an den Lernenden den großen Vorteil von IT-Systemen gegenüber anderen computergestützten Tutoren bilden. Doch in wie weit wird diese Adaptivität tatsächlich erreicht?

Die Schwierigkeiten der Anpassung liegen hier nicht nur in der Umsetzung am Computer, sondern vor allem auch erst mal darin den Menschen zu verstehen. Praktisch jeder Mensch hat einen eigenen Lernstil und andere Vorkenntnisse. Ein ITS kann lediglich versuchen, grobe Lernstile zu implementieren und dem Lernenden diese, vielleicht auch in einer Mischform, zu präsentieren.

4.2. Themengebiete:

Wenn man sich die bisher erschienen IT-Systeme so anschaut, fällt auf, dass die meisten von ihnen Programmiersprachen lehren oder sich mit physikalischen oder mathematischen Themen beschäftigen. Es werden also Themen gewählt, deren Wissen leicht in Regeln zu fassen und das gut abzugrenzen ist.

Offensichtlich ist es viel schwieriger eine Wissensbasis zu generieren, die sich mit einem schwer in Regeln und Fakten zu fassenden Themenkomplex, wie etwa der Psychologie, auseinandersetzt. Zur Zeit scheint sich der Einsatz von ITS also nur in einer begrenzten Auswahl von Themengebieten zu lohnen.

4.3. Lehrstrategien:

Auch in Bezug auf die im pädagogischen Modul enthaltenen Lehrstrategien gibt es Kritik. Ist es überhaupt möglich die Fähigkeit, jemandem etwas beizubringen, in eine Strategie umzuschreiben? Pädagogisches Geschick ist etwas das zum großen Teil doch auf Einfühlungsvermögen beruht, etwas das einer Maschine ganz bestimmt nicht innewohnt. Auch scheinen die bestehenden Strategien kaum mehr als grobe Richtlinien zu sein und nicht etwa genaue Vorgehensregeln, so kann zum Beispiel nicht genau sagen, wie viel Selbstbestimmung gut für den Lernenden ist und wie viel Anleitung er braucht.

Andererseits hat nicht auch ein menschlicher Lehrer Fehler? Er hat Vorurteile gegenüber dem Lernenden, von denen er nur schwer abweichen kann und die ein ITS sicher nicht hat.

Es stellt sich also die Frage, ob der Computer überhaupt ein genauso guter Lehrer wie ein Mensch sein kann.

4.4. Anwendungsbereich:

Die Möglichkeit den Computer als Lehrer zu benutzen, mag so manche Fantasie zu mutigen Spekulationen reizen. Wird die Schule so wie wir sie kennen bald nicht mehr existieren? Werden menschliche Lehrer aus unserem Alltag verschwinden und komplett durch Maschinen ersetzt werden? Können Computer genauso viel wie ein menschlicher Lehrer leisten?

Zumindest die letzte Frage können wir wohl sofort mit einem klaren „Nein“ beantworten. Die Aufgaben eines Lehrers umfassen schließlich weit mehr als bloßes vermitteln von Fachwissen. Neben dem Elternhaus ist die Schule eine der wichtigsten Sozialisationsinstanzen. Die erzieherische Komponente des Lehrberufs wird wahrscheinlich niemals von einer Maschine ausführbar sein. Wie soll ein Computer Werte vermitteln, wenn er selbst nicht weiß wofür sie stehen?

Modernere Ansätze vertreten eher die Meinung, dem Computer eine rein unterstützende Funktion zuzuordnen. Wird in der Schule ein Teil des Stoffes an einer Maschine unterrichtet, so gibt das dem Lehrer Zeit, sich um spezielle Gruppen seiner Schüler - so zum Beispiel Leistungsschwache - zu kümmern.

Eingesetzt werden Computer zur Zeit hauptsächlich als Sprachen- und Rechtschreibprogramme, zur Vermittlung von Basiswissen in Mathematik und Naturwissenschaften, als Vokabel- und Rechentrainer, als Tutorial für Anwendungssoftware oder als Simulatoren zum Beispiel in der Flugschule.

Beispiele für IT-Systeme:

Name	Literatur	Fachbereich
SCHOLAR	Carbonell (1970)	Geographie
MYCIN	Schortliffe (1976)	Identifikation von Infektionen
GUIDON	Clancey (1982) Clancey (1983)	in der Medizin
LMS/PIXIE	Sleeman (1987)	Manipulation von algebraischen Ausdrücken mit Fehlererkennung
SOPHIE	Brown/Burton (1974) Burton (1982)	Fehlererkennung in elektrischen Schaltkreisen, Physik
PROUST	Johnson/Soloway (1987)	Fehlererkennung bei der Programmierung von Pascal
Geometry Tutor	Anderson/Boyle (1985)	Lernprogramm für einfache Geometrie
WEST	Burton/Brown (1979)	Mathematik
BUGGY	Burton/Brown (1978) Brown/Burton (1982)	Mathematik
Algebraland	Brown (1985)	Mathematik
Quadratic Tutor	O'Shea (1982)	Mathematik
STEAMER	Williams/Hollan (1983) Hollan/Hutchins (1987)	Maschine
OHM	Bonar/Logan (1986)	Physik
DiBi	Spada/Opwis (1988) Spada/Stumpf (1989)	Physik
SUOMO	Mandl/Bollwahn (1990)	Physik
LISP Tutor	Anderson/Reiser (1985)	Informatik
BRIDGE Bridge Tutor	Bonar/Cunningham (1988)	Informatik
Smithtown	Shute/Glaser (1989) Shute/Glaser (1990)	Ökonomie
Physics Tutor	Jonassen (1992)	Hypertext-System mit integriertem Expertensystem zur Physik
GEO Tutor	Duchastel (1989) Duchastel (1992)	Geographie
SEDAF	Micarelli/Balestra (1992)	Berechnung von Polynomen

SAMPLE		Berechnung elektrischer Schaltkreise
BIOMECH	Giardina (1992)	Biologie/Medizin
KAVIS	Fischer/Mandl (1988)	Biologie
ILIAD	Applied Informatics Inc., Salt Lake City, UT	Medizin

"Grundlagen hypermedialer Lernsysteme", R. Schulmeister, 1997, S.192 f.

4.5. Einordnung:

Abschließend möchte ich noch kurz anhand folgender Tabelle einen Vergleich verschiedener Lernparadigmen führen und die Einordnung von IST in diese zeigen.

Tabellarischer Vergleich:

Kategorie	Behaviourismus	Kognitivismus	Konstruktivismus
Hirn ist ein	Passiver Behälter	Informationsverarbeitendes "Gerät"	Informationell geschlossenes System
Wissen wird	Abgelagert	Verarbeitet	Konstruiert
Wissen ist	Eine korrekte Input-Output-Relation	Ein adäquater interner Verarbeitungsprozess	Mit einer Situation operieren zu können
Lernziele	Richtige Antworten	Richtige Methoden zur Antwortfindung	Komplexe Situationen bewältigen
Paradigma	Stimulus-Response	Problemlösung	Konstruktion
Strategie	Lehren	Beobachten und helfen	Kooperieren
Lehrer ist	Autorität	Tutor	Coach, (Spieler)Trainer
Feedback	Extern vorgegeben	Extern modelliert	Intern modelliert
Interaktion	Starr vorgegeben	Dynamisch in Abhängigkeit des externen Lernmodells	Selbstreferentiell, zirkulär, strukturdeterminiert (autonom)
Programmmerkmale	Starrer Ablauf, quantitative Zeit- und Antwortstatistik	Dynamisch gesteuerter Ablauf, vorgegebene Problemstellung, Antwortanalyse	Dynamisch, komplex vernetzte Systeme, keine vorgegebene Problemstellung
Software-Paradigma	Lernmaschine	Künstliche Intelligenz	Sozio-technische Umgebungen
"idealer" Softwaretypus	Tutorielle Systeme, Drill & Practice	Adaptive Systeme, ITS	Simulationen, Mikrowelten, Hypermedia

Lernparadigmen und Softwaretypologie (nach [Baumgarten/Payr 94, 110, 174])

Quellen:

- (1) „Grundlagen hypermedialer Lernsysteme“, Rolf Schulmeister, 1997
- (2) „Intelligente Tutorielle Systeme (ITS)“, Andreas Liening, Lit Verlag 1992
- (3) „Computerunterstütztes Lernen in der Aus- und Weiterbildung“, Klaus Götz/Peter Häfner, Deutscher Studien Verlag Weinheim, 3. überarbeitete Auflage 1992