Global J. of Engng. Educ., Vol.6, No.3
Published in Australia

Was können Tests in E-Learning Anwendungen für die Mathematik-Ausbildung Leisten?

Thomas Risse
Hochschule Bremen, Institut für Informatik & Automation
Neustadtswall 30, D-28199 Bremen, Deutschland

Unter Berücksichtigung vorhandener Standards wird untersucht, was mit Tests in e-learning Anwendungen eigentlich bewertet werden kann. Beispiele aus einer kleinen Feld-Studie in der Mathematik illustrieren die Interaktion zwischen Lehrern und Studierenden, zeigen, inwieweit Frage-Antwort-Paare zur Lernerfolgskontrolle auf die standardisierten Tests abgebildet werden können und in welche Richtung diese Standards erweitert werden sollten.

EINLEITUNG

Tests spielen in e-learning Anwendungen eine gewichtige Rolle, einerseits für die Navigation des Lerners auf individuellen Lernpfaden, andererseits für die (Selbst-) Kontrolle des Lernerfolges. Es liegt in der Natur der Sache, daß solche Tests leicht bewerten können, ob der Lerner die richtige Antwort *angekreuzt* hat. Es ist dagegen schwierig, Rechner-gestützt Ansatz, Herangehensweise, erzeugtes Modell usw. zu bewerten.

In dieser Untersuchung wird nun eine traditionelle Präsenz-Lehrveranstaltung analysiert:

- Um die Interaktion zwischen Lehrer und Studierenden im Hinblick auf Unterstützung des Lernprozesses und auf Überprüfen des Lernerfolges zu klassifizieren.
- Um die indentifizierten Typen von Frage-Antwort-Paaren auf – soweit vorhanden – entsprechende Tests, im IMS slang sogenannte *Basic Item Types* [1] abzubilden.

Auf diese Weise wird das derzeitige Potential dieser Art von Computer-Unterstützung deutlich. Deren Defizite sollten durch wünschenswerte Erweiterungen ausgeglichen werden. Darüber hinaus kann die Analyse Hinweise für den Entwurf intelligenter Tutoring Systems, IST liefern [2].

Dieses Ziel vor Augen, müssen zwei Welten verbunden werden: auf der eine Seite die Lehr- und Lernwelt mit der ganz eigenen Gesetzen unterliegenden Interaktion zwischen Lehrern und Studierenden, und auf der anderen Seite die technische Welt der standardisierten Tests, geprägt davon, was mit Rechnern (leicht) machbar ist.

Standardisierung

Seit Jahren investieren nationale und internationale Gremien beträchtliche Ressourcen, um Standards für e-learning Anwendungen zu definieren (sogar für das zu standardisierende Objekt (LOM?) gibt es keine einheitliche Bezeichnung!). Sowohl die Industrie, wie z.B. *Aviation Industry CBT Committee* (AICC) (www.aic.org) wie auch Fachverbände, wie z.B. IEEE veröffentlichen Vorschläge beispielsweise für die Beschreibung von Anwendungen durch Meta-Daten, für die Architektur von Lern-Systemen, für Qualitäts-standards usw., um Lern-Material in verschiedenen Kontexten nutzen zu können. Einige Initiativen mögen als Beispiele dienen:

- *Advanced Distributed Learning* (ADL) ist eine Initiative, angestoßen durch das DoD, die e-learning standardisiert, z.B. durch *Sharable Content Object Reference Model*, (SCORM), s. http://xml.coverpages.org/scorm.html unter der Verwendung von XML, www.adlnet.org
- *Alliance of Remote Instructional Authoring and Distribution Networks for Europe* (ARIADNE) ist ein Projekt der Europäischen Union. Ein Konsortium stellt Meta-Daten, tools, Benutzer-Gruppen, etc, zur Verfügung, www.ariadne-eu.org
- Dublin Core Metadata Initiative entwickelt Standards für online Meta.Ddaten, die einem

breiten Spektrum von Zwecken und business Modellen zu Gute kommen sollen, http://dublincore.org

- *IMS Global Learning Consortium, Inc.* wurde als ein Projekt in der National Learning Infrastructure Initiative von EDUCAUSE gegründet. Benutzer, software Entwickler, Aktivisten stellen XML Spezifikationen für Meta-Daten, question & test, Lern-Ressourcen etc. zur Verfügung, www.imsproject.org
- *Learning Technology Standards Committee, LTSC* der IEEE spezifiziert eine *Learning Technology Systems Architecture, LTSA*, schlägt einen Standard für *Learning Object Metadata* (LOM), etc, vor, http://ltsc.ieee.org
- *Open Universiteit Nederland* schlägt einen Standard für *Educational Modelling Language* (EML) (unter Verwendung von XML) vor und fokussiert damit auf didaktische Aspekte und XML, http://eml.ou.nl

Da alle diese Standards nicht von vorne herein kompatibel sind, wird Verträglichkeit und die Möglichkeit, einen Standard in einen anderen überführen zu können, immer wichtiger. Ein allumfassender Standard will etwa *Universal Learning Format* (ULF) sein [3].

Hier werden die von der IMS vorgeschlagenen Standards verwendet, weil diese ihrerseits andere Vorschläge wie etwa von IEEE LTSC berücksichtigen und weil IMS ausgearbeitete und dokumentierte XML Implementierungen anbietet.

Standardisierte Tests (IMS)

Die Art und Weisen, den Lernerfolg zu überprüfen, haben sich nicht viel geändert, seit etwa IBM oder Lufthansa ihre ersten CBT-Produkte in den Sechzigern vermarktet haben [4]. Dies spiegelt sich auch in den sogenannten *IMS Basic Item Types* wider, die im folgenden zusammen mit der erwarteten Aktion der Benutzer aufgelistet sind [1].

- LID: Logical identifier (Benutzer klicken auf richtige radio buttons oder check boxes).
- X-Y: X-Y co-ordinate (Benutzer klicken auf den richtigen Bereich auf dem Bildschirm, d.h. in einem Bild).
- STR: String (Benutzer tippen die richtige Antwort ein);
- NUM: Numeric (Benutzer tippen die richtige Zahl ein oder bewegen einen Regler in die richtige Position).
- GRP: Logical groups (Benutzer bewegen mit der Maus Objekte in die richtigen Positionen).

Kombinationen dieser Test-Typen sind zulässig. Für weitere Referenz seien nun die IMS Basic Item Types detailliert.

Beispiele für LID sind:

- Standard True/False (text-based options) choice-based rendering;
- Standard Multiple Choice (text-based options) choice-based rendering;
- Standard Multiple Choice (image-based options) choice-based rendering;
- Standard Multiple Choice (audio-based options) choice-based rendering;
- Standard Multiple Response (text-based options) choice-based rendering;
- Multiple Choice with Single Image (image-based options) Image Hot Spot, IHS-based rendering;
- Multiple Response with Multiple Images (image-based options) Image Hot Spot, IHS based rendering;
- Multiple Choice (slider-based options) slider-based rendering;
- Standard Order Objects (text-based objects) object-based rendering;
- Standard Order Objects (image-based objects) object-based rendering;
- Connect-the-points (image-based) Image Hot Spot, IHS-based rendering.

Beispiele für XY co-ordinate sind:

- Standard Image Hot Spot (single image) Image Hot Spot, IHS-based rendering;
- Connect-the-points (image-based) Image Hot Spot, IHS-based rendering.

Beispiele für STR sind:

- Standard Single Fill-in-Blank Fill-in-Blank, FIB-based rendering;
- Standard Multiple Fill-in-Blank Fill-in-Blank, FIB-based rendering;
- Standard Short Answer (text required) Fill-in-Blank, FIB-based rendering.

Beispiele für NUM sind:

- Standard Integer Fill-in-Blank FIB3-based rendering;
- Standard Real number Fill-in-Blank FIB3-based rendering;
- Numerical entry with Slider slider-based rendering.

Beispiele für GRP sind:

- Standard Drag-and-drop (multiple images) object-based rendering.

Die CBT-Anwendungen etwa des *Cisco Networking Academy Programs* liefern aktuelle Beispiele: In quizzes, reviews, online und final exams gibt es ausschließlich LID2-Tests. Rückmeldungen bestehen z.Zt. nur aus richtig/falsch und einem personalisierten feedback in Form einer Liste derjenigen Kapitel des Stoffes, zu denen der jeweilige Teilnehmer Fragen falsch beantwortet hat.

EINE FELD-STUDIE

Im Rahmen einer kleinen Feld-Studie wurden zwei Mathematik-Vorlesungen, gehalten im seminaristischen Stil, aufgezeichnet. Die Auswahl der Vorlesungen war bestimmt durch die Verfügbarkeit der Aufnahme-Geräte und somit eher zufällig. In Auswertung dieser Aufzeichnungen wurden dann die Fragen von Lehrer und Studierenden klassifiziert.

In Abschnitt weiter wird klar werden, wie der Lehrer versuchte, die Studierenden dabei zu unterstützen, sich verschiedene Arten von Kenntnissen anzueignen [5]:

- Deklaratives Wissen, z.B. Was ist ein Vektor Feld?
- Konzeptuelles Wissen, z.B. die Idee, Funktionsgraphen mit Ebenen zu schneiden, um Funktionen zu visualisieren (Graphen partieller Funktionen, Höhenlinien).
- Prozedurales Wissen, z.B. Wie modelliert man ein 3D Objekt?

Der Lehrer versucht, das Interesse der Studenten durch Fragen zu wecken, durch Fragen Hilfestellung zu geben, durch Fragen den Lernerfolg zu testen, d.h. festzustellen, inwieweit Fakten gelernt, Konzepte verstanden und Fähigkeiten erworben wurden, etwa Modelle zu erstellen oder Algorithmen zu entwerfen. Der Lehrer versucht also, verschiedene Arten zu lernen anzuregen, nämlich:

- Studierende lernen Fakten durch Wiederholung, etc.
- Studierende lernen, mathematische Objekte wiederzuerkennen und darauf zugehörige Methoden anzuwenden, sie werden zu explorativem Lernen angeregt, etwa bei dem Einsatz von Programmen zum Modellieren und Darstellen (rendern) von 3D-Objekten.
- Studierende lernen zu modellieren, z.B. mentale Bilder von 3D Objekten zu konstruieren und schrittweise zunehmend komplexe 3D Objekte zu modellieren [5].

Hintergrund und Gegenstand der Lehrveranstaltungen

Die Lehrveranstaltungen wandten sich an Studierende am Ende ihres zweiten Semesters. Diese Studierenden sollten über grundlegende Kenntnisse linearer Algebra, analytischer Geometrie, der Analysis von Funktionen einer Variablen (einschließlich Potenz- und Fourier-Reihenentwicklung) und gewöhnlicher Differentialgleichungen verfügen. Gegenstand der Lehrveranstaltungen war die Einführung von Funktionen mehrerer Variablen. Dazu wurde eine Reihe möglichst eingängiger Beispiele für Skalar- und Vektor-wertige Funktionen einer und mehrerer Veränderlicher erarbeitet. Das Lernziel kann in Tabelle 1 zusammengefaßt werden.

Der Fall $n = 1$ war schon in der vorangehenden Vorlesung behandelt worden. Zudem hatten sich die Studierenden auch schon mit ersten Beispielen von Skalar- und Vektorfeldern beschäftigt. Die Konzepte *partielle Funktion* und *Höhenlinie* waren eingeführt. Parameter Darstellungen etwa von Geraden und Kreisen in der Ebene oder Schraubenlinien im Raum waren behandelt.

Fragen und Antworten in den Vorlesungen

Im Hinblick auf die beabsichtigte Klassifizierung sei der Ablauf der Lehrveranstaltungen anhand der gestellten Fragen mit ihren Antworten dargestellt. Dazu werden Fragen, Antworten und sonstige Erläuterungen von Lehrer (L) und Studierenden (S) wie folgt kenntlich gemacht.

- **LF**: Fragen des Lehrers.
- **LA**: Antworten des Lehrers.
- **LT**: Material/Text des Lehrers.
- **xA**: *(expected/expert/nil) intendierte und erwartete Antwort der Studierenden, die diese jedoch nicht geben.*
- **SF**: Fragen der Studierenden.
- **SA**: Antworten der Studierenden.
- **ST**: Bemerkungen der Studierenden.

Für die Zwecke dieser Untersuchung ist es ausreichend, nur die erste der beiden aufgezeichneten Vorlesungen in der beschriebenen Weise wiederzugeben und sich dabei auf die Interaktion zwischen Lehrer und Studierenden in Form von Paaren von

Tabelle 1: Lernziel.

n\m	Skalar-Felder	Vektor-Felder		
	1	2	3	4
1	y = f(x)	Parameter Darstellung von Kurven im R†	Parameter Darstellung von Kurven im R‡	
2	w = f(x, y), z.B. Temperatur-Verteilung auf einer Fl che	Matrix-Transformationen des R†	Parameter Darstellung von (Ober-) Fl chen im R‡	
3	w = f(x, y, z) oder w = f(x, y, t), z.B. Temperatur-Verteilung im Raum oder Zeit-abh ngig auf einer Fl che	$\begin{pmatrix} u(x, y, z) \\ v(x, y, z) \end{pmatrix}$	Matrix-Transformationen des R‡oder des R† (in homogenen Koordinaten), Felder	
4	w = f(x, y, z, t), z.B. Zeit-abh ngige Temperatur-Verteilung im Raum	$\begin{pmatrix} u(x, y, z, t) \\ v(x, y, z, t) \end{pmatrix}$	$\begin{pmatrix} u(x, y, z, t) \\ v(x, y, z, t) \\ w(x, y, z, t) \end{pmatrix}$ z.B. Zeit-abh ngige Felder	Matrix-Transformationen des R‡in homogenen Koordinaten

Fragen und Antworten zu konzentrieren. (Auf www.weblearn.hs-bremen.de/risse/papers/ICL2002 enthalten die Dateien MAI020628 und MAI020702 die vollständige audio Dokumentation.)

1. **LF**: was waren die Beispiele für und Einsichten im Umgang mit Funktionen $f:R\supset D\rightarrow W\subset R^m$ beim letzten Mal?
   LT: Lernziele, wie sie beim vorigen Mal vorgestellt wurden ...
2. **LF**: Möchten Sie mehr Beispiele, um die Ergebnisse vom letzten Mal aufzufrischen?
   SA: ja
   LT: lassen Sie uns eine Funktion $f : R \supset D \rightarrow R^m$ – wie gewohnt – als $p : R \supset D \rightarrow R^m$ schreiben, z.B. $p(t) = p_0 + t(p_1 – p_0)$. also n = 1, m = 1, 2.
3. **LF**: was für ein Objekt ist $\{p(t) : t \in R\}$?
   SA: Gerade durch p_0 und p_1
4. **LF**: was ist $p_1 – p_0$?
   SA: Richtungsvektor
   LT: dasselbe gilt nicht nur für Geraden im R^2 oder R^3 ...
5. **LF**: weitere Beispiele?
   SA: nein
   LT: wenden wir uns dem Fall n = 2, m = 1 zu.
6. **LF**: wie heißt eine Funktion z = f(x,y)?
   xA: *Skalarfeld*
   LT: erinnern Sie sich an Skalarfelder wie f(x,y) = x+y oder f(x,y) = x y. Betrachten wir ein weiteres Beispiel, etwa $z = f(x,y) = \sin \sqrt{(x^2 + y^2)}$
7. **LF**: wie ist der Graph einer Funktion $f:R^2\supset D\rightarrow R$ definiert? und um was für ein Objekt handelt es sich?
   SA: graph(f) = $\{(x,y,f(x, y)): (x,y) \in D\}$ ist eine Fläche im R^3
8. **LF:** wie visualisiert man den Graphen von $z = f(x, y) = \sin \sqrt{(x^2 + y^2)}$?
   xA: *durch Schnitte, perspektivische Darstellung*
   LT: noch einmal, um sich ein Bild von f zu machen, bemerkt man, daß $\sqrt{(x^2 + y^2)}$ in die Sinus-Funktion eingesetzt ist...
9. **LF:** wie sehen denn die Höhenlinien zu irgendeinem Niveau z_o aus?
   xA: *Kreise*
10. **LF:** welche (x,y) lösen $z_o = f(x,y) = \sin \sqrt{(x^2 + y^2)}$?
    SA: $\arcsin z_o = \sqrt{(x^2 + y^2)}$
    LT: sei etwa $z_o = 1/2$
11. **LF:** was ist arcsin 1/2?
    LT: zur Erinnerung: $\sin \varphi_i = \sqrt{i}/2$ für $\varphi_i \in \{0, \pi/6 , \pi/4 , \pi/3 , \pi/2\}$, i=0,1,2,3,4
    SA: arcsin 1/2 = $\pi/6$
12. **LF:** welche (x,y) lösen denn nun $(\pi/6)^2= (x^2 + y^2)$?
    LT: denken Sie an Pythagoras oder schreiben Sie $|(x,y)| = |(x,y) – (0,0)| = \pi/6$
13. **LF:** komische Frage?
    SA: ja
    LT: erinnern Sie sich an Vektor-Längen,

Abstände. Denken Sie an Schnitte mit den Koordinaten-Ebenen, Nennen Sie $\pi/6$ etwa r ...
SA: Kreise

14. **LF:** wie sehen die partiellen Funktionen $z = f(x, 0)$ und $z = f(0,y)$ aus?
SA: wie Sinus-Funktionen

15. **LF:** was für Schnitte stellen die Graphen von partiellen Funktionen dar?
LT: ...
SA: Schnitte mit den Koordinaten-Ebenen

16. **LF:** wie sieht denn nun $z = f(x,x)$ aus? was für einen Schnitt stellt der Graph von $z = f(x,x)$ dar?
xA: *Schnitt des Graphen von f mit der Ebene x=y , d.h.* $\mathrm{graph}(f) \cap \{x = y\}$

17. **LF:** was für eine Funktion einer **V**ariablen ist $z = f(x,x)$?
SA: wieder eine Sinus-Funktion

18. **LF:** was für ein Schnitt?
LT: ...

19. **LF:** wie lautet die Gleichung der Schnitt-Ebene?
SA: x=y

20. **LF:** warum ist das eine Ebene?
xA: *allgemeine Ebenengleichung lautet* $x - y + 0\,z = 0$
LT: ...

21. **LF:** wo/wie liegt diese Ebene im Raum?
SA: die Ebene enthält die z-Achse
LT: ...

22. **LF:** wie sieht der Schnitt von x = y mit der x-y-Ebene aus?
SA: Diagonale
LT: ...

23. **LF:** visualisiere graph(f)
LT: ... denken Sie an einen Schnappschuß einer konzentrischen Wasserwelle ...

24. **LF:** wie kann man hier Reibung modellieren?
LT: ... die Amplitude sollte indirekt proportional zum Abstand zum Ursprung sein. Erinnern Sie sich an gewöhnliche Differentialgleichungen, deren charakteristisches Polynom zwei paarweise konjugierte komplexe Nullstellen hat ...

25. **LF:** zur Erinnerung: wie sahen gedämpfte Schwingungen aus?
LT: ...

26. **LF:** Amplitude? ergänze den Exponenten in $e^{(...?...)} \sin(\omega t)$
SA: $e^{t/\tau} \sin(\omega t)$

27. **LF:** wie können wir dies auf die zwei unabhängige Veränderliche übertragen, wo die Dämpfung proportional zum Abstand vom Ursprung ist?
LT: ...
SA: $e^{\sqrt{(x^2+y^2)}} \sin \sqrt{(x^2 + y^2)}$

28. **LF:** wie können wir die Zeit in unsere Funktion einführen?
LT: ...
LT: betrachten wir den Fall n = 3, m = 1, z.B. $z = f(x,y,t) = A_o \cos 2\pi\, (\nu t - \sqrt{(x^2 + y^2)}/\lambda)$ ...

29. **LF:** visualisiere die Funktion f
LT: ...
LF: visualisiere f für ein festes t_o
LT: ...

30. **LF:** wie sieht die Höhenlinie zum Niveau $A_o/2$ und für festes t_o aus?
SA: $\{(x, y, t_o): A_o \cos 2\pi\, (\nu\, t_o - \sqrt{(x^2 + y^2)}/\lambda) = A_o/2\}$

31. **LF:** löse nach x und y auf.
SA: $2\pi\, (\nu\, t_o - \sqrt{(x^2 + y^2)}/\lambda) = \pi/3$
LT: weiter ...
SA: $\nu\, t_o - \sqrt{(x^2 + y^2)}/\lambda = 1/6$
LT: weiter ...
SA: $\nu\, t_o - 1/6 = \sqrt{(x^2 + y^2)}/\lambda$
LT: weiter ...
SA: $(x^2 + y^2) = \lambda^2\, (\nu\, t_o - 1/6)^2$

32. **LF:** was stellen diese Höhenlinien geometrisch dar?
SA: wieder Kreise
LT: ...
LT: betrachten wir nun den Fall n = 2, m = 2, d.h. Funktionen $f : R^2 \supset D \to W \subset R^2$

33. **LF:** wie heißt ein solches f?
SA: **V**ektorfeld
LT: hier gilt $f(x,y) = (u(x,y), v(x,y))$ ...

34. **LF:** was sind $u(x,y)$ und $v(x,y)$?
SA: **S**kalarfelder
LT: Vektorfelder sind also Vektoren von Skalarfeldern, z.B. Matrix-Transformationen wie etwa

$$\begin{pmatrix} x \\ y \end{pmatrix} \to \begin{pmatrix} \cos\delta & -\sin\delta \\ \sin\delta & \cos\delta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

35. **LF:** berechne das Produkt
SA: $(x \cos \delta - y \sin \delta,\ x \sin \delta + y \cos \delta)$

36. **LF:** welche Beziehung besteht zwischen dem Argument-Vektor und dem Funktionswert, hier also dem Produkt?
LT: identifizieren Sie, erinnern Sie komplexe Zahlen, Euler-Gleichung und Additionstheoreme von Sinus und Cosinus ...

37. **LF:** was ist die Polar-Darstellung von (x,y)?
SA: $r(\cos \varphi, \sin \varphi)$

38. **LF:** wie sind dabei r and φ bestimmt?
SA: $r^2 = x^2 + y^2$ und $\varphi = \arctan y/x$

39. **LF:** was erhalten Sie, wenn Sie die Polar-Darstellung des Argumentes in das Produkt einsetzen?
SA: $r(\cos \varphi \cos \delta - \sin \varphi \sin \delta,\ \cos \varphi \sin \delta + \sin \varphi \cos \delta)$

40. **LF:** was erhalten Sie, wenn Sie die Additionstheoreme anwenden?
SA: $r(\cos(\varphi + \delta), \sin(\varphi + \delta))$
41. **LF:** was passiert also mit dem Argument-Vektor geometrisch?
SA: rotiert um δ
LT: Es gibt weitere wichtige Matrix-Transformationen der Ebene, z.B. Skalierung, Scherung und sogar – in homogenen Koordinaten – Translation ... Betrachte im Fall n = 2, m = 3 Vektorfelder $f(x,y) = (u(x,y), v(x,y),w(x,y)) : R^2 \supset D \rightarrow W \subset R^3$ ...
42. **LF:** welche Dimension hat beispielsweise $D = [x_{min},x_{max}] \times [y_{min},y_{max}]$?
SA: zwei-dimensional
43. **LF:** welche Dimension hat– im allgemeinen – W?
xA: *zwei-dimensional*
44. **LF:** was erhalten Sie in W, wenn Sie y bei einem y_o festhalten? Was ist $\{f(x,y_o) : (x,y_o) \in D\}$?
xA: *eine Kurve im* R^3, *Wertebereich einer partiellen Funktion*
LT: ...
45. **LF:** was Dimension hat $\{f(x,y_o) : (x,y_o) \in D\}$?
SA: ein-dimensional
46. **LF:** welche Dimension hat W?
SA: zwei-dimensional
LT: Wertebereiche von Funktionen $f(x,y) : R^2 \supset D \rightarrow W \subset R^3$ sind Flächen im Raum in Parameter-Darstellung ... Betrachten Sie beispielsweise die drei zu einer Ebene im Raum gehörenden Vektoren p_0, r_1, r_2.
47. **LF:** welche Rollen spielen diese drei Vektoren?
SA: Ebene durch p_0 mit Richtungsvektoren r_1 und r_2
48. **LF:** wie bekommt man einen jeden Punkt der Ebene?
xA: p_0 *plus eine Linearkombination von* r_1 *und* r_2
LT: erinnern Sie, daß Ebenen im Raum genauso wie Geraden in der Ebene Hyper-Ebenen sind ...
49. **LF:** wie bestimmen sich die Punkte der **Geraden** im R^2 oder im R^3 durch p_o und mit Richtungsvektor r_1?
SA: $r(t) = p_0 + t\, r_1$ für $t \in R$
50. **LF:** wie bestimmen sich die Punkte der **Ebene** im R^3 durch p_o mit Richtungsvektoren r_1 und r_2?
SA: $r(s,t) = p_0 + s\, r_1 + t\, r_2$ für $s,t \in R$
51. **LF:** wie bestimmen sich die Punkte der **Ebene** im R^3 durch p_0, p_1 und p_2?
SA: $r(s,t) = p_0 + s\,(p_1 – p_0) + t\,(p_2 – p_0)$
52. **LF:** wie läßt sich die Ebene x = y von vorhin modellieren?
LT ...
SA: $r(s,t) = (0,0) + s\,(p_1 – (0,0)) + t\,(p_2 – (0,0))$ mit $p_1 = (1,1,0)$ und $p_2 = (1,1,1)$
LT: die Darstellung ist nicht eindeutig, was passiert, wenn Punkte kollinear sind, ...
53. **LF:** warum repräsentieren $r(s,t) = (0,0) + s\, p_1 + t\, p_2$ und etwa $r'(s',t') = (0,0) + s'\, p_1 + t'\, e_z$ (e_z ist der Einheitsvektor auf der z-Achse) dieselbe Ebene?
LT: ...
LF: wie kann man diese Behauptung verifizieren?
54. **LF:** schwierig?
SA: ja
LF: wollen Sie das zu Hause oder gleich hier erledigen?
SA: gleich hier
LT: ...
LT: fassen wir zusammen, was wir heute gelernt haben ... beim nächsten Mal werden wir...

In der zweiten Vorlesung wurden Matrix-Tranformationen wieder aufgergiffen, Skalierung und Rotationen behandelt. Der Fall n = 2, m = 3 wurde durch weitere Parameter-Darstellungen von Flächen im Raum illustriert. Dabei wurde die Analogie zwischen Kurven in der Ebene und Flächen im Raum aufgezeigt. Als Beispiele dienten Ebenen und die Kugel (Globus), für die Parameter-Darstellungen hergeleitet wurden. Bézier- und B-Spline-Flächen zeigten, wie in der generativen Computer-Graphik beliebige Objekte durch ihre Freiform-Oberflächen modelliert werden. Den Studierenden wurde ans Herz gelegt, die 3D modeling and rendering software für eigene Modellierungen einzusetzen und selber zu erweitern [6].

Frage-Typen

Beim Durchstöbern dieses Materials fallen eine Reihe von Frage-Typen sofort ins Auge. Grundsätzlich zielen alle Fragen darauf ab, bestimmte Aktionen der Studierenden auszulösen. Deshalb seien die Fragen nach den Aktionen klassifiziert, die sie auslösen sollen.

Zugleich sollen Beispiele belegen, daß ein bestimmter Frage-Typ in vielen weiteren Zusammenhängen der Mathematik-Ausbildung vorkommt. Auf diese Weise wird versucht, den Umstand auszugleichen, daß in den zwei ausgewerteten Lehrveranstaltungen naturgemäß bestimmte Aktionen (z.B. Visualisierungen) bevorzugt ausgelöst werden sollten und andere (z.B. algorithmische Modellierung) einfach nicht vorkamen.

- **Modelliere, konstruiere:** hier z.B. modelliere Reibung, modelliere Dämpfung, modelliere Freiform-Oberflächen von 3D Objekten
Modellierung ist – beim Problem-basierten Lernen –selbstverständlich wesentlicher Teil einer *jeden*

Problemlösung erst recht, wenn Probleme aus dem wirklichen Leben angegangen werden sollen, die umgangssprachlich, also nicht mathematisch beschrieben sind. Geometrische Modelle sind zu konstruieren, wie ebenso Algorithmen, Basen, Schätzer, Maße, etc.

- **Klassifiziere:** hier z.B. Höhenlinien,
aber ebenso Klassifikation von Differentialgleichungen, um die Lösungsmethode zu ermitteln.
- **Werte aus, lösen, wende an:** hier z.B. arcsin 1/2, Additionstheoreme, Pythagoras
aber ebenso Systeme linearer Gleichungen lösen, Minima oder Maxima bestimmen, Integrale, Nullstellen etwa von charakteristischen Polynomen, Laplace-Transformation, etc, – letztendlich, *jedes* mathematische Verfahren einsetzen oder *jeden* Algorithmus anwenden
- **Identifiziere, erkenne wieder:** hier z.B. Kurven in der Ebene oder im Raum, Parameter Darstellungen, Skalieren und Rotieren
aber ebenso andere Matrix-Transformationen, (Kreis-) Bahn eines Elektrons im Magnet-Feld, Zeit- und Raum-Forderungen von Algorithmen, etc.
- **Finde Beispiele, exemplifiziere:** hier sollten beide Lehrveranstaltungen Skalar- und Vektorwertige Funktionen einer oder mehrerer Variabler durch anschauliche und einprägsame Beispiele illustrieren.
aber ebenso z.B. um eine erste Idee davon zu bekommen, wie ein Algorithmus arbeitet, um Sätze zu veranschaulichen, um Gegenbeispiele zu finden.
- **Generalisiere, überführe:** hier z.B. Visualisieren von Vektorfeldern durch Koordinaten-weises Visualisieren seiner Skalarfelder
aber ebenso z.B. Kriterien für Stetigkeit o.ä. gelten ebenso für reell- wie auch komplex-wertige Funktionen, Taylor-Entwicklung von Funktionen mehrerer Veränderlicher aus Taylor-Entwicklung von Funktionen einer Veränderlicher.
- **Erinnere:** hier z.B. Polar-Darstellung
Sich die Rotation von Vektoren in the Ebene in Erinnerung zu rufen, ist unabdingbar, wenn man etwa CORDIC-Algorithmen verstehen und implementieren will Solide Grundlagenkenntnisse sind die Vorbedingung für das Erlernen eines *jeden* mathematischen Verfahrens oder Algorithmus'.
- **Verifiziere, falsifiziere:** hier z.B. Überprüfung, ob die Punkte $r(\varphi,\psi) = (\cos\psi \cos\varphi, \cos\psi \sin\varphi, \sin\varphi)$ der Parameter-Darstellung der Kugel nun auf der Einheitskugel liegen
aber ebenso z.B. (Güte von) Approximationen, Verifikation, ob Ergebnisse tatsächlich gegebene Probleme lösen, oder *jede* Betrachtung von Plausibilität
- **Visualisiere:** hier z.B. Skalarfelder oder jede Koordinate eines jeden Vektorfeldes, 3D Modelle aber ebenso z.B. Visualisierung einer *jeden* Lösung, eines *jeden* Algorithmus etc.

FRAGEN AUF TESTS ABBILDEN

Die verschiedenen (Typen von) Fragen sollen nun auf die IMS Basic Item Types abgebildet werden.

Inhärente Beschränkungen – Mögliche Erweiterungen

Die IMS Basic Item Types lassen sich in zwei Teilmengen aufbrechen: Basic Item Types, die eine Liste von möglichen Antworten präsentieren (*mit Tipps*) oder nicht (*ohne Tipps*).

Die Menge der Basic Item Types *ohne Tipps* besteht aus STR1, STR2, STR3, NUM1 und NUM2. Alle anderen Basic Item Types präsentieren mögliche Lösungen, schränken den Lösungsraum ein und geben damit Tipps. In diesem Fall kann der Lerner die richtige Antwort auch durch Ausschluß finden.

Andere Basic Item Types können zu Typen *ohne Tipps erweitert* werden, z.B.

- LID8, i.e. Multiple Choice (slider-based options) - slider-based rendering könnte um reell-wertige Schiebe-Regler erweitert werden. Dann wäre z.B. $\sqrt{2}$ auf einem solchen Schieber etwa mit ticks 0, 1, 2 und 3 näherunsgweise einzustellen oder die häufigste Farbe in einem Bild etwa in Falschfarben-Darstellung auf einem Schieber mit dem entsprechenden Farbbereich angenähert auszuwählen.

Aber auch alle fill-in-blank, FIB Basic Item Types *ohne Tipps* sollten erweitert werden, um für Mathematik brauchbar zu sein:

- STR1, ie Standard Single Fill-in-Blank FIB-based rendering könnte dahingehend erweitert werden, daß Lerner Formeln in einer geeigneten Notation (z.B. *TeX*, *Mathematica*, *Maple*, *mathML*, *OpenMath* [1], etc, möglichst mit graphischer Darstellung) eingeben. Dann lassen sich Fragen wie *Gib eine Berechnungsvorschrift für* π *an!* mit 4 arctan 1 oder 2 arcsin 1 oder 8 arctan 1/8 + 4 arctan 1/7 oder 20 arctan 1/7 + 8 arctan 3/79 oder Nullstelle von $f(x) = e^{i}\pi-1$ o.ä. beantworten. Dann kann die Korrektheit der Antworten Rechner-gestützt entschieden werden.

Übrigens, auch ohne Unterstützung durch ausgewachsene Computer-Algebra-Systemen lassen sich Erweiterungen simpler multiple choice Tests realisieren. JavaScript-Funktionen eingebettet in pdf Dokumente erlauben, beispielsweise nicht nur numerische (s. www.weblearn.hs-bremen.de/risse/MAI/docs/numerik.pdf) und kryptographische (s. www.weblearn.hs-bremen.de/risse/MAI/docs/puzzles.pdf) Algorithmen für eigenständiges, exploratives Lernen zur Verfügung zu stellen, sondern auch die Korrektheit der Manipulation algebraischer Ausdrücke (www.weblearn.hs-bremen.de/risse/MAI/docs/vorkurs.pdf) zu testen.

Der Umstand, daß der Lerner die richtige Antwort auf Fragen vom Basic Item Types *'mit Tipps'* erraten kann, muß durch geeignete Bewertungsschemata berücksichtigt werden, z.B. dadurch, daß falsche Antworten mit Punkt-Abzügen geahndet werden (negative point penalty) – wie etwa in *Känguru*, einem internationalen Mathematik-Wettbewerb für Schüler und Studenten (s. www.weblearn.hs-bremen.de/risse/MAI/docs).

Eingeschränktes Abbilden

Trotz dieser inhärenten Beschränkungen sei versucht, die identifizierten Frage/Antwort-Typen auf IMS Basic Types abzubilden. Es überrascht kaum, daß bestimmte Fragen-Antwort-Paare zu mehreren Kategorien gehören: so kann ein Umstand, den es zu erinnern gilt, zugleich als Beispiel dienen, eine Transfer-Leistung kann ein neues Modell erzeugen, Visualisierungen können der Klassifikation dienen, usw.

- **Modelliere, konstruiere:** Fragen 24, 27, 28, 52, und in der zweiten Lehrveranstaltung: 3D Modellierung
  Es ist offensichtlich, daß der Prozess der Modellierung nicht auf IMS Basic Item Types abgebildet werden kann. Einzig Ergebnisse der Modellierung können abgebildet werden – etwa auf erweitertes STR1, falls das Ergebnis als Menge von Differentialgleichungen vorliegt. Falls das Modell beispielsweise aus einem Graphen besteht, ist eine textuelle Eingabe so umständlich, daß LID3 einen Ausweg mit Tipps darstellt. Sonst muß eine graphische Eingabe verarbeitet, Kanten extrahiert, ein Graph konstruiert und die Isomorphie zwischen Ist- und Soll-Graph geprüft werden! Die notwendigen Schritte der Bildverarbeitung führen dabei aus immanenten Gründen nicht immer zu den gewünschten Ergebnissen.
- **Klassifiziere:** Fragen 3, 6-8, 33, 34
  Alle Fragen vom Typ *Von welcher Art ist eine gegebene Funktion?, Zu welcher Klasse gehört eine gegebene Differentialgleichung?*, etc, werden auf LID1 oder – wenn man partout keine Tipps geben will – auf STR1 abgebildet. Dasselbe gilt für Fragen mit mehreren richtigen Antworten, z.B. *Wie visualisiert man Vektorfelder?*
- **Werte aus, lösen, wende an:** Fragen 10, 11, 12, 19, 30, 31, 35, 39, 40, 52
  Wie im Fall des Modellierens können alle numerischen Ergebnisse oder algebraischen Formeln auf erweitertes STR1 abgebildet werden.. Andersartige Ergebnisse erzeugen dieselben Probleme wie etwa *Wende einen spanning tree Algorithmus an.*
- **Identifiziere, erkenne wieder:** Fragen 4, 7, 15-18, 21, 22, 32, 36, 41-47, 53
  Wenn training das Erfolgsrezept bei der Beantwortung derartiger Fragen darstellt, ist folglich die Implementierung solcher Fragen-Antwort-Paare in e-learning Anwendungen entscheidend. Glücklicherweise ist die Abbildung auf LID oder STR angemessen, wenn STR auch Synonyme berücksichtigt. Diese Anforderung kann aber u.U. nicht erfüllbar sein, da z.B. die Lage einer Ebene im Raum auf viele Weisen verbalisiert werden kann.
- **Finde Beispiele, exemplifiziere:** Fragen 24, 29 und in der zweiten Lehrveranstaltung: finde Beispiele für Freiform-Flächen.
  Ob eine beliebige Freiform-Fläche, die Studierende als Beispiel angeben, die geforderten Eigenschaften aufweist, kann nicht auf Basic Item Types abgebildet werden. Auch hier können LID oder STR als Notbehelf fungieren, wobei erneut STR eine automatische Bewertung verhindert.
  Um entartete Fälle zu identifizieren, mag die Eingabe bestimmter Modell-Parameter genügen, die auf Basic Item Types STR (erweitert) oder NUM abgebildet werden kann.
- **Generalisiere, überführe:** Fragen 27 und in der zweiten Lehrveranstaltung: generalisiere z.B. Bézier-Kurven in der Ebene zu Bézier-Flächen im Raum.
  Verallgemeinerung und Analogie-Schlüsse ersparen Arbeit, erzeugen neue Ideen und Einsichten. Beide fußen jedoch auf Argumenten, die kaum auf Basic Item Types abzubilden sind. Z.B. erlauben Eigenschaften bzgl. Produkten Aussagen über Eigenschaften bzgl. Quotienten.
  Aus $\ln(a\,b) = \ln a + \ln b$
  leitet man $\ln 1 = 0$ *und* $\ln(a/b) = \ln a - \ln b$ *ab.*

kann per Analogie überführt werden in
Aus (f g)' = f' g + f g'
leitet man (f/g)' = (f' g - f g')/g² *ab.*
Auch hier kann wieder nur das Ergebnis, nicht aber der Prozess von Verallgemeinerung und Analogie-Schluß auf Basic Item Types abgebildet werden.

- **Erinnere:** Fragen 1, 6, 7, 20, 25, 26, 33, 36-38, 47-51
Ohne die Wiederbelebung von *stillem* Wissen gibt es keinen (Lern-) Fortschritt, keine Festigung und keine Synthese von altem und neuem Wissen! Gerade e-learning Anwendungen können leicht erworbenes Wissen reaktivieren, wann immer dies individuell notwendig ist. Im Übrigen kann kein Computer-Algebra-System überprüfen, ob eine eingegebene Definition (*Was ist ein lokales Extremum?*) korrekt ist. Unabhängig von Problemen mit der Notation kann die Frage nach einer bestimmten Definition nur auf STR ohne automatische Bewertung abgebildet werden.
- **Verifiziere, falsifiziere:** Fragen 7, 20
Verifikation prüft, ob eine Lösung das vorgegebene Problem löst oder nicht, z.B. ob eine Lösung eine Menge von linearen Gleichungen, von Differentialgleichungen befriedigt, ob eine interpolierende Funktion durch die gegebenen Stützstellen verläuft etc. Falsifikation invalidiert eine Hypothese. Je nach Art der Eingabe, werden Verifizierung und Falsifizierung auf gegebenenfalls erweitertes STR1 abgebildet.
- **Visualisiere:** Fragen 8, 9, 14, 16, 22, 23, 29 und in der zweiten Lehrveranstaltung: bestimmte Schnitte der Kugel, bestimmte Dreiecke in der Kugel etc. Der einzige Kandidat ist LID3, eben *mit Tipps*. Im Prinzip scheint nicht realsisierbar, Korrektheit einer *jeden* graphischen Eingabe automatisch entscheiden zu können!

Mit Beschränkungen der Basic Item Types Umgehen

Die typische Interaktion zwischen Lehrer und Lerner macht die Erweiterung der Basic Item Types in zwei Richtungen notwendig: erstens Eingabe mathematischen Texts (*TeX*, *mathML*, etc) über eine Benutzungsschnittstelle ähnlich der in Computer Algebra Systemen (*Mathematica*, *Maple*, *MuPAD*, etc) zu ermöglichen; und zweitens Computer Algebra Systeme zu nutzen, um Korrektheit der Eingaben der Lerner zu entscheiden.

Die Fähigkeit, Ideen graphisch zu skizzieren, seien es Funktionsgraphen, 2D- oder 3D-Objekte – auch paradoxe – seien es schematisch dargestellte physikalische Modelle, ist unverzichtbar, um Ideen zu formulieren und zu kommunizieren. Solche graphischen Skizzen etwa von Flächen im Raum automatisch auf Korrektheit zu testen, scheint nicht realisierbar. Dieses Beispiel belegt somit schwerwiegende Beschränkungen, denen Tests in e-learning Anwendungen unterworfen sind. Zudem wurde offenkundig, daß Integration von Systemen für Visualisierung, Animation oder Simulation äußerst erstrebenswert ist, um Modelle, Algorithmen, Verfahren in Aktion darzustellen. Bis zu einem gewissen Grad leisten dies die erwähnten *interaktiven Dokumente*, nämlich Dokumente, die rechnen können [8] und so ermöglichen, Algorithmen in Aktion zu erleben, auszuprobieren, Leistung zu vergleichen, Vor- und Nachteile zu identifizieren etc., etwa in Numerik, www.weblearn.hs-bremen.de/risse/MAI/docs/numerik.pdf oder in Kryptographie (z.B. RSA), Kodierung (z.B. CRC) und Wahrscheinlichkeitsrechnung (z.B. Erzeugung & Bewertung von Zufallszahlen), www.weblearn.hs-bremen.de/risse/MAI/docs/puzzles.pdf.

Die Integration von virtuellen Experimenten ist naheliegend: in Physik, Biologie, Medizin oder Computer-Architektur. Wie Prozessoren arbeiten und wie man (pipeline) Prozessoren durch schrittweise Verfeinerung entwirft, demonstriert [9]. Prozessoren um die Möglichkeit, neue Instruktionen zu verarbeiten, zu erweitern, führt zu einem tieferen Verständnis der Arbeitsweise von Prozessoren, der Probleme von Test, Verifikation und Leistungsbewertung sowie des gesamten Entwurfsprozesses. Prozessor-Emulatoren, wie etwa *SPIM* oder *WinDLX*, s. www.weblearn.hs-bremen.de/risse/RST, zeigen anschaulich, wie das Ensemble von Steuerwerk, funktionalen Einheiten, Registern, caches und Speicher Instruktionen verarbeitet (dennoch ergänzen in der *Wissenswerkstatt Rechensysteme* Simulationen nur Vorlesungen und Übungen) [10]. Wieder ist die Integration eines erweiterbaren Emulators in eine e-learning Anwendung für Computer Architektur naheliegend. Dann können Studierende die hardware/software Schnittstelle eines Prozessors modifizieren und die Modifikation einer Nutzwert-Analyse unterziehen.

Andere Reaktionen auf die inhärenten Beschränkungen der Basic Item Types sind einfallsreiche Eigenschöpfungen in e-learning Anwendungen. Z.B. gibt es in Mathe Online [2] neben LID, GRP (sogenannten puzzles) zwei weitere Formen interaktiver Tests, nämlich *Koordinaten ablesen*, d.h. zu trainieren, Objekte in Koordinaten-Systemen zu lokalisieren und zu positionieren, (zweimal NUM1 oder das Inverse von LID6) oder Kreuzworträtsel – nur in der deutschsprachigen Version – (ähnlich zu LID7).

Fragen vom Typ *wo liegt der Fehler?* – ebenfalls nur in der deutschsprachigen Version – sind in Mathe Online als LID2 implementiert.

SCHLUSS UND AUSBLICK

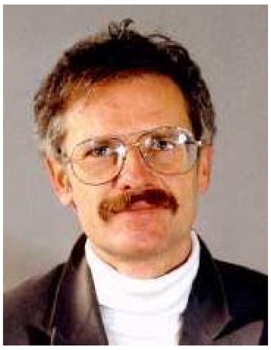
Die Analyse – untermauert durch eine kleine Feldstudie im Bereich der Mathematik-Ausbildung – zeigte, in welchem Maß IMS Basic Item Types verwendet werden können, um Lerner und Lernerfolg zu testen und zu bewerten. Erwartungsgemäß ist die Prüfung deklarativen Wissens in den meisten Fällen leicht auf IMS Basic Item Types abzubilden, während konzeptuelles and mehr noch prozedurales Wissen nur dann mit den durch durch IMS Basic Item Types spezifizierten Hilfsmitteln getestet werden kann, wenn die Prozedur auf das Ergebnis, wenn der Prozess auf einen End-Zustand reduziert wird.

Erweiterungen der IMS Basic Item Types – besonders um etwa mathematischen Text eingeben zu können – verbreitert die Anwendungskontexte in Mathematik z.B. dann, wenn etwa deklaratives Wissen wie *Wie ist* p *definiert?* getestet werden soll und eine Formel als Antwort erwartet wird. Beispiele aus der Computer-Architektur belegen, daß grundsätzlich weitere Computer-unterstützte Test-Schemata nötig sind: IMS Basic Item Types sind so zu erweitern, daß Simulationen, virtuelle Experimente, interaktive Dokumente, etc. in e-learning Anwendungen integriert werden können. Die Analyse von unformatiertem, unstrukturiertem, nicht ausgezeichnetem Text ist aufwändig, aber die Analyse von Bildern wie z.B. graphischen Skizzen ist richtig teuer, wenn nicht gar unerschwinglich. Dennoch sollte in e-learning Anwendungen grundsätzlich möglich sein, automatisch zu entscheiden, ob eine graphische Skizze sagen wir eines 3D Körpers die Ideal-typische Idee dieses Körpers vermittelt.

REFERENZEN

1. IMS Question & Test Interoperability: ASI Best Practice & Implementation Guide, Version 1.2, www.imsglobal.org
2. Embacher, F. und Oberhuemer, P., Mathe online – eine Galerie multimedialer Lernhilfen (2002), www.mathe-online.at
3. www.saba.com/standards/ulf/Specification/Frames/specification.htm
4. Risse, T, e-learning – ideas, intentions, illusions vs status, strategies, stakes. *Proc. 18th Scientific Colloquium*, Pécs, Ungarn (2002), www.weblearn.hs-bremen.de/risse/papers/Kolloq18
5. Holzinger, A., Basiswissen Multimedia – Lernen: kognitive Grundlagen multimedialer Informationssysteme. Würzburg (2000), http://www-ang.kfunigraz.ac.at/˜holzinge/multimedia/multimedia-lernen-theorie.html
6. www.weblearn.hs-bremen.de/risse/MAI/programs/gcc/3D-model
7. Caprotti, O., Cohen, A.M., Cuypers, H. und Sterk, H., OpenMath Technology for Interactive Mathematical Documents. Eindhoven University of Technology (2002), www.win.tue.nl/˜hansc/lisbon.pdf
8. Risse, T., Interactive documents - yet another way to activate students in mathematics. *Proc. ICL2001 Workshop Interactive Computer Aided Learning – Experiences and Vision*, Villach, Osterreich (2001), www.weblearn.hs-bremen.de/risse/papers/ICL2001
9. Patterson, D.A. und Hennessy, J.L., *Computer Organisation & Design – The Hardware/Software Interface*. San Francisco: Morgan Kaufman (1997).
10. Lucke, U., Tavangarian, D. und Vatterrott, H.R., The use of XML for the development of an adaptive multimedia teaching and learning system. *Proc. World Congress on Networked Learning in a Global Environment (NL2002)*, Berlin, Deutschland (2002), www.wwr-project.de/de/infopool/veroeffentlichungen.htm

BIOGRAPHIE



Thomas Risse ist Diplom-Mathematiker und promovierter Physiker. Derzeit arbeitet er als Professor für Rechnerstrukturen, Computer-Graphik und Mathematik im Fachbereich Elektrotechnik & Informatik der Hochschule Bremen. In der Forschung war er z.B. als visiting scientist am T.J. Watson Research Center, Yorktown Heigths der IBM u.a. zu den Themen: Fehler-tolerante Systeme und Computer-Graphik tätig. Er konnte Erfahrungen in seinen praktischen Tätigkeiten u.a. in den Bereichen Datenschutz und Datensicherheit sowie Medizin-Informatik sammeln.