

Pragmatische Empfehlungen zur Entwicklung von interaktiven Lernumgebungen

Ruedi Arnold
Institut für Pervasive Computing
ETH Zürich
CH-8092 Zürich
rarnold@inf.ethz.ch

Werner Hartmann
Zentrum für Bildungsinformatik
PH Bern
CH-3012 Bern
werner.hartmann@phbern.ch

Abstract: Computergestützten interaktiven Lernumgebungen wird seit den 70er Jahren ein grosses Potential vorausgesagt. Die Realität sieht meist anders aus. Viele Lernprogramme beschränken sich auf „Drill & Practice“, sie sind kaum wirklich interaktiv. Andere sind trotz hohem Entwicklungsaufwand oft schon nach wenigen Jahren aus technischen Gründen gar nicht mehr betriebsfähig.

Eine nachhaltige Entwicklung interaktiver Lernsoftware muss gleichzeitig sowohl der Schulpraxis als auch den Regeln der Softwareentwicklung und dem Wissensstand der Lehr- und Lernforschung Rechnung tragen. Eine solche interdisziplinäre Vorgehensweise nimmt bewusst eine wissenschaftliche Unschärfe in Kauf und setzt sich damit der Kritik der Scientific Communities der einzelnen Fachbereiche aus.

Am Beispiel von InfoTraffic, einer neuen Sammlung von kleinen Lernumgebungen zur Logik, zu Warteschlangen und zu dynamischen Systemen, illustrieren wir diesen ingenieurwissenschaftlich geprägten Ansatz, der sich in der Praxis bewährt hat.

1 Interaktive Lernumgebungen – Erwartungen und Enttäuschungen

Mitte der 20er Jahre des vergangenen Jahrhunderts gingen die ersten Schulfunksendungen in den Äther. Die Erwartungen an das neue Medium waren gross. Patrik Wülser [Wü06], Journalist beim Schweizer Radio DRS schreibt:

Vorträge, Sprachkurse und Beratungen für fast alle Lebenslagen prägten von Beginn an die Programme, wie beispielsweise ein Blick in die Sendewoche vom 5. bis 11. Februar 1927 zeigt. Radio Zürich bot Kinder-, Jugend-, Schüler- und Frauenstunden an, gab eine Englischlektion, liess den Vortragsdienst der Volkshochschule zweimal sprechen und veranstaltete einen bunten Strauss

von Vorträgen wie „Klassische und moderne Bildhauerei“, „Winterschnitt am Kernobst“, die „Physik des Mondes“ oder „Zürcher Verkehrsfragen“.

Begleitet waren die Schulfunksendungen durch Initiativen zur Ausrüstung von Schulen mit Empfangsgeräten, quasi „Schulen ans Radio“.



Abbildung 1: Schulen ans Radio. (Quelle: Archiv Radio DRS, Zürich.)

Mit dem Aufkommen des optischen Mediums Fernsehen verlor das Radio seine Faszination, Schulfernsehen war angesagt, der Schulfunk spielte nur noch eine marginale Rolle. Mit dem Aufkommen des Computers in den 60er Jahren erwuchs auch dem Schulfernsehen ernsthafte Konkurrenz, welche durch den Boom des Internet noch weiter verstärkt wurde. Lernsoftware, interaktive Lernumgebungen in Form web-basierter Applets bis hin zu Edutainment versprachen kostengünstiges, individualisiertes, orts- und zeitunabhängiges Lernen samt adaptivem Benutzerfeedback. Inzwischen ist klar, dass sich viele dieser Erwartungen nicht erfüllt haben.

Betrachtet man die Nutzung der drei Medien Radio, Fernsehen sowie Computer und Internet zu Unterrichtszwecken im Rückblick, fallen verschiedene Parallelen auf:

Mangelnde Interaktivität: Die Lernenden finden sich beim Einsatz aller drei Medien zu oft in der Rolle passiver Konsumenten wieder. Bei Lernsoftware muss die Interaktion über reine Navigation hinaus gehen. Lernende sollen aktive Kontrolle über den Inhalt, Ablauf und Aspekte der Präsentation besitzen. Brenda Laurel [La93] bringt es sehr schön auf den Punkt: „You either feel yourself to be participating in the ongoing action of the representation or you don't.“

Interessant in diesem Zusammenhang ist, dass die Fernsehanstalten heute zwecks stärkerer Kundenbindung vermehrt auf interaktives Fernsehen setzen, etwa mit Video on Demand oder Teleshopping.

Hoher Entwicklungsaufwand, geringe Halbwertszeit: Bei diesen drei Medien ist der Aufwand für die Erstellung von Lehr- und Lernmaterialien sehr gross. Der Entwicklungsaufwand rechtfertigt sich deshalb nur bei langfristig bedeutungsvollen Themen und einem breiten Zielpublikum.

Für das Erstellen professioneller Radio- und Fernsehbeiträge reichen Amateurkenntnisse nicht aus. Das Gleiche gilt für computergestützte Lernumgebungen. Die Jugendlichen haben aufgrund ihres von dauerndem Mediengeriesel geprägten Lebensumfelds hohe Erwartungen und mit bescheidenen Mitteln erzeugtes Edutainment verkommt schnell zu „Edupainment“.

Auch die Wartung und Aktualisierung von radio-, fernseh- oder computergestützten Unterrichtseinheiten ist anspruchsvoller als von klassischen gedruckten Lehrmitteln. Im Fall von computergestützten Lernumgebungen führen die kurzen Entwicklungszyklen bei Soft- und Hardware in schöner Regelmässigkeit dazu, dass Unterrichtseinheiten nach wenigen Jahren rein technisch nicht mehr betriebsfähig sind.

Hohe Anforderungen an die Infrastruktur: Papier und Schreibzeug, sowie Lehrmittel in Buchform können „just in time and anywhere“ eingesetzt werden. Auch bei der Wandtafel handelt es sich um eine relativ günstige, mehrfach verwendbare und langlebige Unterrichtshilfe.

Im Unterschied dazu sind der Einsatz von Radio, Fernseher sowie Computer und Internet mit grossen Investitionen in die Infrastruktur verbunden. Für Lehrpersonen ist die Handhabung dieser Werkzeuge zudem nicht einfach und mit Fallstricken versehen. Bei computergestützten Lernumgebungen muss oft noch zusätzliche Software installiert werden.

Im Folgenden legen wir das Schwergewicht auf die Entwicklung computergestützter Lernumgebungen für den Informatikunterricht. Themen aus der Informatik sind naheliegende Unterrichtsgegenstände. Zum einen lassen sich viele informatische Sachverhalte einfach formalisieren und damit überhaupt erst für einen Computer fassbar machen, zum anderen sind Informatiklehrer auch mit der Entwicklung und dem Einsatz von Softwaresystemen vertraut.

Die Informatik hatte auch eine Vorreiterrolle beim Einsatz des Computers im Unterricht. Zu den wichtigen frühen Projekten zählen Seymour Papert's Aktivitäten rund um die Programmiersprache LOGO für Kinder oder John Kemeny and Thomas Kurtz's Program-

miersprache BASIC. Noch einen Schritt weiter ging in den 70er Jahren das System PLATO (Programmed Logic for Automated Teaching Operations), welches während rund zwei Jahrzehnten regelmässig bei Tausenden von Studierenden zum Einsatz gelangte.

Die teils euphorischen Berichte aus den 70er Jahren über das Potential von CAI (computer aided instruction) unterscheiden sich nur wenig von heutigen Einschätzungen. Jürg Nievergelt [Ni75], selbst im Projekt PLATO engagiert, warnte aber bereits 1975 vor übertriebenen Erwartungen, unter anderem:

Restriction to a few fixed teaching strategies appeared to be unreasonable. Programmed instruction and drill in particular, with their rigid control of the dialog by the program, should yield to (or at least not exclude) modes where the user controls the dialog, such as inquiry and simulation. [...]

Resources had been diluted into too many projects of insufficient size; CAI research and development should be carried out by sizable groups of system designers and authors.

Nievergelt schliesst mit folgenden Empfehlungen:

I came to the conclusion that there is no systematic body of knowledge which is of relevance to such a task. I am afraid that this paper does not change this situation at all. The advice I might give to someone intent on building a computer-based instructional system could be summed up in a few phrases: get the best terminals you can pay for, good programmers, try everything out in actual instruction as soon as possible, and follow your nose.

2 Das Dilemma: Wissenschaftlichkeit oder Wirkung?

Man mag die vorangehenden, 30 Jahre zurückliegenden Empfehlungen von Nievergelt belächeln, sie widerspiegeln aber nur das Spannungsfeld zwischen einem wissenschaftlichen Anspruch und gleichzeitiger Wirksamkeit auf die Schulpraxis.

In den letzten Jahren entstand in der Schweiz eine ganze Reihe interaktiver Lernumgebungen für den Informatikunterricht. Der programmierbare Marienkäfer Kara [RNH04] zum Einstieg in die Programmierung ist das wohl prominenteste Beispiel. Weitere Beispiele sind Graphbench [BN04] (NP-Vollständigkeit und Graphen-Algorithmen) oder Soekia (eine didaktische Suchmaschine). Die Lernumgebungen sind frei und inklusive Begleitmaterialien auf dem Bildungsserver SwissEduc [Sw07] verfügbar.

Bei der Entwicklung dieser Umgebungen haben wir das Spannungsfeld zwischen sogenannter Wissenschaftlichkeit und Bildungsinnovation mehrfach erlebt. Aus Sicht der Lehr- und Lernforschung fehlt den Lernumgebungen ein wissenschaftlich begründeter Wirkungsnachweis. Die Komplexität interaktiver Lernumgebungen mit ihren vielen Variablen setzt aber einer methodisch abgestützten Evaluation enge Grenzen. Gabi Reinmann führt in [Re06] aus, wie Bildungsforscher zu fast zwanghafter Differenzierung und Kontrolle im methodischen Design von Studien neigen, um so zu besser verallgemeinerbaren

Aussagen zu kommen. Die Folge seien artifizielle Lernumgebungen, die für den Unterrichtsalltag bedeutungslos sind.

Aus der Sicht der Ingenieurwissenschaft Informatik fehlt bei schultauglichen Lernumgebungen ebenfalls der Forschungsaspekt. Das behandelte Thema wird durch das Curriculum der Schulen festgelegt, wodurch es sich naturgemäss nicht um ein aktuelles Forschungsgebiet handelt. Zudem gilt bei der Entwicklung von Lernumgebungen der Grundsatz „so einfach wie nur möglich“. Die Benutzerschnittstelle soll sich auf das Allernötigste beschränken, damit im Unterricht keine Zeit mit der Einarbeitung in die Systembedienung verloren geht. Auch die Softwarearchitektur soll schlank sein, damit die spätere Wartung der Lernumgebung im Rahmen der meist nur sehr beschränkt vorhandenen Ressourcen möglich bleibt.

Fassen wir das Dilemma zusammen: Wer Lernumgebungen entwickelt, die das Lehren und Lernen nachhaltig beeinflussen, wird fast zwangsmässig von den Erziehungswissenschaften als unwissenschaftlich abgestempelt. In den Augen der Informatik auf der anderen Seite fehlt solchen Lernumgebungen die Innovation. Wer funktionierende interaktive Lernumgebungen entwickelt, die im Unterricht auf breiter Basis und über einen längeren Zeitraum zum Einsatz kommen, begibt sich aus Sicht der Scientific Community also fast zwangsmässig ins Abseits.

3 InfoTraffic – Fallbeispiel interaktiver Lernumgebungen

Der Entwurf praxistauglicher computergestützter Lernumgebungen setzt unserer Ansicht und Erfahrung nach die interdisziplinäre Mitwirkung von drei verschiedenen Personengruppen voraus: Themen, Aufgabenstellungen und Einsatzszenarien von Lernumgebungen müssen von tätigen Lehrerinnen und Lehrern mit grosser Schulpraxis festgelegt werden. Das effiziente Erstellen von qualitativ hoch stehenden interaktiven Lernumgebungen mit vertretbarem Kostenaufwand und einer grossen Halbwertszeit ist eine Ingenieuraufgabe und gehört in die Verantwortung von Softwareentwicklern. Das Evaluieren der Wirksamkeit interaktiver Lernumgebungen ist eine schwierige Aufgabe und bedarf einer besonderen Methodik, welche nur die Bildungsforschung leisten kann.

Ist in einem Entwicklungsteam eine der obigen Personengruppen nicht vertreten, sind Fehlentwicklungen vorprogrammiert. In den letzten zehn Jahren wurden immer wieder mit grossem Aufwand E-Learning Systeme entwickelt, für die es in der Schulpraxis gar keinen Bedarf gab. Wurden Schulpraktiker oder Erziehungswissenschaftler selbst als Programmierer tätig, bestand das Ergebnis nicht selten in benutzerunfreundlichen und technisch nur bedingt lauffähigen Programmen.

Im Folgenden beschreiben wir anhand von InfoTraffic [ALH07], [AH07] unseren pragmatischen Ansatz beim Erstellen interaktiver Lernumgebungen für die Informatik. InfoTraffic besteht aus drei Programmen zu den Themen Aussagenlogik (LogicTraffic), Warteschlangentheorie (QueueTraffic) und dynamische Systeme (DynaTraffic) und ist inklusive Unterrichtsmaterialien online unter [Sw07] frei verfügbar.



Abbildung 2: Lernumgebung LogicTraffic.

3.1 LogicTraffic – Aussagenlogik und Sicherheit bei Strassenkreuzungen

LogicTraffic bietet eine alltagsnahe Einführung in die Aussagenlogik, z.B. Boole'sche Operatoren, Wahrheitstabellen und Normalformen. Die Grundidee besteht darin, mittels aussagenlogischer Formeln die Ampeln einer Strassenkreuzung so zu steuern, dass zwischen einzelnen Spuren keine Kollisionen auftreten können. Aussagenlogik wird nicht abstrakt und formal als Formelsprache eingeführt, sondern spielerisch anhand einer Situation aus dem Alltag.

Die Programmoberfläche von LogicTraffic (Abbildung 2) besteht im Wesentlichen aus der Darstellung einer Strassenkreuzungen, einer Wahrheitstabelle, der Formel zur Wahrheitstabelle, einem Formeleditor und Steuerelementen.

3.2 QueueTraffic – Warteschlangen bei Strassenkreuzungen

QueueTraffic führt am Beispiel der Verkehrssteuerung wichtige Grundbegriffe der Warteschlangentheorie wie Ankunftsrate, Durchsatz oder Verteilungen ein. Parameter wie das Verkehrsaufkommen oder die Länge von Grünphasen können verändert werden und die Auswirkungen lassen sich direkt in der Simulation visuell und statistisch beobachten und analysieren.

Die Programmoberfläche von QueueTraffic (Abbildung 3) besteht im Wesentlichen aus der Darstellung einer Strassenkreuzung, einer Anzeige für Daten und Statistiken, sowie je einem Bereich zur Kontrolle von Verkehr, Verkehrsaufkommen und Simulation.

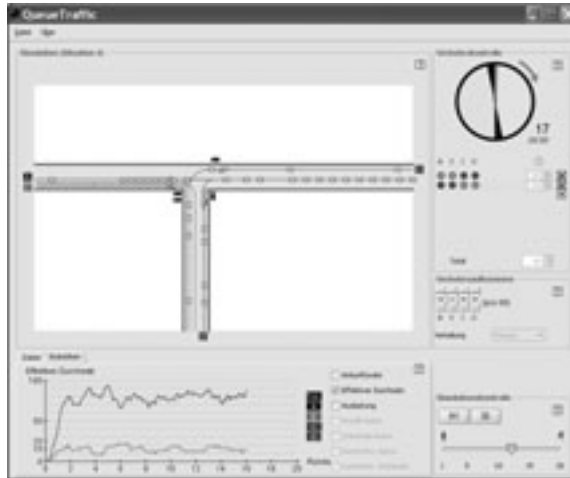


Abbildung 3: Lernumgebung QueueTraffic.

3.3 DynaTraffic – Dynamische Systeme und Verkehrsaufkommen

DynaTraffic simuliert das Verkehrsaufkommen in einem Strassensystem mit mehreren Kreuzungen. Im Zentrum stehen Themen wie stationärer Zustand, Adjazenzmatrix, Markovkette oder Kantengraph.

DynaTraffic ist noch in Entwicklung, Abbildung 4 zeigt einen Entwurf der geplanten Programmoberfläche, die fünf Bereiche beinhaltet: Darstellung eines Strassensystems, Markovmodell der Situation als Graph und als Übergangsmatrix, Visualisierung der aktuellen Verkehrsverteilung und Simulationskontrolle.

4 Pragmatische Empfehlungen bei der Entwicklung interaktiver Lernumgebungen

Gute interaktive Lernumgebungen bzw. E-Learning Systeme ganz allgemein erfüllen im Idealfall drei übergeordnete Kriterien: (a) Die Lernumgebungen bringen einen didaktischen Mehrwert; die Schüler lernen mehr und besser. (b) Die Lernumgebungen schaffen einen organisatorischen Mehrwert, der Unterricht wird für den Lehrer und die Schüler einfacher. (c) Die Lernumgebungen rechnen sich auch ökonomisch, der Unterricht wird für die Bildungsinstitution „billiger“. Es wäre vermessen zu glauben, alle drei Ziele würden sich gleichzeitig erreichen lassen. Wir sind aber überzeugt, dass die nachfolgenden pragmatischen Empfehlungen zumindest ein Schritt in die richtige Richtung sind. Jede der Empfehlung kommentieren wir kurz aus Sicht der Lernumgebung InfoTraffic.

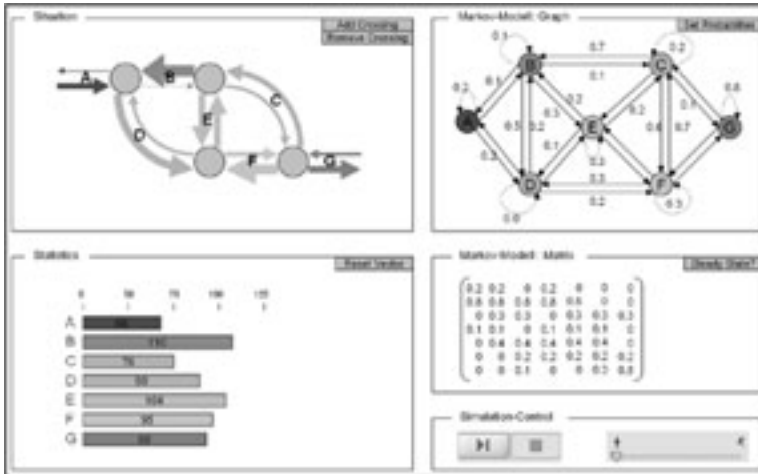


Abbildung 4: Mock-up GUI der Lernumgebung DynaTraffic.

1. Braucht es dazu überhaupt den Computer?

Es macht keinen Sinn, Lernumgebungen zu entwickeln für Themen, die man genau so gut oder besser ohne Computer vermitteln kann. Speziell interessant sind deshalb gerade abstrakte und schwierige Themen, bei denen im Unterricht durch Individualisierung den unterschiedlichen Kenntnissen und Lerntempi Rechnung getragen werden kann. Reine „Drill & Practice“ Übungen können oft einfacher auf Papier durchgeführt werden.

In der Schule ist der Logikunterricht meist unnötig abstrakt. Der intuitive und alltagsnahe Zugang über die Steuerung von Verkehrskreuzungen in LogicTraffic stellt einen didaktischen Mehrwert dar.

2. Ist das Unterrichtsthema auch in 10 Jahren noch relevant?

Der Entwicklungsaufwand für interaktive Lernsoftware ist im Allgemeinen gross und rechtfertigt sich nur für längerfristig relevante Themen.

Logik begleitet uns im Alltag und wird auch in Zukunft in vielen Wissenschaftsgebieten von zentraler Bedeutung sein.

3. Use-Cases: Ist Interaktivität möglich?

Ein hoher Grad an Interaktivität (zum Beispiel gemäss der Taxonomie von Schulmeister [Sc03]) ist ausschlaggebend für die Qualität einer computergestützten Lernumgebung. Das Lesen von Bildschirmtexten oder das Betrachten einer Animation allein löst noch keinen Lernprozess aus.

Nicht jedes Thema eignet sich für eine computergestützte Mensch-Maschinen-Interaktion. Es empfiehlt sich deshalb bereits in der Spezifikationsphase des Projektes Musteraufgaben zusammenzutragen, welche mithilfe der geplanten Lernumgebung bearbeitet werden sollen. In der Sprache der Software-Entwicklung stellen diese Aufgaben Use-Cases für

das Programm dar. Aufgrund der Musteraufgaben kann abgeschätzt werden, ob die Lernumgebung zu einer wirklichen Interaktion mit den Lernenden führt und ob verschiedene kognitive Stufen angesprochen werden.

„Analysieren Sie die vorliegende Verkehrssteuerung. Mit welchen Massnahmen kann die globale Wartezeit minimiert werden?“ lautete eine Musteraufgabe bei QueueTraffic. Solche Aufgabenbeispiele legen fest, welche Funktionen die Lernumgebung anbieten muss.

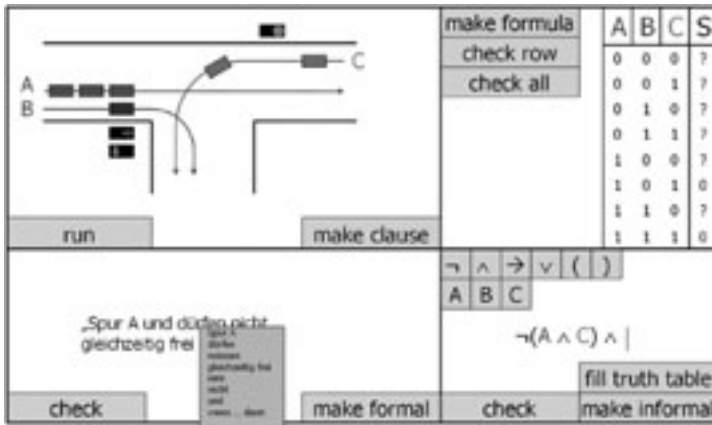


Abbildung 5: Mock-up GUI Entwurf der Lernumgebung LogicTraffic.

4. Paper Based Prototyping

Aus der Softwareentwicklung ist bekannt, dass ein früher Einbezug von Aspekten der Benutzerschnittstelle entscheidend Kosten spart. Bei Lernumgebungen kommt der Benutzerschnittstelle eine besonders hohe Bedeutung zu. Es lohnt sich, so lange als möglich nur auf Papier zu arbeiten. Neben Papier kommen in dieser Phase auch Klebband, Post-It, Schere, Flipchart und andere einfache Werkzeuge zum Einsatz. Für das schnelle Erstellen und Variieren von Programmoberflächen eignen sich auch Präsentationsgrafik-Werkzeuge.

Bei der Entwicklung von InfoTraffic haben wir lange konsequent nur mit Papierentwürfen gearbeitet. An einem Prototypen auf Papier lassen sich mögliche Aufgaben und Einsatzszenarien sehr gut analysieren, durchspielen und diskutieren. Einer der Hauptvorteile von Paper Prototyping: In Diskussionen trauen sich die Leute eher, kritische Bemerkungen anzubringen oder Teile eines Projektes ganz in Frage zu stellen. Liegt der Prototyp bereits als mit grossem Aufwand erstelltes Programm vor, ist die Hemmschwelle für grundlegende Kritik viel höher. Eine gute Einführung in die Methode „Paper Prototyping“ findet sich in [Sn03].

Abbildung 5 zeigt einen frühen PowerPoint-Prototypen von LogicTraffic. In diesem Mock-up war auch noch ein Fenster für die natürlichsprachliche Formulierung aussagenlogischer Formeln vorgesehen. Diese Option erwies sich beim Lösen von Aufgaben anhand des Papier-Prototypen als schwierig formalisierbar und wurde fallen gelassen.

5. Rapid Prototyping

Nachdem die Use-Cases und die Benutzerschnittstelle auf Papier festgelegt worden sind, wird ein erster Prototyp implementiert. Dieser soll sobald als möglich einzelnen ausgewählten Testpersonen – Schülerinnen und Lehrern – gezeigt und experimentell erprobt werden.

Abbildung 6 zeigt die erste öffentlich vorgestellte Version von LogicTraffic. Die Unterschiede zur aktuellen Version (Abbildung 2) sind offensichtlich. Auch die interne Programmstruktur wurde überarbeitet und Verkehrskreuzungen werden beispielsweise nicht mehr durch Bilddateien erzeugt, sondern automatisch aus einer xml-Datei generiert.

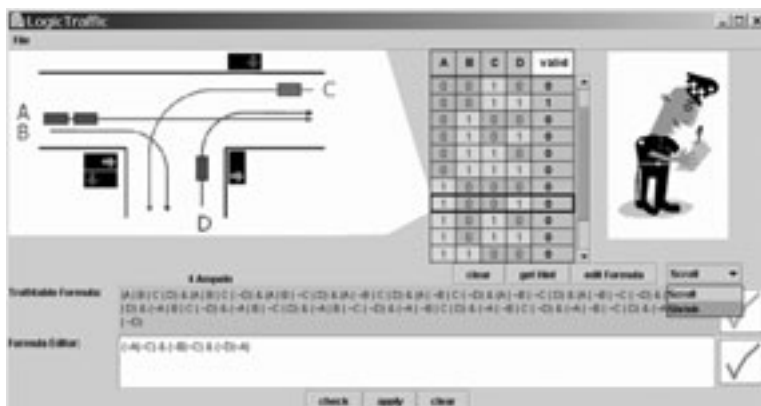


Abbildung 6: Ältere Version der Lernumgebung LogicTraffic

6. Technische Anforderungen: so einfach wie nur möglich

Der grosse Entwicklungsaufwand für eine Lernumgebung rechtfertigt sich nur, wenn die Software einfach auf verschiedenen Plattformen eingesetzt werden kann und wenn auch die Wartung längerfristig sichergestellt ist. Einfachheit und Verzicht auf unnötige Funktionalitäten sind deshalb gefragt.

InfoTraffic wurde in Java programmiert und benötigt nur eine Java Runtime Umgebung (JRE); diese ist heute verbreitet auf Schulrechnern zu finden.

7. Frühzeitige Erprobung

Sobald als möglich sollen erste Tests mit ganzen Schulklassen durchgeführt werden. Diese Tests zwingen dazu, konkrete Aufgabenstellungen und begleitendes Unterrichtsmaterial auszuarbeiten, und decken in der Regel eine ganze Reihe von Schwachstellen auf. Schüler denken und handeln oft anders, als Lehrpersonen sich das vorstellen!

Bei allen Teilen von InfoTraffic haben die Erprobungen beispielsweise ergeben, dass die Schüler sich auch ohne ausführliche Bezeichnungen von einzelnen Buttons etc. zurechtfinden und mit englischsprachigen Bezeichnungen keine Probleme haben.

8. Sparsamkeit bei der Benutzerschnittstelle

Die Benutzeroberfläche einer Lernumgebung soll im Idealfall selbsterklärend sein. Nach ersten Erprobungen ist es deshalb wichtig, das GUI der Lernumgebung einem Review-Prozess zu unterwerfen. Bei jedem Button, jeder Beschriftung usw. ist kritisch zu hinterfragen, ob dieses Element wirklich benötigt wird und ob es an der richtigen Stelle platziert ist.

Unsere Erfahrung zeigt, dass im Laufe dieser kritischen Analyse bis zur Hälfte aller ursprünglich vorhandenen Elemente wegfallen. Erstellt man professionell Lernumgebungen, sollte man sich auch mit wissenschaftlichen Erkenntnissen aus dem Bereich Multimedia Learning auseinandersetzen (vgl. zum Beispiel [Ma01]).

9. Verbreitung der Lernumgebung

Steht die Lernumgebung in einer ersten stabilen Version zur Verfügung, ist die Arbeit nicht abgeschlossen. Um überhaupt eine Wirkung der Lernumgebung im Unterricht zu erreichen, muss die Lernumgebung Schülern, Lehrern und weiteren Benutzern zur Verfügung gestellt werden, beispielsweise über eine bekannte Website. Die Verteilkanäle sind schon vorgängig genau abzuklären, es reicht heute längst nicht mehr, einfach eine Domain – zum Beispiel in unserem Fall www.infotraffic.ch – aufzuschalten.

Unsere Erfahrung zeigt, dass das Angebot von Lernumgebungen mit frei verfügbaren didaktischen Begleitmaterialien (z.B. Einführungsvortrag, Anleitung zur Bedienung, Aufgabenblätter, didaktische Hintergrundinformationen) begleitet werden muss.

10. Sicherstellung von Unterhalt und Kontinuität

Nach der Veröffentlichung, also gewissermassen nach dem Übergang in den Einsatz, muss (wie bei Informatikprojekten üblich) der Unterhalt sichergestellt werden. Dies beinhaltet insbesondere das Beheben etwaiger Fehler und die Umsetzung allfälliger Erweiterungen und Anpassungen an neue Software-Versionen. In dieser Phase gilt es auch, Vertrauen zu schaffen. Fehlende Zeit ist eines der Hauptprobleme im Berufsalltag der meisten Lehrpersonen. Eine Lehrerin überlegt sich deshalb gut, ob sie den Aufwand für die didaktische Aufbereitung eines Themas basierend auf einer neuen Lernumgebung leisten soll oder nicht. Sie wird abwägen, ob die Software in ein paar Jahren noch zur Verfügung stehen wird. Viele interaktive Lernumgebungen entstehen im Hochschul Umfeld im Rahmen studentischer Arbeiten und die geforderte Kontinuität ist deshalb nicht sichergestellt. Hier gilt es, rechtzeitig Massnahmen zu ergreifen.

Üblicherweise werden auch Wünsche nach Erweiterungen der Software an ein Entwicklerteam herangetragen. Hier heisst es vorsichtig zu sein: Die Qualität einer Lernumgebung misst sich nicht an der Fülle der angebotenen „Features“. Häufige Überarbeitungen und Erweiterungen verunsichern die Benutzer und führen dazu, dass bereitgestellte Unterrichtsmaterialien aktualisiert werden müssen. Und jede Erweiterung erhöht auch die Komplexität des Programmcodes und erschwert die Wartung. „Weniger ist mehr“ lautet deshalb das Motto.

5 Zusammenfassung und Fazit

Die Geschichte zeigt, dass die Entwicklung qualitativ hoch stehender interaktiver Lernumgebungen ein aufwändiges und schwieriges Unterfangen ist. Entscheidend für den Erfolg einer Lernumgebung sind unserer Überzeugung nach weniger die zur Verfügung stehenden Ressourcen, sondern das Zusammenspiel von erfahrenen Lehrpersonen, versierten Softwareentwicklern und Fachleuten aus der Bildungsforschung, welche bereit sind, sich auf eine nach ingenieurwissenschaftlichen Prinzipien funktionierende Forschungsarbeit einzulassen.

Der Einsatz computergestützter Lernumgebungen ist komplex und lässt sich nicht mittels einer den strengen wissenschaftlichen Kriterien der Bildungsforschung gerecht werdenden Methodik evaluieren. Gefragt ist von allen Beteiligten die Bereitschaft zu interdisziplinärer Zusammenarbeit und ein pragmatisches Vorgehen. Ganz speziell braucht es auch den Mut, sich der fast sicheren Kritik aus der Scientific Community der Lehr- und Lernforschung auf der einen Seite und der Informatik auf der anderen Seite auszusetzen. „Try everything out and follow your nose“ ist in akademischen Kreisen verpönt, aber ein erfolgsversprechender Ingenieuransatz, wenn es um Innovation im Bereich E-Learning geht.

Literaturverzeichnis

- [AH07] R. Arnold und W. Hartmann. LogicTraffic - Logik in der Allgemeinbildung. Informatik-Spektrum, Volume 30, Number 1, Seiten 19–26, 2007.
- [ALH07] R. Arnold, M. Langheinrich und W. Hartmann. InfoTraffic - Teaching Important Concepts of Computer Science and Math through Real-World Examples. In Proceedings of the ACM SIGCSE Technical Symposium 2007, Covington, Kentucky, USA, March 2007.
- [BN04] M. Braendle und J. Nievergelt. Tackling Complexity: A Case Study on Educational Software. World Conference on E-Learning in Corp., Govt., Health., and Higher Ed. (ELEARN), Volume 2004, Issue 1, Seiten 1794–1799, 2004.
- [La93] B. Laurel. Computers as Theatre. Addison-Wesley Longman, 1993.
- [Ma01] R. Mayer. Multimedia Learning. Cambridge University Press, 2001.
- [Ni75] J. Nievergelt. Interactive Systems for Education: The New Look of CAI. Proc. IFIP Conf. on Computers in Education, 53, No. 4:465–472, 1975.
- [Re06] G. Reinmann. Nur „Forschung danach“? Vom faktischen und potentiellen Beitrag der Forschung zu alltagstauglichen Innovationen beim E-Learning. Arbeitsbericht Universität Augsburg, Nr. 14, 2006.
- [RNH04] R. Reichert, J. Nievergelt und W. Hartmann. Programmieren mit Kara. Ein spielerischer Zugang zur Informatik, (ergänzte Neuauflage). Springer, Berlin, Dezember 2004.
- [Sc03] R. Schulmeister. Taxonomy of Multimedia Component Interactivity. A Contribution to the Current Metadata Debate. Studies in Communication Sciences. Studi di scienze della comunicazione., 3(1):61–80, 2003.
- [Sn03] C. Snyder. Paper Prototyping - The Fast and Easy Way to Design and Refine User Interfaces. Elsevier Science, 2003.
- [Sw07] SwissEduc. Bildungsserver SwissEduc, Sammlung von Unterrichtsmaterialien, Unterbereich Informatik. <http://www.swisseduc.ch/informatik/>, Januar 2007.
- [Wü06] P. Wülser. Unterricht fürs Ohr - Podcasting in der Schule. Unveröffentlichtes Manuskript, 2006.