

Implementierung von eLearning-Szenarien nach der Theorie der kognitiven Lehre

Ulrik Schroeder

Lehr- und Forschungsgebiet für Informatik IX
Computerunterstütztes Lernen und Wissensstrukturierung
RWTH Aachen
52056 Aachen
schroeder@cs.rwth-aachen.de

Christian Spannagel

Institut für Mathematik und Informatik
Pädagogische Hochschule Ludwigsburg
Reuteallee 46
71634 Ludwigsburg
spannagel_christian@ph-ludwigsburg.de

Abstract: Die Realisierung innovativer, didaktisch und lerntheoretisch begründeter eLearning-Szenarien benötigt dezidierte Werkzeuge, die neuartige Lehr- und Lernformen adäquat unterstützen. In diesem Artikel werden eLearning-Szenarien vorgestellt, die das Modell der kognitiven Lehre, das *Cognitive Apprenticeship Model* [CBN89], implementieren. Hilfreich für ihre Umsetzung ist die Aufzeichnung von Lernverhalten, das somit über das Netz übertragen und entfernt wiedergegeben und analysiert werden kann. Aus den beschriebenen Beispielsituationen werden Anforderungen an ein Werkzeug abgeleitet, das die Umsetzung des theoretischen Modells unter besonderer Betonung des Lernprozesses ermöglicht. Die Implementierung CleverPHL unterstützt neben der kognitiven Lehre auch qualitative und quantitative Analysen von Benutzerverhalten, z.B. um empirische Untersuchungen zur Evaluation des Lernprozesses durchzuführen. Der framework-basierte Ansatz ermöglicht zudem das einfache Einbringen eigener Strukturierungs-, Analyse-, Wiedergabe- und Editierfunktionalität der Lernschrittaufzeichnungen, um anwendungsspezifische, semantische Analysen zu implementieren.

1. Motivation

Virtuelle Veranstaltungen, wie beispielsweise *eLectures*, und im Web bereitgestellte Lernmaterialien sollen es Studierenden in einem gewissen Rahmen ermöglichen, zu jeder Zeit an jedem Ort und mit individuellem Lerntempo und -strategie zu lernen. Es wird allerdings häufig beklagt, dass diesen Online-Angeboten pädagogische und

lerntheoretische Grundkonzepte fehlen, lediglich technische Möglichkeiten ohne Mehrwert für das Lernen genutzt und soziale Folgen nicht ausreichend bedacht werden [Ha01]. Daher wird zunehmend gefordert, Pädagogik und Technik besser aufeinander abzustimmen, um so sinnvolle neue Lehr- und Lernformen zu etablieren.

Um den genannten Forderungen nachzukommen, haben wir entsprechende eLearning-Szenarien entwickelt, die sich aus aktuellen Lerntheorien und didaktischen Modellen ableiten. Das zur Realisierung konzipierte eLearning-Werkzeug CleverPHL überträgt in diesen Szenarien Konzepte des *Cognitive Apprenticeship Models* auf Situationen virtuellen Lehrens und Lernens. Nach der Darstellung einiger didaktischer Anforderungen situierter Lernmodelle konkretisieren wir diese exemplarisch in praktikable eLearning-Szenarien. Daraus leiten wir Anforderungen an ein System ab, das die Übertragung des Modells auf virtuelle Lernszenarien unterstützt. Die Erfüllung dieser Anforderungen durch CleverPHL und Aspekte der Adaption und Übertragbarkeit werden zum Abschluss vorgestellt.

2. Situierete Lernmodelle

Moderat-konstruktivistische Modelle betonen die Aktivität Lernender in möglichst authentischen Situationen. Wesentliche Ziele der Lernmodelle der situierteren Kognition [GM95] sind die Förderung von Problemlösefähigkeiten und die Anwendbarkeit erlernten Wissens; Beispiele sind die *Anchored Instruction* [BR90], die reale Probleme zur Motivation an den Anfang von Lerneinheiten stellt, und das *Cognitive Apprenticeship Model* [CBN89], das Konzepte der Handwerkslehre für kognitive Lernziele überträgt. In der kognitiven Lehre verfolgen die „Lehrlinge“ zunächst die Lösungsschritte des „Meisters“ bei der Bearbeitung eines realistischen Problems. Anschließend versuchen sie bei der Lösung eines einfacheren Problems, die beobachteten Schritte zu übertragen und in dem neuen Kontext anzuwenden. Ihre Lösungsbemühungen werden vom Experten beobachtet, der Rückmeldungen und Hilfestellungen gibt, die sich nicht nur auf das Ergebnis, sondern vor allem auf den Prozess des Lösens beziehen.

In vielen Anwendungen dieses Modells hat sich gezeigt, dass Lernende Schwierigkeiten haben, während der Beobachtung eines Experten dessen interne Lösungsschritte selbständig zu erschließen. Es wird daher vom Experten während der Vorführung gefordert, seine internen Vorgänge zusätzlich zu externalisieren, indem er seine Handlungen kommentiert. Ebenso sind die Lernenden anschließend gefordert, ihre eigenen Lösungsstrategien zu artikulieren und mit denen des Experten zu vergleichen.

Das *Cognitive Apprenticeship Model* kann auch in eLearning-Szenarien zum Einsatz kommen. Hier ist die Implementierung des Modells jedoch ungleich schwieriger als in Unterrichtssituationen, da die Vorführung des Experten und das Beobachten durch die Lernenden meist nicht zur selben Zeit oder am selben Ort stattfinden. Die Handlungen des Experten und dessen Kommentare müssen aufgenommen und gespeichert werden, damit sie zu einer anderen Zeit abgerufen oder an einen anderen Ort übertragen werden können. Ebenso muss die Möglichkeit für Lernende gegeben sein, ihren Lösungsprozess

einem Experten zu übermitteln, der wiederum Rückmeldung zum eingeschickten Lösungsprozess geben möchte. Die Externalisierung von Lösungsstrategien und die Möglichkeit zum Vergleich des eigenen Lösungswegs mit denen des Experten oder anderer Lernender ist in eLearning-Szenarien von besonderer Wichtigkeit, da diese die Reflexion des Lernverhaltens auf einer Meta-Ebene anregen. Diese Reflexion ist zur Bildung neuer Kompetenzen für neue Wege des Lernens von entscheidender Bedeutung [FS02].

3. Cognitive Apprenticeship in eLearning -Szenarien

Die folgenden Szenarien sollen exemplarisch demonstrieren, wie sich das Modell der kognitiven Lehre auf unterschiedliche Situationen im eLearning übertragen lässt. Wir haben diese Szenarien für den Einsatz an Hochschulen in der Informatik entworfen; sie lassen sich aber auch einfach auf andere Fächer und Bildungssituationen übertragen.

Tele-Teaching: Nach der Aufzeichnung einer Vorlesung werden Ausführungen eines Dozenten über einen Server den Teilnehmern der Veranstaltung über das Netz zur Verfügung gestellt. Kommt in der Vorlesung ein Programm (z.B. ein Editor oder Debugger zur Programmierung oder die Durchführung eines Experiments anhand einer interaktiven Simulation) zum Einsatz, so liegt es nahe, auch die mit dem Programm vorgeführten Schritte aufzuzeichnen und zur späteren Wiederholung zur Verfügung zu stellen.

Für die kognitive Lehre sollte der Dozent seine aufgezeichneten Aktionen durch die Externalisierung seiner Problemlöseschritte kommentieren. Dazu gibt es prinzipiell zwei Möglichkeiten: geschriebener oder gesprochener Text. Die erste Variante zwingt den Betrachter allerdings dazu, seine Aufmerksamkeit zwischen dargestelltem Prozess und erklärendem Text aufzuteilen. Dieser *Split-Attention Effect* [MM98] führt zu einer ungünstigen Auslastung des Arbeitsgedächtnisses (*cognitive load*) [CS91]. Nach dem Modalitätsprinzip der Theorie des Multimedia-Lernens von Mayer [Ma01] sind auditive Annotationen zu Animationen vorteilhaft, da die visuell präsentierten Aktionen und die gesprochene Information im Arbeitsgedächtnis getrennt verarbeitet werden. Es ist also plausibel, die Aufzeichnung einer Programmbedienung durch auditive Kommentare anzureichern.

Bei der Betrachtung aufgezeichneter Lösungsschritte sollen Lernende auch bestimmte Sequenzen auswählen und separat betrachten können, um z.B. bei der Erledigung von Teilaufgaben lediglich relevante Passagen der Expertendemonstration hinzuzuziehen. Das Abspielen der Demonstration sollte jederzeit unterbrochen werden können, um die Benutzung des Programms manuell fortführen zu können. Dies erfordert die Ausführung der Aufzeichnung auf dem tatsächlichen Programm und nicht nur eine Simulation des Vorganges oder das Abspielen einer Videoaufzeichnung. Zusätzlich sollen Lernende die Abspielgeschwindigkeit und den Detaillierungsgrad des abgespielten Vorganges gemäß ihres Lerntempos und Vorwissens selbst bestimmen.

Online-Tutorien und Hilfsfunktionen einer Software beschreiben analog zu Szenario 1 die Benutzung eines Programms. Gellevij et al. führen an, dass Screenshots, die zielgerichtet in ein textuelles Manual eingebettet werden, den rein textuellen Handbüchern überlegen sind [Ge02]. Das wichtigste Ziel dieser Integration ist die Entwicklung eines mentalen Modells des Computerprogramms. In ihren Untersuchungen konnte gezeigt werden, dass Benutzer mit Handbüchern, die durch Screenshots angereichert worden sind, mehr lernen, und zwar in kürzerer Zeit. Dies entspricht auch den Erkenntnissen der Theorie der dualen Kodierung [Pa71].

Eine konsequente Weiterentwicklung von Text-Bild-Manuals sind Aufzeichnungen mit auditiven Erklärungen. In solchen zeitabhängigen Manuals können Aufgabenteile, in denen der Leser aktiv werden muss, in die Vorführung integriert werden. Hierbei versetzt die Hilfsfunktion das Programm in einen bestimmten Zustand, von dem aus der Benutzer die Benutzung fortführen soll. Die Lösungsschritte des Benutzers können dann anschließend mit den Schritten einer angebotenen Beispiellösung verglichen werden. Dies ermöglicht dem Benutzer die Reflexion und Optimierung des eigenen Verhaltens.

Interaktive Aufgabenstellungen: In typischen Programmierpraktika an Hochschulen werden Lösungen zu Aufgaben durch Studierende eingereicht und durch Tutoren korrigiert und bewertet. Bei vielen Aufgabenstellungen gerade zu Anfang einer Lehrveranstaltung besteht das Problem, dass zunächst ein Ausführungsrahmen geschaffen werden muss, in den das eigentliche Teilproblem eingebettet werden muss, um es auszuführen. Für diese Fälle können die ersten Schritte und der Rahmen der Lösung in einer Programmierumgebung als interaktive Aufgabenstellung vorgegeben werden. Lernende führen dann lediglich die relevanten Schritte zur Komplettierung der Lösung aus. Lernenden zeichnen wiederum ihre Lösungsschritte auf und reichen diese anstelle des Programms zur Korrektur und Bewertung ein. Handelt es sich bei einer solchen Korrekturinstanz um ein Programm, so muss die Aufzeichnung automatisch ausgewertet werden können. Dieses Vorgehen fokussiert im Sinne des *Cognitive Apprenticeship Models* den Lösungsprozess und nicht nur das Produkt. Das Vorgehen lässt sich auf beliebige Übungsaufgaben, bei denen zur Modellierung oder Implementierung Entwicklungsumgebungen eingesetzt werden, übertragen.

Werden derartige Aufgaben im Rahmen von **Online-Prüfungen** gestellt, so ermöglicht das dem Prüfer, nicht nur die Lösung als solches zu bewerten, sondern auch den Lösungsprozess, der unter Umständen zu einem falschen Ergebnis geführt hat, aber korrekte Anteile enthält. Einfaches Kopieren von Lösungen im Laufe der Prüfung wird damit ebenfalls unterbunden.

Erfolgt die **Betreuung** von ePraktika durch Online-Tutoren, schicken Lernende im Problemfall eine Email mit angehängtem Programmtext. Im Sinne der kognitiven Lehre ist es aber wünschenswert, dass der Tutor nicht nur das Programm mit Fehler betrachtet, sondern das Vorgehen der Lernenden bei der Entstehung des Fehlers beobachtet und gezielt Feedback gibt. Wenn Lernende den gesamten Prozess des Lösungsversuchs bis zum Auftauchen des Problems aufzeichnen und einsenden, kann der Tutor falsche Ansätze identifizieren. Als Beratung kann der Tutor dann z.B. ab der Problemstelle

Schritte zur korrekten Fortsetzung aufzeichnen und zusammen mit sprachlichen Kommentaren an Lernende zurückschicken.

Diese Art der Programmierberatung kann dann in Form von **Live-FAQs** (*frequently asked questions*) als eine Datenbank häufiger Fehler und Schritten zur Problembehebung aufbereitet und über Netz für Studierende angeboten werden. Auch hier liegt der Fokus wieder auf dem Lösungsprozess und der Externalisierung durchzuführender Aktionen.

Es gibt zahlreiche weitere didaktische Anwendungsfelder für das Aufzeichnen von Benutzerverhalten, z.B. das **Testen von Hypothesen über Lernaktionen** oder die generelle Überprüfung der **Benutzbarkeit und Nützlichkeit von Lernsoftware**. Diese analytischen Funktionen sind unabhängig von der Implementierung eines bestimmten didaktischen Modells und sollen hier daher nicht näher betrachtet werden. Sie implizieren allerdings weitere Anforderungen an die im Folgenden vorgestellte Software.

4. Anforderungen an ein Aufzeichnungs- und Wiedergabesystem

Aus den oben beschriebenen Szenarien lassen sich Anforderungen an ein System ableiten, welches die benötigten Funktionalitäten bezüglich der Aufzeichnung, der Wiedergabe, der Bearbeitung und der Analyse von Lernprozessen bietet:

- *Capture & Replay-Funktionalität*: Das System ermöglicht die Aufzeichnung und Wiedergabe von Benutzerverhalten; zur Speicherung und zum Transfer der Daten definiert es eine geeignete Datenstruktur.
- *Ausführende Wiedergabe*: Die Wiedergabe erfolgt auf dem tatsächlichen Programm, ähnlich der Ausführung eines aufgezeichneten Makros. Das Abspielen des aufgezeichneten Verhaltens kann unterbrochen und durch den Anwender fortgeführt werden.
- *Verschiedene Wiedergabemodi*: Die Wiedergabe kann selektiv erfolgen. Abspielgeschwindigkeit und Detaillierungsgrad können vom Betrachter bestimmt werden. Hierbei können Funktionen, die vom Videorekorder her bekannt sind, wie schnelles Spulen, Pause, Einzelbild etc. integriert werden.
- *Editierbarkeit*: Die Aufzeichnung des Benutzerverhaltens kann nachträglich bearbeitet werden; dies betrifft zum einen die Möglichkeit der Definition von Strukturen zur Zusammenfassung und Gruppierung einzelner Aktionen und zum anderen das Einfügen oder Löschen von Benutzer- oder Programmaktionen analog zum Videoschnitt aufgenommener Filmsequenzen.
- *Multimedia-Unterstützung*: Die Aufzeichnung kann multimedial angereichert werden. Dies geschieht beispielsweise durch die Kopplung mit Audio- und Videosequenzen und mit Aufgabentexten.
- *Symbolische und hierarchische Repräsentation*: Die Speicherung der Benutzeraktionen erfolgt in einer symbolischen Repräsentation ähnlich Log-Dateien, so dass automatische Analysemechanismen angewendet können. Außerdem ermöglicht dies dem Betrachter, das Verhalten nicht wiedergeben zu müssen, um bestimmte Aktionen zu entdecken, da er die symbolische

Repräsentation durchsuchen kann. Das aufgezeichnete Verhalten wird hierbei auf unterschiedlichen Auflösungsstufen repräsentiert.

- *Erweiterbarkeit*: Verfahren zum Einbringen eigener Verhaltensbeschreibungen und von Erkennungsalgorithmen für spezielle Verhaltenskategorien sind notwendig, um Aktionsfolgen mit Kenntnis der aufgezeichneten Software semantisch zu strukturieren.
- *Analysierbarkeit*: Die Aufzeichnung muss sowohl qualitativ als auch quantitativ orientierte Analysemethoden erlauben.

5. CleverPHL

CleverPHL wurde in Hinblick auf die beschriebenen Anforderungen aus dem Framework Jacareto in Zusammenarbeit der Technischen Universität Darmstadt und der Pädagogischen Hochschule Ludwigsburg [Sp03] entwickelt. Das Programm, dessen Einsatzgebiet ursprünglich zum automatisierten Testen eines Software-Produkts liegt, ermöglicht die Aufzeichnung und Wiedergabe von Benutzeraktionen auf Java-Zielanwendungen (*Capture & Replay-Funktionalität*). CleverPHL zeichnet aber nicht nur Benutzer-, sondern auch Programmaktionen, wie das Öffnen eines neuen Fensters oder die Änderung angezeigter Daten, auf. Sowohl Benutzer- als auch Programmaktionen werden in der Reihenfolge ihres zeitlichen Auftretens in einer linearen Datenstruktur, der *Interaktionsaufzeichnung*, in einem speziellen XML-Format gespeichert (*symbolische Repräsentation*). Sie können an andere Anwendungen z.B. zur statistischen Auswertung exportiert werden. Weiterhin können neben Interaktionen weitere Elemente, z.B. Annotationen (*Multimedia-Unterstützung*), Pausen und weitere Kontrollelemente eingefügt werden. Für jedes Aufzeichnungselement existiert ein Editor, mit dessen Hilfe die Elementattribute bearbeitet werden können (*Editierbarkeit*).

Die Wiedergabe einer Interaktionsaufzeichnung erfolgt auf einer neuen Instanz der Zielanwendung. Die Wiedergabeschritte werden demnach nicht simuliert, sondern tatsächlich auf der Zielanwendung ausgeführt (*ausführende Wiedergabe*). Dies hat den Vorteil, dass die Wiedergabe an beliebigen Stellen angehalten und mit interaktiver Bedienung des Programms fortgefahren werden kann. Die Wiedergabe kann in Echtzeit oder mit frei wählbaren Zeitabständen zwischen Ereignissen erfolgen. Im letzten Fall können die Stellen, an denen die nächste Aktion erfolgen wird, farblich markiert werden, um die Aufmerksamkeit des Betrachters zu lenken (Abb. 1). Weiterhin ist die Wiedergabe einzelner Abschnitte der Interaktionsaufzeichnung möglich. Hierzu wird zunächst bis zum wiederzugebenden Abschnitt im Schnelldurchlauf vorgespult und anschließend selektiv wiedergegeben (*Verschiedene Wiedergabemodi*).

Interaktionsaufzeichnungen haben in ihrem Rohzustand eine sehr hohe Auflösung: jede einzelne Mausbewegung und jeder einzelne Tastendruck werden gespeichert. In dieser Form sind die Aufzeichnungen sehr unübersichtlich. Daher fasst CleverPHL elementare Sequenzen von Aufzeichnungselementen automatisch zu größeren Strukturen zusammen (Abb. 2 - *hierarchische Repräsentation*). Diese Strukturierung erlaubt es, das Verhalten zu inspizieren, ohne es wiederzugeben. Hierzu hat man zunächst einen groben Blick auf

das aufgezeichnete Verhalten, kann sich aber Details einzelner Verhaltenssequenzen durch Öffnen des entsprechenden Baumknotens näher betrachten.

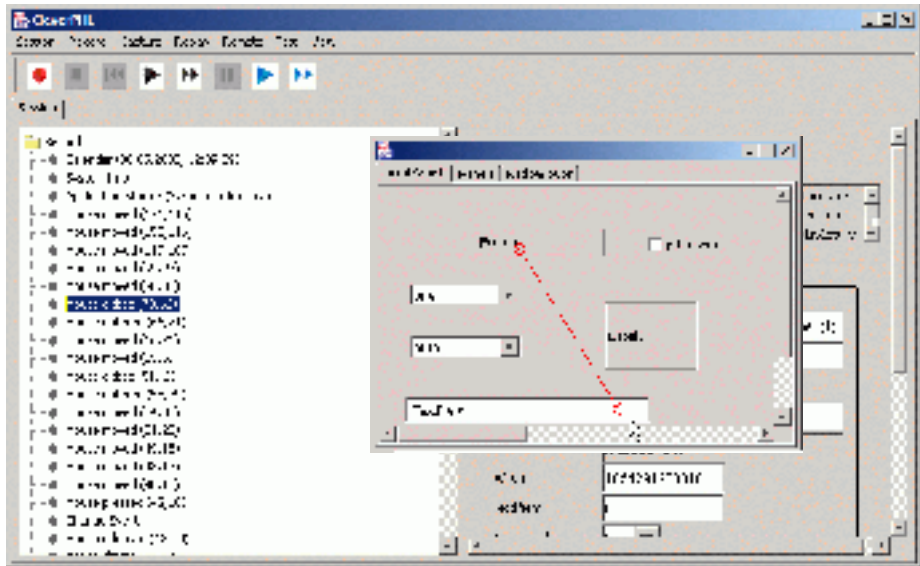


Abbildung 1: CleverPHL (im Hintergrund) und die Zielanwendung (im Vordergrund) während der Wiedergabe mit optischer Hervorhebung der Aktionsbereiche

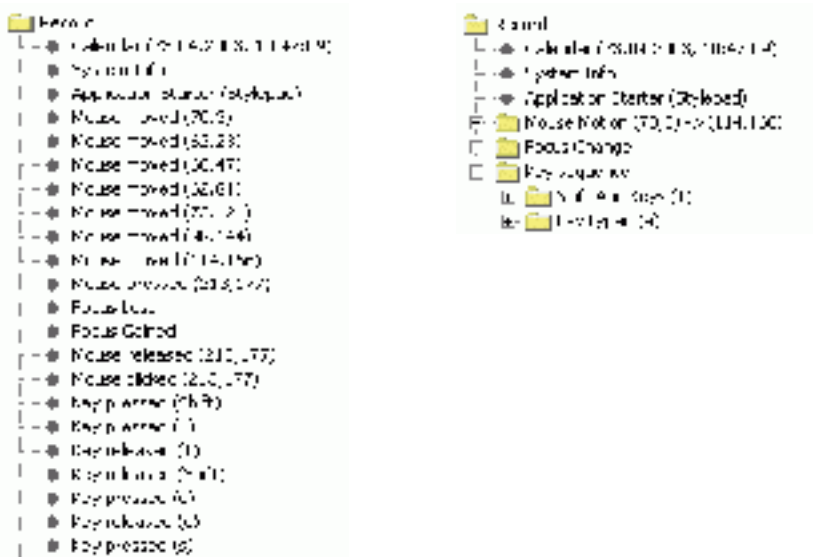


Abbildung 2: Interaktionsaufzeichnung, unstrukturiert (links) und strukturiert (rechts)

Für jedes vordefinierte *Strukturelement* (Knoten des Strukturbaums) gibt es einen Erkennungsalgorithmus. Eigene Erkennungsalgorithmen können eingebracht werden, um automatisch Verhaltenskategorien zu entdecken und zu Strukturen zusammenzufassen (*Erweiterbarkeit*). Dies ermöglicht eine automatische, qualitative Analyse der Aufzeichnung. Quantitative Analysen können durchgeführt werden, in dem die Attribute gefundener Strukturelemente in einem Datensatz zusammengestellt und in Statistikprogramme übertragen werden (*Analysierbarkeit*). Eine *semantische Strukturierung* von Interaktionssequenzen ist nur mit Kenntnis der Anwendung möglich. So kann beispielsweise die Bedienung eines Druckdialogs aus den Elementen *Format gewählt*, *Seitenzahl gewählt* und *Druck gestartet* bestehen, die selbst aus einzelnen Mausbewegungen und Tastatureingaben zusammengesetzt sind. Auch diese Strukturen können durch Definition entsprechender Analysealgorithmen in das Framework eingehängt werden.

6. Technische Aspekte

CleverPHL basiert auf dem Jacareto-Framework mit Hot Spots für alle relevanten Kernfunktionen zur Aufzeichnung, Strukturierung, Wiedergabemodi und Editoren. Beispielsweise sind Wiedergabemodule durch Subklassen der Klasse `Replayer` definiert. Jedes dieser Module behandelt während der Wiedergabe eine Kategorie von Wiedergabeelementen. Wiedergabemodule können mit einer bestimmten Priorität bei der Klasse `Replay`, die die Wiedergabe steuert, registriert werden. Während der Wiedergabe werden nun alle registrierten Module gemäß ihrer Priorität gefragt, ob sie für das aktuelle Strukturelement zuständig sind. Das so ermittelte Wiedergabemodul darf dann die Wiedergabe des Strukturelements ausführen (*Chain of Responsibility Pattern*).

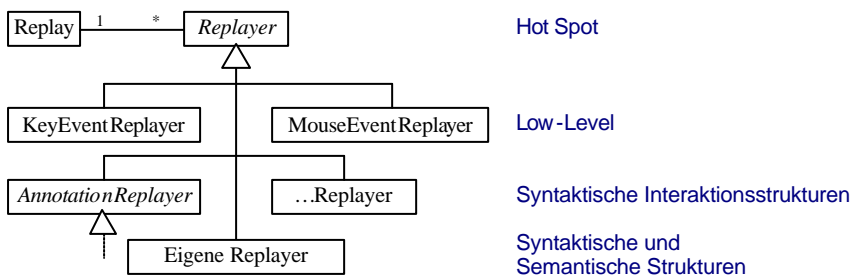


Abbildung 3: Hot Spot zum Einbringen eigener Wiedergabeklassen

Die Interaktionsstrukturierung erfolgt in einem adaptierten Top-Down-Strukturanalyseverfahren aus dem Compilerbau [ASU85] auf der Basis einer formalen Strukturbeschreibung, der *Aufzeichnungsgrammatik*. Die Analyse wird von der Klasse `RecordTokenizer` organisiert. Klassen, die Strukturelemente repräsentieren, besitzen eine Methode, die den Erkennungsalgorithmus beinhaltet. Das Einbringen eigener Algorithmen erfolgt über die Implementierung solcher Klassen.

7. Zusammenfassung und Ausblick

CleverPHL setzt die Anforderungen aus Abschnitt 4 weitgehend um. Somit bietet es die Möglichkeit, das Konzept der kognitiven Lehre in eigenen eLearning-Szenarien umzusetzen. Wählt man für seine eigene Veranstaltung ein didaktisches (blended eLearning) Konzept, welches durch die Aufzeichnung und die Wiedergabe von Lernschritten adäquat unterstützt wird, so kann CleverPHL die Brückenfunktion zwischen Zeit und Raum übernehmen. Die in Abschnitt 5 beschriebenen Darstellungs- und Bearbeitungsfunktionen bieten hierbei den benötigten flexiblen Zugriff auf die Aufzeichnung. Sowohl Lehrende als auch Lernende werden angeregt, ihre eigenen Lösungsstrategien zu reflektieren und die Strategien anderer nachzuvollziehen. Die Funktionen zur strukturierten Analyse bilden zusätzlich ein geeignetes Mittel zur Untersuchung von Lernprozessen.

Zukünftige Erweiterungen betreffen im Wesentlichen die Multimedia-Funktionalität. Momentan ist lediglich die rudimentäre Unterstützung von Audiokommentaren gegeben. Hierbei können in die strukturierte Interaktionsaufzeichnung Audiodateien integriert werden, die während der Wiedergabe der Aufzeichnung abgespielt werden. Die Kopplung mit Videos und anderen Medien wird aktuell bearbeitet. Hierbei werden vor allem Kriterien wie Synchronizität, Speicherbedarf und Streamingfähigkeit beachtet.

Ein erster Einsatz in Lehrveranstaltungen zur Programmierung ist zum kommenden Wintersemester an der RWTH Aachen geplant. Ebenso wird CleverPHL in dem Forschungskolleg *Fachintegratives Lernen mit digitalen Medien* der PH Ludwigsburg eingesetzt, das multimediale Lernsoftware für den fächerübergreifenden Unterricht an Realschulen entwickelt und evaluiert. Hier werden u.a. Aufzeichnungssequenzen mit auditiven Kommentaren erstellt, die in die Benutzung eines Lernmoduls einführen.

Literaturverzeichnis

- [ASU85] Aho, A., Sethi, R. & Ullman, J.: Compilers. Principles, Techniques and Tools. Addison-Wesley, 1985.
- [Br90] Bransford, J. D., Sherwood, R. D., Hasselbring, T. S., Kinzer, C. K. & Williams, S. M.: Anchored Instruction. Why we need it and how technology can help. In (D. Nix & R. Spiro, Hrsg.): Cognition, education, and multimedia: Exploring ideas in high technology. Lawrence Erlbaum Associates, 1990; S. 115-141.
- [CBN89] Collins, A., Brown, J. S. & Newman, S. E.: Cognitive Apprenticeship. Teaching the Crafts of Reading, Writing, and Mathematics. In (L. B. Resnick, Hrsg.): Knowing, learning, and instruction. Lawrence Erlbaum Associates, 1989. S. 453-494
- [CS91] Chandler, P. & Sweller, J.: Cognitive Load Theory and the Format of Instruction. Cognition and Instruction 8(4), 1991. S. 293-332.

- [FS02] Farinetti, L. & Schroeder, U.: Self-directed Distance Learning Systems. In: Proceedings of Int. Conference on Open and Distance Learning, European Distance Learning Network, EDEN 2002.
- [Ge02] Gellevij, M., van der Meij, H., de Jong, T. & Pieters, J.: Visuals in Instruction. Functions of Screen Captures in Software Manuals, 2002.
<http://www.iwm-kmrc.de/workshops/visualization/gellevij.pdf> (Abruf am 17.5.2003)
- [GM95] Gerstenmaier, J. & Mandl, H.: Wissenserwerb unter konstruktivistischer Perspektive. Zeitschrift für Pädagogik, 41, 1995. S. 867-888.
- [Ha01] Harper, B., O'Donoghue, J., Oliver, R. & Lockyer, L.: New Designs for Web Based Learning Environments. In (C. Montgomerie & J. Viteli, Hrsg.): Proceedings of ED-MEDIA 2001, World Conference on Educational Multimedia, Hypermedia & Telecommunications. Tampere, Finland: Association for the Advancement of Computing in Education, 2001. S. 674-675.
- [Ma01] Mayer, R.: Multimedia-Learning. Cambridge University Press, New York, 2001.
- [MM98] Mayer, R. & Moreno, R.: A Split-Attention Effect in Multimedia Learning. Evidence for Dual Processing Systems in Working Memory. Journal of Educational Psychology, Vol. 90, No. 2, 2001. S. 312-320.
- [Pa71] Paivio, A.: Imagery and Verbal Processes. Holt, Rinehart & Winston, New York, 1971.
- [Sc02] Schroeder, U.: Meta-Learning Functionality in eLearning Systems, International Conference on Advances in Infrastructure for Electronic Business, Education, Science, and Medicine on the Internet 2002, L'Aquila, Italy, 2002.
- [Sp03] Spannagel, C.: Jacareto Website. Download, Dokumentation und weitere Informationen. <http://www.ph-ludwigsburg.de/mathematik/personal/spannagel/jacareto/> [April 2003].