

Adaptive Rückmeldungen im intelligenten Tutorensystem LARGO

Niels Pinkwart¹, Vincent Aleven², Kevin Ashley³, Collin Lynch⁴
Autor für Korrespondenz, E-Mail: niels.pinkwart@tu-clausthal.de

¹ Technische Universität Clausthal
Institut für Informatik

² Carnegie Mellon University
Human Computer Interaction Institute

³ School of Law und Learning Research & Development Center
University of Pittsburgh

⁴ Intelligent Systems Program
University of Pittsburgh

urn:nbn:de:0009-5-16085

Zusammenfassung

Das intelligente Tutorensystem LARGO für die Rechtswissenschaften soll Jurastudenten helfen, Argumentationsstrategien zu lernen. Im verwendeten Ansatz werden Gerichtsprotokolle als Lernmaterialien verwendet: Studenten annotieren diese und erstellen graphische Repräsentationen des Argumentationsverlaufs. Das System kann dabei zur Reflexion über die von Anwälten vorgebrachten Argumente anregen und Lernende auf mögliche Schwächen in ihrer Analyse des Disputs hinweisen. Zur Erkennung von Schwächen verwendet das System Graphgrammatiken und kollaborative Filtermechanismen. Dieser Artikel stellt dar, wie in LARGO auf Basis der Bestimmung eines „Benutzungskontextes“ die Rückmeldungen im System benutzungsadaptiv gestaltet werden. Weiterhin diskutieren wir auf Basis der Ergebnisse einer kontrollierten Studie mit dem System, welche mit Jurastudierenden an der University of Pittsburgh stattfand, in wie weit der automatisch bestimmte Benutzungskontext zur Vorhersage von Lernerfolgen bei Studenten verwendbar ist.

Stichwörter: e-learning; tutoring systems, learning management system, intelligente Tutorensysteme, juristische Argumentation

Abstract

The Intelligent Tutoring System LARGO is designed to help law students learn argumentation skills. The approach implemented in LARGO uses transcripts of oral arguments as learning resources: Students annotate them and create graphical representations of the argument flow. The system encourages students to reflect upon arguments proposed by the attorneys and helps students detect possible weaknesses in their analysis of the dispute. Technically, graph grammar and collaborative filtering algorithms are employed to detect these weaknesses. This article describes how “usage contexts” are determined and used to create adaptive feedback in LARGO. On the basis of a controlled study with the system that took place with law students at the University of Pittsburgh, we discuss to what extent the automatically calculated usage contexts can predict student’s learning gains.

Keywords: e-learning; tutoring systems, learning management system

1. Einleitung

E-Learning-Systeme haben im Bereich der Rechtswissenschaften eine lange Tradition und existieren für ganze eine Reihe juristischer Disziplinen [MB01, Ro92, Ha07]. In der juristischen Praxis ist Argumentation von zentraler Bedeutung: die Fähigkeit, vor Gericht überzeugende Argumente zu formulieren und diese angemessen im Kontext von Gegenargumenten zu verteidigen, kann oft über Erfolg oder Misserfolg entscheiden und ist daher für Anwälte essentiell. Daher ist juristische Rhetorik und Argumentation Gegenstand vieler Übungsveranstaltungen in Jura-Studiengängen. Es ist in diesem Zusammenhang überraschend, dass trotz der Vielfalt von e-Learning-Systemen für verschiedene juristische Bereiche nur wenige Anwendungen wie CATO [Al03] und ArguMed [Ve03] existieren, die Lernende speziell in der Argumentation unterstützen. CATO operiert dabei mit Techniken des fallbasierten Schließens auf Basis existierender Urteile mit deren Faktoren, ArguMed zielt auf strukturelle Aspekte in Argumenten ab und hilft Lernenden dabei, visuelle Repräsentationen anfechtbarer Aussagen zu erstellen. Teilweise kann die geringe Anzahl von e-Learning-Systemen für juristische Argumentation dadurch erklärt werden, dass die zugrunde liegende Domäne schlecht strukturiert ist [Ly06]. Im Gegensatz zu gut strukturierten Problem-domänen wie z.B. Mathematik gibt es im Bereich der juristischen Argumentation oft keine eindeutig definierbare „korrekte Lösung einer Aufgabe“, die als Modell in einem Tutorensystem verwendet werden könnte: obwohl Argumente typischerweise rational auf Basis von Gesetzestexten oder vorangegangenen Urteilen in anderen ähnlichen Fällen begründet werden, ist der Argumentationsprozess ein sprachlicher Diskurs, in dessen Verlauf juristische Prinzipien und Gesetzesartikel im Angesicht spezifischer Faktenlagen interpretiert werden müssen. Die fehlende Lösungseindeutigkeit wird in zahlreichen Urteilsrevisionen in höheren Instanzen wie auch in der Schwierigkeit, Verfahrensausgänge zu prognostizieren, deutlich.

Dieser Artikel beschreibt das intelligente Tutorensystem LARGO, das Studenten beim Lernen juristischer Argumentationsstrategien unterstützt. Hierbei nutzen die Studenten Protokolle von mündlichen Gerichtsverfahren und beschreiben die Positionen und Beiträge der Akteure in Diagrammform. Das System kann dabei zur Reflexion der durch die Anwälte vorgebrachten Argumente anregen und die Lernenden auf mögliche strukturelle wie inhaltliche Schwächen in ihrer Analyse des Disputs hinweisen.

Die Systemrückmeldungen erfolgen in Form von Aufforderungen zur Selbsterklärung [Ch00]. Im Schwerpunkt beschreibt dieser Artikel die Prioritätsbestimmung für Rückmeldungen, welche den aktuellen Kontext der Systemnutzung durch den Lernenden über mehrere charakteristische Parameter ermittelt. Auf Basis dieses Benutzungskontextes gestaltet LARGO die Rückmeldungen an die Nutzer adaptiv. Der Artikel thematisiert weiterhin, in wie weit die Verfahren, die LARGO zur Bestimmung des Benutzungskontextes verwendet, auch zur Vorhersage von Lernerfolgen bei den Personen, die diese Diagramme erstellt haben, verwendbar sind.

2. Diagramme zur Argumentvisualisierung

Im angloamerikanischen „Common Law“-System formulieren die Anwälte beider Seiten in Gerichtsverhandlungen üblicherweise Vorschläge zur Interpretation der Gesetzestexte, auf Grund derer der behandelte Fall zu ihren Gunsten entschieden werden sollte. Diese Vorschläge haben oft die strukturelle Form einer auch auf vergleichbare Faktenlagen anwendbaren allgemein gehaltenen Entscheidungsregel (Test). Um abzuwägen, welcher Argumentationslinie das Gericht in seiner Urteilsprechung folgt, konstruieren Richter oft hypothetische Sachverhalte mit Herausforderungscharakter und befragen die Anwälte, wie diese hypothetischen Sachverhalte unter der vorgeschlagenen Entscheidungsregel zu bewerten seien. Die Anwälte rechtfertigen typischerweise ihre Positionen, z.B. durch Veränderung ihres Tests oder Unterscheidung der Faktenlage von der Hypothese [As07]. Diese Dispute vor Gericht sind die fundamentale Basis für das gesamte „Common Law“-System und illustrieren hervorragend die wichtigen Prozesse der juristischen Argumentation und Begriffsbildung [As90, As07].

Eine umfassende Studie aus dem Jahr 1997, bei der die Entwicklung der Rechtsprechung in neun Ländern (darunter Deutschland) systematisch verglichen wurde, kam zu dem Schluss, dass in allen betrachteten Ländern – auch in kontinentaleuropäischen Systemen mit römischer Rechtsprechungstradition – Argumentationen der beschriebenen Art, welche auf dem Vergleich von Faktenlagen mit Präzedenzfällen oder hypothetischen Sachverhalten beruhen, zu beobachten sind [MS97, S. 535]. Ein Grund hierfür ist es, dass Gerichte in europäischen Ländern nicht (nur) Entscheidungen der EU-Gerichtshöfe, sondern zunehmend auch solche aus anderen EU-Mitgliedsstaaten berücksichtigen. Ein zweiter Grund ist die heute einfache digitale Verfügbarkeit von Urteilen [Lu98, S. 223-224]

In Europa spielt fallbasierte juristische Argumentation vor allem auf Verfassungsebene sowie in den EU-Gerichtshöfen eine sehr wichtige Rolle. Sie ist jedoch aus den genannten Gründen auch in untergeordneten Instanzen mittlerweile durchaus beobachtbar, wenn auch – z.B. bedingt durch Unterschiede in der Bedeutung einzelner Entscheidungen für Folgefälle und der Art der Urteilsbegründungen – in anderer Form als im „Common Law“-System [MS97, S. 536-539].

Aus diesem Grund ist es für Jurastudenten auch in Deutschland sinnvoll, die grundlegenden Prinzipien der fallbasierten juristischen Argumentation mit hypothetischen Sachverhalten, Präzedenzfällen und Entscheidungsregeln zu verstehen. Notwendig wird dies, um Rechtsprechung und unterschiedliche Argumentationsprinzipien im internationalen Kontext wie z.B. dem Handelsrecht (z.B. CISG: UN Convention on International Sales of Goods) sinnvoll nachvollziehen zu können [Cu01, S. 67-68].

Protokolle von gerichtlichen Argumentationen stellen dabei eine wertvolle Lernressource für Jurastudenten dar, welche jedoch auf Grund der enormen Komplexität der Argumente schwierig zu nutzen ist. Eine Möglichkeit, mit dieser Komplexität umzugehen, besteht darin, Diagramme zur Argumentrepräsentation einzusetzen.

Forschungsergebnisse zeigen, dass integrierte Text/Graphik-Repräsentationen eine Lernhilfe darstellen können [Ai99] und die Fähigkeit zum kritischen Denken schulen [Ge02]. Auch speziell im juristischen Bereich existieren einige e-Learning-Systeme, die verschiedene Arten von Diagrammen zur Darstellung von Argumenten verwenden. Carr

[Ca03] verwendet hierzu Toulmin-Schemata [To58] – Argumentationsdiagramme, die aus den elementaren Bausteinen „Behauptung“, „Fakt“, „Zusicherung“, „Unterstützung“, „Widerlegung“ und „Kennzeichner“ bestehen. Im System Araucaria [RR04] kommen Voraussetzungs/Schlussfolgerungsketten zum Einsatz. ArguMed [Ve03] beinhaltet zusätzlich zu visuellen Repräsentationen einen „Assistenten“, der Diagrammstrukturen analysiert und den Lernenden auf dieser Basis Hilfe anbietet.

Auch in Deutschland gibt es einige eLearning-Anwendungen, die zum Training juristischer Argumentation dienen können. TakeLaw [1], dessen Schwerpunkt eher auf Fallanalyse als auf Disput liegt, erkennt strukturelle Fehler in studentischen Lösungsdiagrammen und bietet multimedial unterstützte Korrektur von Aufgaben, jedoch keine freie Texteingabemöglichkeiten an. Dies ist einerseits ein Vorteil, da gesicherte Aussagen über Korrektheit von Diagrammen möglich werden, andererseits aber hinsichtlich der Ausbildung von Argumentationsfähigkeiten nicht unproblematisch. Der in das Normfall-System eingebettete Normfall-Lerntrainer [2] [Ha07] ermöglicht es, komplexe juristische Strukturen an Hand von konkreten Fällen einzuüben und anzuwenden. Der an der Universität Hamburg entwickelte „Trainer Zivilrecht“ [3] bietet eine Hypertext-Ansicht des examensrelevanten Zivilrechts mit Übungsfällen. Dieses durch die Lernenden erweiterbare System ermöglicht das Gutachtentraining und beinhaltet eine Vielzahl von Gutachtenskizzen und Urteilen. Die „KnowledgeTools“-Methode [4] ist kein juraspezifisches eLearning-System im engeren Sinn, kann aber dafür verwendet werden, das Verständnis für Zusammenhänge von Regeln in den Rechtswissenschaften zu fördern: die Methode basiert auf einer strukturierten Visualisierung von Sachzusammenhängen über logische Operatoren (und/oder/entwederoder) und hierarchische Verknüpfungen. Ebenfalls nicht primär auf eLearning-Anwendungen für juristische Argumentation zugeschnitten – aber durchaus dafür einsetzbar – sind Forschungsarbeiten im Bereich der Entscheidungsunterstützung. So beschreibt z.B. Phillips [Ph95], wie Fuzzy-Logik als Hilfe bei der Gewichtung von Beurteilungskriterien und bei der Interpretation von Gesetzestexten eingesetzt werden kann.

Zu keinem dieser genannten Systeme existiert jedoch eine empirische Studie zur Belegung der lernunterstützenden Wirkung [Br06]. Die genannten Ansätze und Systeme aus Deutschland bieten keine Funktionen, um die „common law“-typische Argumentation an Hand von Fallvergleichen zu trainieren. Auch bietet kein existierendes System zur Argumentationsunterstützung eine Hilfefunktion an, die über rein strukturelle Faktoren in Diagrammen hinausgeht. So bleibt der unzweifelhaft sehr wichtige vom Lernenden eingetragene Inhalt in Diagrammelemente typischerweise (auf Grund von fehlenden geeigneten Verfahren) uninterpretiert.

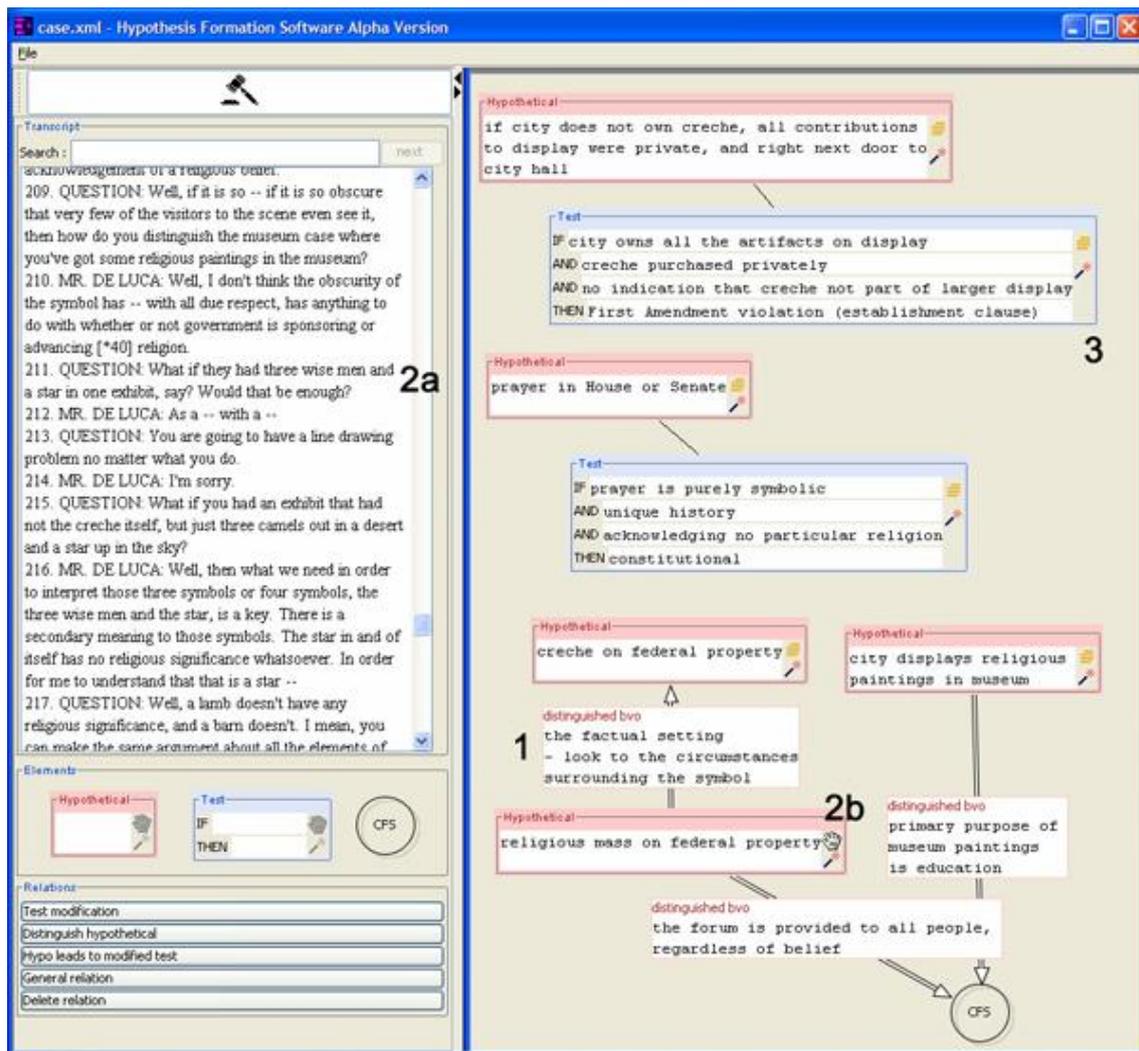


Abbildung 1: Beispiel eines Argumentationsdiagramms in LARGO

Wie in Abschnitt 3 und [Pi06a, Pi06b] beschrieben, versuchen wir dies im hier vorgeschlagenen Ansatz mit Hilfe von kollaborativen Filteralgorithmen – Verfahren, welche darauf basieren, dass Lernende die Diagramme anderer Lernender kommentieren und bewerten [KR02] – zu ändern. Unser Ansatz verwendet „Tests“, „Fakten“ und „Hypothesen“ als primäre ontologische Kategorien zur Analyse von juristischen Argumenten, und konsequenterweise auch als Elementtypen in Diagrammen. Ähnlich wie das Araucaria-System [RR04] erlaubt unser Ansatz, ein Textdokument (Verfahrensprotokoll) mittels eines Diagramms zu annotieren. Hierbei kann jedes Element des Diagramms zu einem speziellen Ausschnitt des Textes in Beziehung gesetzt werden.

Abbildung 1 zeigt das von einem Studenten im Rahmen von Pilotversuchen mit dem System erstellte Diagramm, welches die Argumentation des Verteidigers (Herr de Luca) im vor dem U.S. Supreme Court verhandelten Fall „Lynch vs. Donnelly“, 465 U.S. 668 (1984), beschreibt. In diesem Fall wurde verhandelt, ob das Aufstellen einer Weihnachtsausstellung mit Krippe und Jesuskind durch eine Stadt mit dem Prinzip der Trennung von Staat und Kirche, welches ein wichtiger Bestandteil der US-Verfassung ist, vereinbar ist. Der Anwalt de Luca argumentiert gegen die Verfassungsmäßigkeit; eine

seiner vorgeschlagenen Entscheidungsregeln ist in Element (3) in Abbildung 1 zu sehen: wenn die Stadt die Ausstellung (bis auf die Krippe) besitzt und die Krippe privater Eigentum ist, dies aber für die Besucher nicht ersichtlich ist, dann sei die Verfassung verletzt. Die Abbildung zeigt weiterhin einen anderen Test sowie fünf hypothetische Sachverhalte, die im Verfahren diskutiert wurden.

Die linke Seite des Fensters enthält den aktuell sichtbaren Teil des Protokolls und eine Palette mit den Elementen (Test, Hypothetische Situation, Fakten) und Relationen (Modifikation, Unterscheidung, Analogiebildung, kausale Beziehung, allgemeine Beziehung), welche zwischen den Elementen herstellbar sind. Die Erstellung von Diagrammen erfolgt in LARGO über einfache Drag&Drop-Operationen von der Palette in den Arbeitsbereich. Diagrammelemente können über Hyperlinks mit den Stellen des Protokolls, welche sie beschreiben, verbunden werden. Dieser Prozess ist in Video 1 [5] illustriert.

3. Schwachstellenermittlung in Diagrammen

Traditionelle intelligente Tutorensysteme basieren darauf, dass die von den Lernenden ins System eingegebenen Aufgabenlösungen durch das System analysiert werden, dann ggf. mit dem Lernermodell verglichen werden, und schließlich zu einer Systemrückmeldung führen, welche üblicherweise Lernende auf Fehler in deren Aufgabenbearbeitung hinweist. Die Lokalisierung von „Fehlern“ in Argumentdiagrammen ist aus zwei Gründen problematisch: eine eindeutige „beste Lösung“ existiert oft nicht, und die Interpretation des Textinhalts der Diagramme ist mit heutigen sprachverarbeitenden Verfahren nicht sinnvoll: diese wären in der Domäne der juristischen Argumentation, in der bereits Formulierungsnuancen große Unterschiede machen können, sehr fehleranfällig. Es ist nicht zu erwarten, dass in absehbarer Zeit ein Computersystem in der Lage sein wird, selbständig automatisch zu bestimmen, ob zwei gegebene sprachliche Formulierungen die gleiche juristische Bedeutung haben.

Unser Verfahren beruht auf der Ermittlung von Schwächen (im Sinne einer Heuristik für potenzielle Schwachstellen in Argumenten) in Diagrammen. Hierbei werden strukturelle, kontextbezogene und inhaltliche Schwächen unterschieden. Unser Ansatz zur Ermittlung dieser drei Typen von Schwächen wird im Folgenden kurz beschrieben und ist in [Pi06a, Pi06b] technisch detaillierter dargestellt. Strukturelle Schwächen sind allein auf Grund der im Diagramm auftretenden Typen (Elemente und Relationen) definiert. So werden in Abbildung 1 etwa zwei Hypothesen voneinander unterschieden (1). Dies kann durchaus im Verfahren geschehen, typischerweise sollten Hypothesen jedoch auch in Beziehung zu den Fakten des Falls gesetzt werden. Dies ist im Diagramm für die Hypothese „creche on federal property“ nicht erkennbar. Eine weitere strukturelle Schwäche des Diagramms besteht darin, dass drei der Hypothesen mit keinem Test in Verbindung stehen. Da Hypothesen zum Ziel haben, Tests zu überprüfen (auf Sinngehalt und Konsistenz mit den Artikeln des Gesetzes), stellt das Fehlen solcher Verbindungen eine Schwäche dar.

Kontextbezogene Schwächen beziehen sich auf die vom Lernenden erzeugten Verbindungen zwischen den Diagrammelementen und dem Verhandlungsprotokoll. Sind etwa wichtige Teile des Protokolls gar nicht oder mit einem unangemessenen Diagrammelement markiert worden (z.B. Auffassung einer Hypothese als Faktum), so stellt dies eine kontextbezogene Schwäche dar.

In Abbildung 1 wurde z.B. die im Argumentationsverlauf wichtige Hypothese (2a) nicht im Diagramm berücksichtigt, weiterhin ist das Element unter (2b) nicht mit dem Protokoll verlinkt. Kontextbezogene und strukturelle Schwächen werden im System mit Hilfe einer auf dem Diagramm operierenden Graphgrammatik [RS97] erkannt. Diese Grammatik besteht aus 44 Regeln zur Definition von konkreten Schwächenarten sowie einem Verfahren, welches diese Schwächen in einem beliebigen Diagramm diagnostizieren kann. Regeln zu strukturellen Schwächen sind dabei unabhängig vom konkret verwendeten Gerichtsprotokoll, während kontextbezogene Schwächen nur durch Regeln mit protokollabhängigen (d.h. fallspezifischen) Parametern erkannt werden können. Durch die Grammatikregeln kann eine Ähnlichkeit von Diagrammen mit einer partiell definierbaren „Expertenlösung“ überprüft werden. Dies ist sinnvoll, da die Korrektheit von Argumentationsdiagrammen zwar nicht eindeutig definierbar ist, lokale Lösungsbedingungen (z.B. eine Markierung der wichtigsten Passagen im Protokoll durch die Lerner) jedoch durchaus angegeben werden können.

Auch wenn ein Argumentationsdiagramm keine strukturellen Schwächen aufweist und alle wichtigen Textpassagen sinnvoll mit Diagrammelementen verlinkt sind, können Verständnisschwierigkeiten beim Lernenden nicht ausgeschlossen werden – es ist z.B. möglich, dass ein Student die Hauptaussage eines vorgebrachten Arguments nicht verstanden hat und demzufolge keine qualitativ gute Beschreibung im entsprechenden Diagrammelement enthalten ist. Diese inhaltlichen Schwächen sind schwierig zu diagnostizieren. Selbst für einen menschlichen Tutor ist es nicht einfach, zu entscheiden, ob das Element (3) in Abbildung 1 eine adäquate Wiedergabe des durch den Anwalt im Verfahren vorgebrachten Tests ist – für ein Computersystem ist die Entscheidung nicht leichter. In unserem Ansatz werden inhaltliche Schwächen durch einen kollaborativen Filteralgorithmus [KR02] erkannt. Hierbei wird ausgenutzt, dass durch die strukturellen und kontextbezogenen Regeln systemseitig bekannt ist, welches Diagrammelement welchen Teil des Protokolls beschreibt. Ist eine wichtige Passage des Protokolls vom Lernenden annotiert worden, bietet das System eine Auswahl von Beschreibungsalternativen der gleichen Textpassage an, die durch andere Studenten erstellt wurden (für die ersten Nutzer, bei denen noch keine Lösungen anderer vorliegen, werden Musterlösungen variierender Qualität verwendet). Abbildung 2 illustriert dieses Prinzip. Der Lernende hat nun die Aufgabe, alle Alternativen zu bestimmen, die er für gut hält.

Da dem System eine heuristische Bewertung der anderen Alternativen vorliegt, ist aus der Auswahl des Lernenden annäherungsweise bestimmbar, ob dieser das Argument in der entsprechenden Textpassage verstanden hat oder nicht. Hinzu kommt, dass ab diesem Zeitpunkt auch die eigene Lösung anderen Systemnutzern zur Bewertung vorgelegt wird. Diese entscheiden durch ihr Auswahlverhalten (Empfehlung/keine Empfehlung), wie sich die systeminterne Heuristik zum Verständnis der Textpassage durch den Lernenden im Laufe der Zeit verändert. Bei einer insgesamt geringen Qualitätseinschätzung wird eine inhaltliche Schwäche im System diagnostiziert. Details des in LARGO verwendeten Algorithmus sind in [Pi06b] dargestellt.

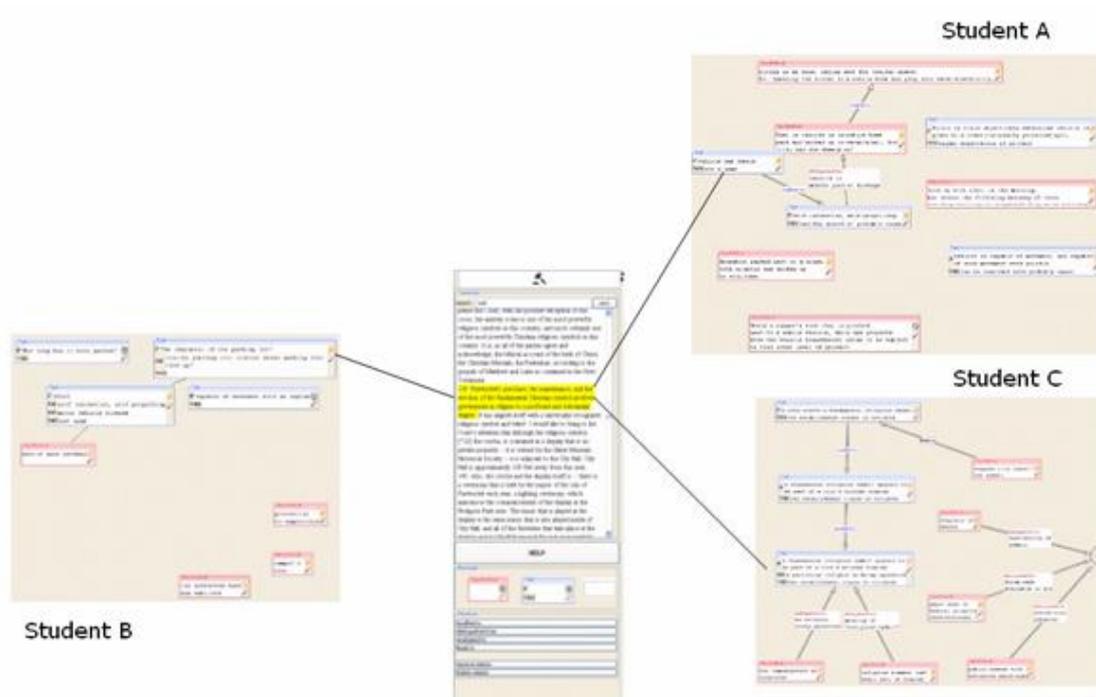


Abbildung 2: Grundlage des kollaborativen Filterns in LARGO – das System erkennt, welche Elemente in verschiedenen Diagrammen den gleichen Textausschnitt beschreiben

4. Rückmeldungen zu Benutzeraktionen

Eine wichtige Designentscheidung für ein Tutorensystem besteht darin, in welcher Form Lernern Rückmeldungen zu deren Aktionen gegeben werden. Hierbei sind mehrere Faktoren zu berücksichtigen, die in unserer Anwendung auf Grund der verwendeten visuellen Repräsentation und der schlecht strukturierten zugrunde liegenden Domäne anders als in der Mehrheit der existierenden Tutorensystemen gelagert sind. Diese werden in den folgenden Unterabschnitten näher diskutiert.

4.1. Inhalt der Rückmeldungen

Wichtigster Aspekt ist sicherlich der Inhalt der Rückmeldungen. In vielen intelligenten Tutorensystemen wird der Lernende in Rückmeldungen über Fehler in seinen Aufgabenbearbeitungen informiert, z.B. in Form von Tipps, die schrittweise bis hin zur Preisgabe von Lösungsteilen gehen können. Dies ist in unserer Anwendung nicht möglich, da das System weder über ein präzises Lösungsmodell noch über eine definitive Fehlererkennung verfügt und auch nicht verfügen kann: so ist es z.B. durchaus möglich, dass der Lernende den Sinnzusammenhang eines Arguments verstanden hat, aber nicht die „passende“ Stelle im Text markiert hat. Ebenso kann durch eine vermeintlich „falsche“ Verwendung der im System angebotenen Elemente und Relationen nicht automatisch darauf geschlossen werden, dass der Nutzer die dadurch repräsentierten

Argumentationskonzepte nicht verstanden hat – obwohl dies durchaus möglich ist. Eine direkte Rückmeldung eines zu korrigierenden Fehlers ist in solchen unsicheren Diagnosesituationen nicht angemessen. Folgt man jedoch der Idee der „Schwäche“ in Argumentationsdiagrammen im Sinne der Lokalisierung von potenziellen Fehlern, so bietet sich es an, erkannte Schwächen dazu zu nutzen, den Lerner zur Selbsterklärung des betreffenden Argumentationsteils aufzufordern. Selbsterklärungen haben sich als effektive Lehrmethode in schlecht strukturierten Domänen erwiesen [SR02], und die Nutzung von erkannten Schwächen in Diagrammen vermeidet „unnötige“ Aufforderungen zur Selbsterklärung von Problemaspekten, in denen sich der Lernende gut auskennt. Hier ein Beispiel einer Systemrückmeldung zu einer erkannten Kontextschwäche „wichtiger Teil des Textes wurde nicht markiert“:

In Ihrer Analyse des Textes haben Sie einen wichtigen Teil übersehen. Bitte lesen Sie den nun gezeigten Teil nochmals– finden Sie hier eine durch einen Anwalt vorgeschlagene Regel, die nahe legen soll, wie der Fall zu entscheiden ist? Erklären Sie, welche Bedeutung der gezeigte Textausschnitt im Verfahren hat. Wenn Sie zur Ansicht kommen, dass hier ein Test geäußert wird, dann fügen Sie ein entsprechendes Element zum Diagramm hinzu und verlinken es mit dem Text. Notieren Sie Ihre Gedanken im Textfeld.

4.2. Prioritätsbestimmung für Rückmeldungen

Die große Mehrheit der existierenden intelligenten Tutorensysteme arbeitet dialogorientiert. Hierdurch ist zu jedem Zeitpunkt dem System genau bekannt, an welchem Teil einer Aufgabe der Benutzer arbeitet. Auf einer eindeutigen Zuordnung von Domänenmodell und Systemeingaben basieren z.B. die kognitiven Tutoren [An95]. Bei Benutzerschnittstellen, die eine solche Zuordnung nicht gewährleisten, kann Mehrdeutigkeit durch direkte Nachfragen aufgelöst werden, welche den Lernenden bitten, exakt zu spezifizieren was er gerade tut [ACC89]. Diese Nachfragen können jedoch für die Lernenden sehr störend sein – sie lenken vom eigentlichen Lerninhalt ab und bieten keinen direkten Mehrwert. Zudem sind diese Nachfragen nur dann technisch sinnvoll realisierbar, wenn die Menge der korrekten Lösungsverfahren einer Aufgabe präzise beschreibbar ist. Dies ist in LARGO nicht der Fall: zwar können Schwächen in Diagrammen genau beschrieben werden (und dienen als Basis für die Graphgrammatik), es gibt jedoch keine „idealen“ Diagramme.

Verzichtet man bei freieren Interaktionsformen auf direkte Nachfragen, so ist für ein Tutorensystem nicht mehr exakt bestimmbar, an welchem Teil einer Aufgabe ein Lerner arbeitet und wie genau der Stand seiner Aufgabenbearbeitung zu bewerten ist – d.h., eine optimale Rückmeldung oder Hilfestellung ist nicht genau bestimmbar. Gleichzeitig kann es aber durchaus möglich sein, dass das System nach Analyse des aktuellen Standes der Aufgabenbearbeitung eine Vielzahl von möglichen Rückmeldungen ermittelt. In der von uns betrachteten Anwendung ist dies der Fall: in Studien mit dem System traten Diagramme auf, die über 100 verschiedene Rückmeldungsmöglichkeiten aktivierten. Da eine Überfrachtung des Lernenden mit Feedback wenig sinnvoll ist, ergibt sich das Problem der Selektion von Rückmeldungsmöglichkeiten: wie kann das Tutorensystem aus der Vielzahl von zum Diagramm passenden Rückmeldungen die für den Lernenden beste auswählen?

Der von uns hierzu verwendete Algorithmus basiert auf dem Prinzip, möglichst genau den aktuellen Benutzungskontext des Lerners zu bestimmen. Hierzu wird zunächst für jede mögliche Rückmeldung eine Priorität berechnet. Dann wird pro Rückmeldungstyp (d.h. pro rückmeldungserzeugender Regel in der Graphgrammatik) jeweils die Rückmeldung mit der höchsten Priorität bestimmt und schließlich hieraus diejenigen fünf mit der höchsten Priorität ausgewählt. Diese werden dann dem Benutzer zur Auswahl angeboten (siehe 4.3). Dieses Verfahren garantiert, dass typgleiche Rückmeldungen, welche sich nur im Kontext im Diagramm unterscheiden, ausgeschlossen sind. Eine Priorität wird dabei auf Basis von vier Parametern berechnet:

1. Visuelle Information: Rückmeldungen zu aktuell sichtbaren Teilen des Textes bekommen eine hohe Priorität, nicht sichtbare Teile eine geringere abhängig von der Entfernung zum sichtbaren Textausschnitt. Auch der aktuell sichtbare Ausschnitt des Arbeitsbereichs wird berücksichtigt: sind alle zu einer Rückmeldung gehörenden Diagrammelemente im Moment sichtbar (und damit evtl. im Fokus des Lernenden), bekommt diese eine hohe Priorität.
2. Zeitliche Information: Da davon ausgegangen werden kann, dass kürzlich editierte Elemente des Diagramms wahrscheinlich im aktuellen Benutzungskontext des Lernenden liegen, bekommen Regeln, die Rückmeldungen zu diesen anbieten, eine hohe Priorität. Analog wird beim betrachteten Teil des Textes berücksichtigt, wann eine Passage das letzte Mal angezeigt wurde – Rückmeldungen, die sich auf Textteile beziehen, die vor kurzem gelesen wurden, erhalten eine höhere Priorität als solche, die vor langer Zeit (oder noch gar nicht) gelesen wurden.
3. Typbasierte Information: Jede Rückmeldung ist zu einer Regel in der Graphgrammatik assoziiert, wobei jede Regel auf viele Stellen des Diagramms passen kann. Um den Benutzer nicht mit zu vielen ähnlichen Rückmeldungen nacheinander zu langweilen, merkt sich das System, wann eine Regel der Grammatik das letzte Mal Basis für eine Rückmeldung war. Kürzlich zurückliegende erhalten eine niedrigere Priorität.
4. Phaseninformation: Im Rahmen von Pilotstudien mit dem System konnten wir typische wiederkehrende Verwendungsphasen (die nicht notwendig in einer festgelegten Reihenfolge auftreten) ausmachen: Orientierung im System (A), Markierung und Verlinkung des Textes (B), Verbindung von Diagrammelementen (C), Analyse und Korrektur des Diagramms (D) und Reflexion (E). Ist es möglich, diese Phasen im System zumindest heuristisch zu bestimmen, so kann die Rückmeldung auch hierauf eingehen, indem zur Phase passende Meldungen eine höhere Priorität erhalten.

Die Bestimmung der aktuellen Benutzungsphase ist der algorithmisch anspruchsvollste Teil der Kontextbestimmung. Wir stellen im Folgenden zwei Möglichkeiten dar, diese annäherungsweise zu bestimmen – eine exakte Berechnung ist in unserem Verfahren weder notwendig noch möglich, weil Phasen nicht scharf voneinander zu trennen sind und sich teilweise nicht in Aktionen im System manifestieren (z.B. kurzzeitige Reflexion über das Diagramm, ohne dass Eingaben gemacht werden).

Grammatikbasierter Ansatz

Die erste von uns betrachtete Möglichkeit zur Bestimmung der Benutzungsphase basiert auf den in der Graphgrammatik enthaltenen Regeln. Jede dieser Regeln lässt sich relativ eindeutig einer der fünf Benutzungsphasen zuordnen. Tabelle 1 illustriert diesen Zusammenhang.

Phase	Beispiel für Aktivierungsbedingung einer zugehörigen Produktionsregel
A: Orientierung im System	Diagramm enthält keine „Hypothesen“-Elemente
B: Markierung und Verlinkung des Textes	Keines der „Test“-Elemente im Diagramm annotiert eine wichtige Stelle im Textdokument, die die Formulierung eines Tests beinhaltet
C: Verbindung von Diagrammelementen	Ein „Test“-Element im Diagramm ist mit keinem „Hypothesen“-Element verbunden
D: Analyse und Korrektur des Diagramms	Das Diagramm enthält ein „Hypothesen“-Element, das in kausaler Beziehung („führt zu“) zu mehreren Tests steht
E: Reflexion	Im Diagramm sind zwei weder direkt noch indirekt verbundene „Test“-Elemente enthalten (Rückmeldung: Anregung zur Diskussion dieses Teils des Arguments)

Tabelle 1: Regelzuordnung zu Benutzungsphasen

Diese Zuordnung von einzelnen Produktionsregeln zu Phasen liefert die Basis für eine Heuristik zur Bestimmung der aktuellen Benutzungsphase, in der sich der Lernende befindet. Hierzu werden zunächst alle anwendbaren Regeln bestimmt und die Summe der anwendbaren Regeln pro Phase berechnet. Da die unterschiedlichen Phasen verschieden viele zugeordnete Regeln haben (in unserem Fall: A:4, B:11, C:11, D:13, E:5), werden diese absoluten Anzahlen noch hierzu ins Verhältnis gesetzt und resultieren in einer Wahrscheinlichkeitsverteilung für die aktuelle Benutzungsphase. Dieses Vorgehen stellt sicher, dass z.B. drei passende „Phase A-Regeln“ ein größeres Gewicht haben als drei passende „Phase B-Regeln“.

Ein Beispiel zu diesem Algorithmus: bei dem in Abbildung 1 gezeigten Argumentdiagramm sind insgesamt neun Regeln anwendbar. Davon gehören fünf zu Phase B (z.B.: „Wichtige Textstelle nicht beachtet“), eine zu Phase C und drei zu Phase E (z.B.: „Reflektiere über isolierte Hypothese“). Damit ergibt sich folgende Heuristik für die aktuelle Benutzungsphase: $P(A)=0\%$, $P(B)=40\%$, $P(C)=8\%$, $P(D)=0\%$ und $P(E)=52\%$. Der Lernende ist also mit hoher Wahrscheinlichkeit entweder noch in der Phase der Studie und Markierung des Textes oder schon in der Reflexionsphase. Diese Wahrscheinlichkeiten sind direkt als Prioritäten verwendbar: jede Regel erhält im Parameter „Phaseninformation“ die ihrer Phase entsprechende Wahrscheinlichkeit zugeordnet.

Aktionsbasierter Ansatz

Eine weitere Datenquelle, die zur heuristischen Bestimmung der aktuellen Benutzungsphase verwendet werden kann, sind die vom Benutzer im System durchgeführten Aktionen. So ist das Erzeugen einer Kante im Graph eher der Phase C zuzurechnen als der Phase B, und das Editieren eines bereits bestehenden Elements kann eher auf Phase D als auf Phase A hinweisen. Eine eindeutige Zuordnung von Aktionen zu Phasen ist jedoch nicht möglich. Die beobachtbaren Benutzeraktionen, welche in einem heuristischen Zusammenhang zu nicht beobachtbaren Benutzungsphasen stehen, legen die Verwendung eines Hidden Markov Modells (HMM) zur Bestimmung der wahrscheinlichsten aktuellen Benutzungsphase nahe. Die in Tabelle 2 dargestellte Matrix enthält die von uns im Modell verwendeten Übergangswahrscheinlichkeiten zwischen den Phasen. Diese beruhen auf unseren Beobachtungen in den ersten Einsätzen des LARGO-Systems und weiteren Annahmen, wie z.B. dass ein direkter Übergang von der „Startphase“ A in die „Revisionsphase“ D nicht möglich ist.

Von \ Nach	A	B	C	D	E
A: Orientierung im System	0,4	0,4	0,2	0	0
B: Markierung und Verlinkung des Textes	0,1	0,3	0,3	0,1	0,2
C: Verbindung von Diagrammelementen	0,1	0,2	0,2	0,3	0,2
D: Analyse und Korrektur des Diagramms	0,1	0,2	0,2	0,3	0,2
E: Reflexion	0	0,1	0,1	0,2	0,6

Tabelle 2: Übergangswahrscheinlichkeiten zwischen Benutzungsphasen im HMM

Diese Werte stellen eine Heuristik für typische Phasenabfolgen dar. Analog verhält es sich mit dem zweiten für die Spezifikation des HMM notwendigen Parameter, der Beobachtungswahrscheinlichkeit einzelner Aktionen in den jeweiligen Phasen. Eine allgemeine Zuordnung ist relativ leicht möglich (so weist z.B. das Löschen einer Relation im Diagramm auf Phase D oder E hin, während die Markierung eines Teils des Protokolls den Phasen A oder B zugeordnet werden kann), konkrete für das HMM benötigte Wahrscheinlichkeiten sind ohne empirische Nutzungsdaten zunächst nur schätzbar.

Aktion \ In Phase	A	B	C	D	E
Hinzufügen eines Diagrammelements	0,33	0,25	0	0,125	0,125
Löschen eines Diagrammelements	0	0	0	0,125	0,125
Verlinken eines Elements mit dem Verfahrensprotokoll	0	0,25	0	0	0
Markieren eines Teils des Verfahrensprotokolls	0,33	0,25	0	0	0
Beschreibung zu einem Diagrammelement hinzufügen	0,34	0,25	0	0	0
Beschreibung in einem Diagrammelement ändern	0	0	0	0,125	0,125
Hinzufügen einer Relation zum Diagramm	0	0	0,5	0,125	0,125
Löschen einer Relation im Diagramm	0	0	0	0,125	0,125
Beschriftung einer Relation erstellen	0	0	0,5	0,125	0,125
Beschriftung einer Relation ändern	0	0	0	0,125	0,125
Ändern des Typs einer Relation	0	0	0	0,125	0,125

Tabelle 3. Beobachtungswahrscheinlichkeiten im HMM

Wir haben uns in den durchgeführten Pilotversuchen zunächst für die in Tabelle 3 angegebenen Wahrscheinlichkeiten, die eine Gleichverteilung aller innerhalb einer Phase beobachtbaren Aktionen darstellen, entschieden. Daten von systematischen Studien mit dem System können zum Training dieses Modells verwendet werden, dies ist jedoch auf Grund der geringen Stichprobengröße bisheriger Studien noch nicht durchgeführt worden.

Die Wahrscheinlichkeiten für die aktuelle Benutzungsphase eines Systemnutzers können auf Basis der Tabellen 2 und 3 mittels des Viterbi-Algorithmus [Vi67] bestimmt werden. Dieser gibt direkt die Wahrscheinlichkeiten für spezifische Phasenfolgen aus. Über Summierung erhält man die Wahrscheinlichkeit einer spezifischen Endphase. Diese nutzen wir (analog zum grammatikbasierten Ansatz) für die Prioritätsbestimmung von einzelnen Regeln. Im Gegensatz zum grammatikbasierten Ansatz, bei dem die Phasenbestimmung alleine auf Basis des aktuellen Zustands des Diagramms durchgeführt wird, berücksichtigt der aktionsbasierte Ansatz dessen Entstehungsgeschichte. Damit wird der Kontext des Lernenden eher prozess- als produktorientiert definiert. Eine eindeutige zum Beispiel in Abbildung 1 passende Heuristik einer Benutzungsphase ist daher ohne Angabe der Historie des Diagramms nicht möglich. Für einen sehr einfachen Erstellungsprozess, der aus den minimal zur Konstruktion des Diagramms notwendigen 36 Schritten besteht,

ergeben sich nach dem Viterbi-Algorithmus folgende Wahrscheinlichkeiten: $P(A)=7\%$, $P(B)=18\%$, $P(C)=18\%$, $P(D)=28\%$ und $P(E)=29\%$. Es wird ersichtlich, wie diese von den Ergebnissen des grammatikbasierten Ansatzes abweichen – auch wenn in beiden Verfahren Phase E als die Wahrscheinlichste ausgegeben wird. Insbesondere sind auf Grund der (notwendigerweise) nicht exakten Zuordnung von Beobachtungen zu Phasen die Ausgabewahrscheinlichkeiten weniger scharf voneinander abgegrenzt als im grammatikbasierten Verfahren. In den ersten Anwendungen von LARGO haben sich beide Verfahren als praktisch geeignet (im Sinne der Auswahl von „passenden“ Rückmeldungen) erwiesen.

4.3. Zeitpunkt für Rückmeldungen

Ein weiterer wichtiger Faktor beim Design eines Rückmeldungsmechanismus ist, *zu welchem Zeitpunkt* vom System Feedback angeboten wird. Forschungsergebnisse lassen hier zwar keine allgemeingültigen Aussagen über einen besten Zeitpunkt zu, einige Studien belegen jedoch, dass sofortige Rückmeldungen zumindest so effektiv wie verzögerte Rückmeldungen sind [Ba91]. In unserer Anwendung sind sofortige Rückmeldungen jedoch nur wenig hilfreich: wenn Fehler nicht exakt zu diagnostizieren sind, stören häufige Systeminterventionen (insbesondere wenn sie zu kognitiv aufwändigen Selbsterklärungen führen) den Benutzer eher und unterbrechen ihn in seinem Gedankengang, als dass sie hilfreich sind.

Zusätzlich ist es bei einer konstruktiv orientierten Umgebung wie der von uns verwendeten nicht immer erkennbar, wann eine „Eingabe“ beendet ist: ein im Diagramm ausgedrückter Gedanke kann durchaus mehrere Elemente und Relationen umfassen. Es ist unangemessen, wenn hier während der Erstellung des Diagramms das System mit „Hilfestellungen“, die etwa ausdrücken, dass noch wichtige Teile im Diagramm fehlen, interveniert. Aus diesen Gründen haben wir uns dazu entschlossen, den Benutzer aktiv entscheiden zu lassen, wann er Rückmeldungen bzw. Hilfe bekommen möchte, selbst wenn dies das Risiko birgt dass ggf. verfügbare Hilfe nicht genutzt wird [AK00]. Der Benutzer kann sich über einen „Hilfe“-Button jederzeit einzeilige Kurzversionen von fünf Rückmeldungen anzeigen lassen. Dies trägt der nicht eindeutig bestimmbaren „besten“ Rückmeldung Rechnung und reduziert dabei gleichzeitig die kognitive Belastung (Lesen zu vieler kompletter Rückmeldungstexte) auf ein Minimum. Wählt der Lernende eine dieser Kurzversionen aus, so bekommt er die entsprechende Meldung angezeigt und wird ggf. zu einer Selbsterklärung aufgefordert. Der Zusammenhang einer erkannten Schwäche im Argumentationsdiagramm wird dabei visuell hervorgehoben. Dieser Prozess wird in Video 2 [6] dargestellt.

5. Versuchsplan

Im Herbst 2006 wurde LARGO in einer ersten kontrollierten Studie eingesetzt. Das Ziel dieser Studie war die Untersuchung, ob das graphische Repräsentationsformat und die Systemrückmeldungen von LARGO Studierenden besser beim Lernprozess helfen können

als eine „traditionelle“ Anwendung. Letztere war in Form eines digitalen Notizblocks (zur Annotierung der Gerichtsprotokolle in Textform) mit der zusätzlichen Funktion, Protokollteile markieren zu können, realisiert.

Das Experiment fand mit 38 Studierenden im ersten Jurasemester der University of Pittsburgh statt. 28 Teilnehmer beendeten die Studie und erhielten 80\$ für ihre Teilnahme. Die in der Studie untersuchten Fälle hatten das Prinzip „Gerichtbarkeit“ als Thema— dieses wird im 1. Semester des Studiums behandelt. Die Teilnehmer der Studie wurden zufällig zu den beiden Versuchsgruppen (LARGO / digitaler Notizblock) zugeordnet.

Die Studie wurde in vier je zweistündigen Sitzungen im Laufe eines Monats durchgeführt. Die erste Sitzung beinhaltete einen Eingangstest und eine kurze Einführung in die zu verwendende Software (LARGO oder digitaler Notizblock) sowie in das Prinzip der Argumentation mit Tests und hypothetischen Sachverhalten. In der zweiten und dritten Sitzung arbeiteten die Teilnehmer dann mit insgesamt vier Protokollen aus zwei Fällen, die mit dem Prinzip „Gerichtbarkeit“ zusammenhängen. Die Teilnehmer in der „LARGO“-Gruppe repräsentierten die Fälle in der graphischen Notation und konnten auf Nachfrage jederzeit Systemrückmeldungen zu ihrem Diagramm bekommen. Hierbei war die Bestimmung der Benutzungsphase zustandsbasiert über die Graphgrammatik realisiert (siehe Abschnitt 4.2). Die Teilnehmer in der anderen Gruppe („Text“) erstellten textuelle Notizen zu den Protokollen und markierten wichtige Stellen darin; sie konnten keine Rückmeldungen zu ihren Aktionen im System bekommen. In der vierten Sitzung fand der Abschlusstest statt. Eingangs- und Abschlusstest bestanden aus Multiple-Choice-Fragen, die das Wissen und die Fähigkeiten der Teilnehmer in Bezug auf juristische Argumentation und das von uns verwendete Argumentationsmodell (Tests und Hypothesen) überprüften. Zusätzlich zu den Multiple-Choice-Tests wurden in den zweiten und dritten Sitzungen alle Aktionen der Teilnehmer in den Systemen automatisch gespeichert.

6. Ergebnisse und Diskussion

Die von uns in der Studie untersuchte Primärhypothese war, dass LARGO den Teilnehmern besser beim Lernen des Argumentationsmodells hilft als der digitale Notizblock. Eine erste vergleichende Analyse der Eingangs- und Abschlusstests der beiden Gruppen wurde in [Pi07] veröffentlicht. Hinsichtlich der Primärhypothese können die Ergebnisse wie folgt zusammengefasst werden: die Studenten, die LARGO verwendet haben, waren im Abschlusstest durchschnittlich erfolgreicher als die Teilnehmer in der Kontrollgruppe, der Unterschied ist jedoch statistisch nicht signifikant ($t(1,26)=-.92$, $p>.1$). Eine Post-Hoc Analyse, bei der die Teilnehmer in drei Klassen gemäß ihres „LSAT“-Ergebnisses (Eingangstest zur juristischen Fakultät; diese Tests sind eine gute Vorhersage für Erfolg im Jurastudium) eingeteilt wurden, zeigte, dass die „unteren“ Klassen am meisten von LARGO profitierten. Betrachtet man nur die Studenten aus den unteren beiden Klassen, so erzielten die „LARGO“-Nutzer signifikant höhere Resultate als die Kontrollgruppe in vielen Fragekategorien des Abschlusstests. Dies schließt die für das zu lernende Argumentationsmodell wichtigste Fragenkategorie „Bewertung eines hypothetischen Sachverhalts hinsichtlich eines Test“ ein ($t(1,17)=2.73$, $p<.05$). Damit stützen die Ergebnisse unsere Primärhypothese, wenn auch nur beschränkt auf die Gruppe der

juristisch weniger talentierten Teilnehmer (gemäß LSAT-Messung). Für die anderen erwies sich LARGO nicht als besser (aber auch nicht als schlechter) als in den Rechtswissenschaften „traditionelle“ textbasierte Notationsformen.

Eine nachfolgend erfolgte Analyse der Logdateien zeigte, dass alle LARGO-Nutzer die Systemrückmeldungen regelmäßig und häufig genutzt haben – der „Hilfe“-Button wurde im Mittel 10,1-mal pro Gerichtsprotokoll, d.h. pro Stunde, verwendet. Auch die Mitglieder der „besten“ LSAT-Klasse haben die Rückmeldungen häufig genutzt (mit durchschnittlich 17,9 Aufrufen pro Stunde sogar häufiger als der Durchschnitt). In 75 Prozent der Fälle, in denen über die Rückmeldungsfunktion die Kurzversionen der Hilfen angezeigt wurden, haben die Teilnehmer eine dieser Kurzversionen ausgewählt und sich die volle Rückmeldung anzeigen lassen. Es ließ sich weiterhin feststellen, dass die Verwendung der Hilfsfunktion nicht über die Zeit abnahm, sondern tendenziell sogar zunahm (12,3 und 8,6 in den letzten beiden Protokollen; 7,3 und 9,8 in den ersten beiden).

Insgesamt lassen diese relativ hohen und kontinuierlichen Nutzungsdaten die Interpretation zu, dass die Teilnehmer die von LARGO generierten Rückmeldungen zu ihren Diagrammen als wertvoll bzw. hilfreich erachtet haben (andernfalls hätte sicherlich nach eventueller anfänglicher Neugierde die Nutzung schnell aufgehört). Dies lässt wiederum den Rückschluss zu, dass die in diesem Artikel dargestellte Bestimmung des Benutzungskontextes ein angemessener Ansatz ist – durch diesen war es möglich, aus der Fülle der möglichen Rückmeldungen zu einem Diagramm diejenigen auszuwählen, am besten zu den aktuellen Aktivitäten des Nutzers passen. Damit ist der wichtigste Zweck der Bestimmung des Benutzungskontextes erfüllt.

Wir haben uns darüber hinaus nachfolgend mit der Frage beschäftigt, in wie weit der Algorithmus zur Bestimmung des Benutzungskontexts auch zur Diagnose von Lernerfolg nutzen lässt – kann aus dem erstellten Diagramm vorhergesagt werden, ob ein Student die Argumentationskonzepte und –techniken gelernt hat? Um dies zu evaluieren, haben wir die von den Teilnehmern der Studie erstellten Endversionen der Diagramme nochmals durch LARGO analysieren lassen. Hierbei waren wir insbesondere an der Phasenheuristik interessiert – geht eine durch LARGO geschätzte hohe Wahrscheinlichkeit der „Analyse- und Reflexionsphasen“ mit besseren Testergebnissen einher?

Eine Korrelationsanalyse bestätigte dies nicht: für keine der Fragenkategorien im Ausgangstest besteht eine statistisch signifikante Korrelation mit der diagnostizierten Phasenwahrscheinlichkeit. Jedoch ließen sich folgende Zusammenhänge beobachten: Die Häufigkeit der Nutzung von Rückmeldungen korreliert positiv mit der (LARGOgenerierten) Wahrscheinlichkeit, dass die Studenten sich am Ende der Sitzung in der Reflexionsphase befinden (Pearsons $r=.53$, $p<.05$). Dies lässt sich als Indikator dafür deuten, dass die LARGO-Rückmeldungen den Nutzern dabei hilfreich waren, von den „ersten“ Phasen (Orientierung im System, Textmarkierung und Verlinkung) zu den späteren Phasen zu gelangen.

Die Arten und Anzahlen von Schwächen in den von den Studenten erstellten Diagrammen zeigen weitere interessante Ergebnisse auf. Die Anzahl der diagnostizierten Schwächen ist negativ korreliert zum Ergebnis von Fragen im Ausgangstest, die Wissenstransfer (Argumentation in einem anderen Gebiet als „Gerichtsbarkeit“) zum Inhalt hatten ($p=-.54$, $p<.05$) – Diagramme mit weniger Schwächen wurden also tendenziell von Personen erzeugt, die diese Transferfragen besser beantworteten. Dieses Ergebnis ließ sich für die Fragentypen, die keinen Wissenstransfer erforderten, nicht beobachten ($r=.19$, $p>.5$). Eine

dritte bemerkenswerte Korrelation ist, dass die verschiedenen Arten von Schwächen (d.h., die verschiedenen Weisen, auf die ein Diagramm gemäß der Graphgrammatik Schwächen aufweist) der Phasen 4 und 5 positiv mit dem Ergebnis der Fragen im Ausgangstest korreliert ist, die Wissen im Bereich „Gerichtbarkeit“ testeten ($r=.65$, $p<.01$). Je mehr „höhere“ Arten von Schwächen ein Diagramm aufweist, desto bessere Ergebnisse wurden von den Autoren dieser Diagramme also in dieser Fragenkategorie erzielt. Dies ist auf den ersten Blick überraschend, lässt sich jedoch dadurch erklären, dass diese „höheren“ Schwächentypen nicht ausschließlich für „fehlerhafte“ Repräsentationen des Argumentationsverlaufs stehen, sondern auch Gelegenheiten zur Reflexion über ein (gutes) Diagramm beinhalten (siehe Abschnitt 3).

Diese Ergebnisse der Analysen haben sicherlich nur begrenzte Aussagekraft hinsichtlich der Fragestellung, ob die Diagrammanalyse von LARGO mittels der Phasheuristik eine Voraussage für Lernerfolg ermöglicht – die Untersuchungsmethodik (Korrelation; keine Aussage über kausale Zusammenhänge), die geringe Stichprobengröße (15 Teilnehmer in „LARGO“-Versuchsgruppe) sowie die nur eingeschränkt beobachtbaren Zusammenhänge machen verallgemeinerbare Aussagen gegenwärtig unmöglich. Jedoch halten wir die im Rahmen der Post-Hoc Analyse der ersten Studie durchgeführten Analysen für interessant und werden die Hypothese, dass wenig Schwächen in Diagrammen (aber viele Arten von „Schwächen“ der höheren Benutzungsphasen) eine Vorhersage des Lernerfolgs ermöglichen, weiter untersuchen.

7. Zusammenfassung und Ausblick

Das in diesem Artikel beschriebene System dient dazu, Jurastudenten im Bereich des gerichtlichen Argumentierens auszubilden. Der Ansatz beruht auf der Annotation von Gerichtsprotokollen mittels graphischer Modelle, welche eine Domänenontologie repräsentieren. Die durch die Studenten erstellten und mit dem Protokoll verlinkten Modelle werden durch ein Verfahren, welches technisch auf Graphgrammatiken und kollaborativem Filtern beruht, auf Schwächen überprüft. Das System liefert Rückmeldungen zu Schwächen in Form von Aufforderungen zu Selbsterklärungen. Dies erscheint im betrachteten Problembereich, in dem oft viele Lösungen akzeptierbar sind und eine klare Entscheidung zwischen richtig und falsch teils unmöglich ist, sinnvoller als Fehlermeldungen, wie sie in traditionellen Tutorensystemen verwendet werden.

Im Unterschied zu eher dialogorientierten Systemen, bei denen der aktuelle Fokus des Lernenden leicht durch das aktive Eingabefeld zu bestimmen ist, tritt in unserer Anwendung das Problem auf, dass zu einem Diagramm typischerweise viele Rückmeldungen möglich sind, ohne dass dabei a priori klar wäre, welche zur aktuellen Aktivität des Benutzers am besten passt. Die Beantwortung dieser Frage ist, insbesondere auf Grund der von uns verwendeten kognitiv sehr aufwändigen Rückmeldungsart, essentiell. Dieser Artikel stellt ein Verfahren zur Prioritätsbestimmung für Rückmeldungen vor, welches mehrere Faktoren (lokaler und zeitlicher Kontext, Rückmeldungshistorie und Benutzungsphase) des Benutzungskontexts berücksichtigt. Für die heuristische Bestimmung der Benutzungsphase haben wir zwei Verfahren (prozessorientiert vs. produktorientiert) näher diskutiert. Im Rahmen von Pilotstudien haben sich beide Ansätze als grundsätzlich praktikabel erwiesen.

In einer kontrollierten Studie wurde LARGO (mit der prozessorientierten Variante zur Bestimmung der Benutzungsphase) eingesetzt. Die Ergebnisse der Studie zeigten, dass das System insbesondere für weniger begabte Lernende gut geeignet ist – im Vergleich zu einer rein textuellen Umgebung ohne Rückmeldungen führte LARGO zu Verbesserungen im Bereich der Analyse von Argumentationszusammenhängen bei diesen Studierenden. Die Hilfsfunktionen wurde von den Studierenden häufig genutzt, und es hat sich gezeigt, dass die von LARGO verwendete Heuristik für die Bestimmung der Benutzungsphase (welche für die Generierung der Rückmeldungen verwendet wird) möglicherweise auch als Vorhersageinstrument für den Lernfortschritt der Studenten einsetzbar sein könnte: Lernende, deren Diagramm weniger Schwächen (aber viele Arten von „Schwächen“ höherer Phasen) aufwies, erzielten in gewissen Kategorien des Ausgangstests bessere Ergebnisse.

Literaturhinweise

- [ACC89] Anderson, J.; Conrad, F.; Corbett, A.: Skill acquisition and the LISP tutor. *Cognitive. In: Science* 13. 1989; p. 467-505
- [AK00] Aleven, V.; Koedinger, K.: Limitations of Student Control: Do Students Know when they need help? In: *Proceedings of the International Conference on Intelligent Tutoring Systems*. Springer Verlag, Berlin, 2000, p. 292-303
- [Ai99] Ainsworth, S.: The functions of multiple representations. *Computers and Education* 33. 1999; p. 131-152
- [AI03] Aleven, V.: Using Background Knowledge in Case-Based Legal Reasoning: A Computational Model and an Intelligent Learning Environment. *Artificial Intelligence* 150. 2003, p. 183-238
- [An95] Anderson, J.; Corbett, A.; Koedinger, K.; Pelletier, R.: Cognitive tutors: Lessons learned. In: *The Journal of Learning Sciences* 4. 1995, p. 167-207
- [As90] Ashley, K. D.: *Modeling Legal Argument: Reasoning with Cases and Hypotheticals*. MIT Press/Bradford Books, Cambridge, 1990
- [As07] Ashley, K. D.: Interpretive Reasoning with Hypothetical Cases. In *Proceedings of the 20th International FLAIRS Conference, Special Track on Case-Based Reasoning*. Key West, 2007
- [Ba91] Bangert-Drowns, R. L.; Kulik, C.; Kulik, J. A.; Morgan, M. T. : The instructional effect of feedback in test-like events. In: *Review of Educational Research*, 61(2). 1991; p. 213-238
- [Br06] Van den Braak, S. W.; Van Oostendorp, H.; Prakken, H.; Vreeswijk, G.: A critical review of argument visualization tools: Do users become better reasoners? In: *Workshop Notes of the ECAI-06 Workshop on Computational Models of Natural Argument*. Riva del Garda, 2006
- [Ca03] Carr, C.: Using Computer Supported Argument Visualization to Teach Legal Argumentation. In: *Visualizing Argumentation*. Springer Verlag, London, 2003, p. 75-96

[Ch00] Chi, M.: Self-explaining expository texts: The dual process of generating inferences and repairing mental models. In: *Advances in Instructional Psychology*. Lawrence Erlbaum, Hillsdale, 2000, p. 161-238

[Cu01] Curran, V. G.: Romantic Common Law, Enlightened Civil Law: Legal Uniformity and the Homogenization of the European Union. *7 Columbia Journal of European Law* Winter 63.

[Ge02] van Gelder, T: Argument Mapping with Reason!Able. In: *The American Philosophical Association Newsletter on Philosophy and Computers*. 2002, p. 85-90

[Ha07] Haft, F.; Hassenpflug, P; Lecker, H.: Proposal of a System for Computer-Based Case and Evidence. In: *Proceedings of IMF 2007*, S. 51-60.

[KR02] Konstan, J.; Riedl, J.: Collaborative Filtering: Supporting social navigation in large, crowded infospaces. In: *Designing Information Spaces: The Social Navigation Approach*. Springer Verlag, Berlin, 2002, p. 43-81

[Ly06] Lynch, C.; Ashley, K. D.; Aleven, V.; Pinkwart, N.: Defining Ill-Defined Domains; A literature survey. In: *Proceedings of the Workshop on Intelligent Tutoring Systems for Ill-Defined Domains at the 8th International Conference on Intelligent Tutoring Systems*. National Central University, Jhongli, 2006, p. 1-10.

[Lu98] Lundmark, T.: Book Review, *Interpreting Precedents: A Comparative Study*. MacCormick, D.N.; Summers, R.S. (ed.): *46 American Journal of Comparative Law* 211.

[MS97] MacCormick, D.; Summers, R.: *Interpreting Precedents: A Comparative Study*. Dartmouth Publishing, Aldershot, 1997

[MB01] Muntjewerff, J.; Breuker, J.: Evaluating PROSA, a system to train solving legal cases. In: *Proceedings of the International Conference on Artificial Intelligence in Education*. IOS Press, Amsterdam, 2001, p. 278–285

[Ph95] Philipps, L.: Just Decisions Using Multiple Criteria Or: Who Gets the Porsche? An Application of Ronald R. Yager's Fuzzy Logic Method. In: *Proceedings of ICAIL 1995*, p. 195-200.

[Pi06a] Pinkwart, N.; Aleven, V.; Ashley, K.; Lynch, C.: Toward Legal Argument Instruction with Graph Grammars and Collaborative Filtering Techniques. In: *Proceedings of the International Conference on Intelligent Tutoring Systems*. Springer Verlag, Berlin, 2006, p. 227–236

[Pi06b] Pinkwart, N.; Aleven, V.; Ashley, K.; Lynch, C.: Using Collaborative Filtering in an Intelligent Tutoring System for Legal Argumentation. In: *Proceedings of Workshops held at the 4th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*. Lecture Notes in Learning and Teaching. National College of Ireland, Dublin, 2006, p. 542-551

[Pi07] Pinkwart, N.; Aleven, V.; Ashley, K.; Lynch, C.: Evaluating Legal Argument Instruction with Graphical Representations using LARGO. To appear in *Proceedings of the International Conference on Artificial Intelligence in Education*. IOS Press, Amsterdam, 2007

- [Ro92] Routen, T.: Reusing formalisations of legislation in a tutoring system. In: Artificial Intelligence Review 6. 1992, p. 145–159
- [RR04] Reed, C.; Rowe, G.: Araucaria: Software for Argument Analysis, Diagramming and Representation. In: International Journal of AI Tools 14, 2004; p. 961-980
- [RS97] Rekers, J.; Schürr, A.: Defining and Parsing Visual Languages with Layered Graph Grammars. In: Journal of Visual Languages and Computing, 8(1), 1997; p. 27-55.
- [SR02] Schworm, S.; Renkl, A.: Learning by solved example problems: Instructional explanations reduce selfexplanation activity. In: Proceedings of the Annual Conference of the Cognitive Science Society. Lawrence Erlbaum, Mahwah, 2002, p. 816-821
- [To58] Toulmin, S.: The Uses of Argument. Cambridge, Cambridge University Press, 1958
- [Ve03] Verheij, B.: Artificial argument assistants for defeasible argumentation. In: Artificial Intelligence, 150, 2003; p. 291-324
- [Vi67] Viterbi, A.: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. In: IEEE Transactions on Information Theory, 13(2), 1967; p. 260–267

[1] <http://www.take-law.de>

[2] <http://www.normfall.de>

[3] <http://bgb.jura.uni-hamburg.de>

[4] <http://www.knowledgetools.de>

[5] eeed.campussource.de/_/archive/5/1608/video1.avi (Zum Abspielen des Videos ist ggf. der TSCC-codec erforderlich)

[6] eeed.campussource.de/_/archive/5/1608/video2.avi (Zum Abspielen des Videos ist ggf. der TSCC-codec erforderlich)