Erfahrungen bei der Erstellung des Web-basierten Kurses "Grammar Formalisms and Parsing"*

Frank Richter

11. September 2003

Zusammenfassung

Das Papier beschreibt die Komponenten des Web-basierten Kurses *Grammar Formalisms and Parsing*, die Voraussetzungen, auf denen seine Erstellung beruht, und Arbeitserfahrungen seiner Autoren bei der Softwareentwicklung und dem Einsatz des Materials in der Lehre. Besonderes Augenmerk wird dabei auf die materiellen Ressourcen und den Personalaufwand gelegt. Ein vorrangiges Anliegen des Papiers ist es, praktische Probleme zu beschreiben, mit denen sich die Autoren unter den heutigen technischen Gegebenheiten konfrontiert sehen.

We describe the components of the web based course *Grammar Formalisms and Parsing*, the prerequisites of its creation, and the experiences of its authors with the software development and the use of the course materials in class. Special attention is paid to the necessary resources and man power for creating the course. The paper focuses especially on practical problems that are caused by the software equipment available to date.

1 Kursgegenstand

Der Kurs *Grammar Formalisms and Parsing*[1] entsteht am Seminar für Sprachwissenschaft (SfS), Abteilung Computerlinguistik, an der Universität Tübingen in einem Teilprojekt des Konsortiums *Medienintensive Lehrmodule in der Computerlingustik-Ausbildung* (MiLCA). Ziel dieses Web-basierten Kurses ist es, einem seiner Herkunft nach heterogenen Publikum die komplexen Zusammenhänge zwischen moderner linguistischer Theoriebildung, ihrer mathematischen Fundierung, der Implementierung logisch exakt spezifizierter Grammatiken sowie den dazu angewandten Parsingtechnologien zu vermitteln. Zentrales verbindendes Konzept sind dabei so genannte *Merkmalsstrukturen* (engl. *Feature Structures*).

Merkmalsstrukturen sind ein mathematisches Konstrukt, das anfänglich in Form von Record Structures aus der Informatik stammt und später in der Sprachwissenschaft als geeignete Datenstruktur zur Modellierung von sprachlichen Entitäten wie zum Beispiel

*Das diesem Bericht zugrunde liegende Projekt MiLCA wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01 NM 167 A gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

[1] `http://milca.sfs.uni-tuebingen.de/A4/HomePage/top.html`

von Sätzen, von beliebigen Phrasen, oder auch nur von Bestandteilen derselben erkannt und adaptiert wurde. Heute werden Merkmalsstrukturen auch als graphentheoretische Objekte betrachtet, welche Gegenstand sowohl modelltheoretischer Studien zu ihrer Beschreibbarkeit innerhalb von Logiken verschiedener Ausdruckstärke als auch computationell ausgerichteter Studien sind, die sich etwa mit Graphentransformationen beschäftigen. Zu den prominenten Grammatiktheorien, die sich der Merkmalsstrukturen zum Zweck der Repräsentation sprachlicher Objekte bedienen, zählen die *Generalized Phrase Structure Grammar* (GPSG, Gazdar et al., 1985), die *Lexical Functional Grammar* (LFG, Bresnan, 1982) und die *Head-driven Phrase Structure Grammar* (HPSG, Pollard and Sag, 1994).

Von Seiten der mathematischen Logik wurde eine Reihe von für die Sprachwissenschaft geeigneten Merkmalslogiken entwickelt, welche Merkmalsstrukturen als mengentheoretische Konstrukte betrachten, die zur Interpretation von Ausdrücken formaler Sprachen verwendet werden. Diese Betrachtungsweise erlaubt eine präzise Untersuchung der formalen Eigenschaften von Sprachtheorien, die sich solcher Merkmalslogiken zum Zwecke der Beschreibung natürlicher Sprachen bedienen. Die Grundidee dabei ist, dass eine Grammatik formuliert wird als eine Menge von logischen Ausdrücken über einer Signatur einer Merkmalslogik. Eine modelltheoretisch näher charakterisierte Klasse von Merkmalsstrukturen (oder generellerer algebraischer Strukturen) wird dann aufgefasst als entweder die intendierte natürliche Sprache selbst, oder aber als mathematische Idealisierung derselben. Die Verifizierung beziehungsweise Falsifizierung grammatischer Vorhersagen geschieht durch einen Vergleich der im Modell der Grammatik vorkommenden Objekte und ihrer Struktur mit empirisch beobachtbaren sprachlichen Phänomenen, sowie durch Annahmen darüber, wie sich sprachliche Phänomene als Strukturen im Modell der Grammatik darstellen müssten.

Schließlich werden Merkmalslogiken auch in der Logikprogrammierung eingesetzt und eignen sich als (Bestandteile von) Programmiersprachen zur Grammatikimplementierung. Die Verbindung von Parsingsystemen mit Merkmalslogiken hat eine lange Tradition in der Computerlinguistik und kann als einer der technischen Standards bei der automatischen Tiefenanalyse natürlicher Sprache angesehen werden.

Die für den Kurs gewählte Grammatiktheorie ist die HPSG, da sie in ausgezeichneter Weise die genannten grammatiktheoretischen Annahmen, mathematischen Grundlagen und implementatorischen Voraussetzungen in sich vereinigt. Von ihren ersten Anfängen Mitte der 80er Jahre des 20. Jahrhunderts an wurden die sprachwissenschaftlichen Grundkonzepte der HPSG in (über die Jahre wechselnden) Merkmalslogiken präzisiert. Seitdem wurde eine große Anzahl von Sprachen analysiert, und es existiert eine Vielzahl sprachwissenschaftlicher Untersuchungen der verschiedensten empirischen Phänomenbereiche. Obwohl der Schwerpunkt nach wie vor auf syntaktischen Studien liegt, gibt es darunter doch auch detaillierte morphologische, phonologische und semantische Analysen.

Von Anfang an wurden außerdem Anstrengungen hinsichtlich computationeller Anwendungen unternommen. In der Tat ist es so, dass der Bestandteil "HP" der HPSG nicht nur von einer Analogie zur GPSG und der zentralen Stellung der Idee syntaktischer Köpfe (*heads*) und Phrasen herrührt, sondern auch von den Initialen der Firma Hewlett Packard inspiriert wurde, in deren Hauptquartier in Palo Alto in den 80er Jahren über Implementierungsfragen erster Entwürfe der HPSG nachgedacht wurde. Entwicklungen in der Merkmalslogik führten dann rasch dazu, dass seit den frühen 90ern Grammatikimplementierungsumgebungen zur Verfügung stehen, deren Design vorzugsweise auf

HPSG-artige oder von HPSG inspirierte Grammatiken abzielt.

Reiz und Herausforderung der Erstellung eines netzbasierten Kurses zu *Grammar Formalisms and Parsing* in dem damit charakterisierten Umfeld ist seine Lokalisierung an der Schnittstelle dreier Wissenschaftsdisziplinen: der Informatik, der theoretischen Sprachwissenschaft und der mathematischen Logik. Durch Verwendung des Brückenkonzepts der Merkmalsstrukturen und durch geeignete, interaktive und visualisierende Software soll es Studierenden der verschiedenen Disziplinen ermöglicht werden, aufbauend auf ihrem spezifischen Wissens- und Erfahrungshintergrund ihre Kompetenzen über die Grenzen ihrer Herkunftsdisziplin zu erweitern.

Da auf Fragen des wissenschaftlichen Hintergrundes des Kurses und auf die verwendete Grammatikimplementierungsumgebung und Visualisierungssoftware aus Platzgründen unten nicht weiter eingegangen werden kann, seien interessierte Leser auf die in der Bibliographie erwähnte Literatur verwiesen. Wie dort ersichtlich, hat *Grammar Formalisms and Parsing* neben dem Lernmaterial einschlägige Veröffentlichungen zu Aspekten des Kurses und zu den wissenschaftlichen Grundlagen seines Gegenstandes und seiner Software hervorgebracht. An dieser Stelle sei insbesondere auf Meurers et al. (2002) hingewiesen, wo unter anderem didaktische Fragen des gewählten Kursformats diskutiert werden. Online Ressourcen finden sich auch auf den eingangs erwähnten Webseiten des Projekts.

Das vorliegende Papier ist folgendermaßen strukturiert: In Abschnitt 2 beschäftigen wir uns mit der Ausgangssituation des Projektes zu Beginn der Projektförderung und mit den daraus abgeleiteten Planungsmaßnahmen. Abschnitt 3 beschreibt die Voraussetzungen und Probleme, die durch die eingesetzte Kurssoftware und durch die verwendeten Betriebssysteme gegeben waren oder entstanden sind. Abschnitt 4 skizziert die Entwicklung, die der Kurs im Förderzeitraum genommen hat, und geht dabei insbesondere auch auf den Personalaufwand ein, der zu seiner Umsetzung notwendig war. Abschnitt 5 beschäftigt sich mit der Akzeptanz des Kurses von Seiten der Studierenden und zieht ein Fazit aus den Erfahrungen mit dem Projekt.

2 Entwicklungsstrategie

Der Gegenstand des Kurses, ein an den Schnittstellen verschiedener Wissenschaftsdisziplinen angesiedeltes und weitgehend mathematisch präzisierbares Forschungsgebiet, erschien aus mehreren Gründen besonders geeignet für eine Aufbereitung im Format des Web-basierten Trainings.

Zunächst war es klar, dass zentrale Bestandteile der zu vermittelnden Inhalte algorithmisch umsetzbar und durch Software visualisierbar waren. In der Tat leistete der zum Zeitpunkt der Antragstellung existierende, für Forschungszwecke entwickelte Prototyp der für den Kurs vorgesehenen Grammatikimplementierungsumgebung TRALE zusammen mit dem grafischen Benutzerinterface *GRammar vISUalization tool* (GRISU) von seiner Anlage her bereits einen beträchtlichen Teil dessen, was an Anforderungen an eine interaktive Parsing-Software gestellt werden muss. Auf der anderen Seite war erkennbar, dass eine intelligente und interaktive Visualisierung der logischen, modelltheoretischen Grundlagen der HPSG, die jenseits jener Bestandteile der mathematischen Grundlagen der Theorie liegen, die für Grammatikimplementierungsumgebungen unmittelbar relevant sind, aufgrund ihrer bekannten mathematischen Eigenschaften ebenfalls weitgehend möglich war. Allerdings gab es dafür keine Vorarbeiten und insbesondere keine Erfahrung mit Programmiersprachen, die eine Web-fähige und relativ plattformunabhängige Benutzeroberfläche erlaubten.

Die Verbindung der Domänen der Grammatikimplementierung und der logischen Grundlagen der HPSG innerhalb eines terminologischen, notationellen und visuellen Gesamtkonzeptes erschien von vornherein eine reizvolle Aufgabe. Ihre Durchführung versprach Synergieeffekte für die Entwicklung der Grammatikimplementierungsumgebung und erschien geeignet, bei den Studierenden ein tieferes Verständnis des HPSG-Formalismus an sich zu erzielen. Ein solches besseres Verständnis wiederum ist Voraussetzung für einen kreativen und präzisen Umgang mit den Werkzeugen der Grammatiktheorie bei der Formulierung von Generalisierungen über den empirischen Gegenstand der Sprachwissenschaft, nämlich über sprachliche Strukturen.

Wie beschrieben ist das Thema an sich geeignet für eine Umsetzung in Form von elektronischem Lernmaterial. Darüber hinaus führten Erfahrungen mit dem üblichen Vorwissen der Studierenden aus der Unterrichtspraxis zu einer Entscheidung für das Format des Web-basierten Trainings. Das Web-basierte Training überlässt die Taktung des Lernfortschrittes weitgehend den Lernenden und setzt verstärkt auf die selbstständige Auseinandersetzung mit dem Lernmaterial. Die fachliche Herkunft der am Kurs Interessierten reicht von mathematisch oder logisch orientierten Studierenden über Informatiker, Philologen und Studierende der theoretischen Sprachwissenschaft bis hin zu Studierenden der kognitiven Psychologie. Beliebige Kombinationen der genannten Fächer zusammen mit gegebenenfalls weiteren Fächern sind nichts Ungewöhnliches. Für ein derart heterogenes Publikum ist eine Möglichkeit individueller Schwerpunktsetzungen und alternativer Lernpfade durch das Lernmaterial wünschenswert. Gerade diese Flexibilität kann freilich in einem Präsenzseminar kaum angemessen umgesetzt werden. Ein Präsenzseminar birgt die Gefahr, dass ein Teil der Studierenden zeitweise gelangweilt abschaltet, wenn für andere Studierende essentielle Wissensgrundlagen eingeführt werden müssen. Zugleich können Novizen Schwierigkeiten haben, dem Tempo der Veranstaltung zu folgen.

Die Anzahl der Teilnehmer an Präsemzseminaren in dem beschriebenen Themenumfeld ist gerade aufgrund ihrer Schnittstellenposition zwischen verschiedenen Wissenschaftsdisziplinen üblicherweise gering, sofern es sich nicht um eine Pflichtveranstaltung eines Studiengangs handelt. Vor diesem Hintergrund stellt ein mehrere Universitäten übergreifendes Web-basiertes Angebot in mehrfacher Hinsicht eine Ressourcenoptimierung dar: Es wird insgesamt eine größere Anzahl an Studierenden erreicht; aufgrund des Austauschs des Kursangebots zwischen den Universitäten wird das Curriculum der beteiligten Universitäten inhaltlich aufgewertet und um attraktive Komponenten erweitert, wobei die besonderen Stärken der verschiedenen Standorte genutzt werden. Der nicht standortgebundene Charakter eines Web-basierten Kurses trägt entscheidend zur Rentabilität des im Folgenden noch näher beschriebenen Aufwands seiner Erstellung bei und sichert einen breiteren Einsatz des Kursmaterials.

2.1 Zu Projektbeginn vorliegende Materialien

Die Planung des Projektes war von Anfang an darauf ausgerichtet, Material aus früheren Lehrveranstaltungen durch ein elektronisches Lehrbuch und neue Softwarekomponenten mit Glossaren und Handbüchern so zu erweitern und aufzubereiten, dass Studierende und Interessierte in Zukunft in der Lage sein würden, sich die mathematischen, linguistischen und computationellen Grundlagen der Grammatikentwicklung und Grammatikimplementierung in der HPSG im Rahmen des Web-based Training durch die Arbeit mit interaktiver Lernsoftware weitestgehend selbstständig anzueignen. Dabei konnte in großem Ausmaß auf in Vorläuferkursen gesammelte Lehrerfahrungen, auf konkrete Aufgabenstellungen und auf vorstrukturiertes Lehrmaterial zurückgegriffen werden.

Von sprachwissenschaftlicher Seite diente eine Einführungsveranstaltung in die HPSG als Vorlage, die zu den regelmäßigen Lehrveranstaltungen am SfS zählte und sich am Buch von Pollard and Sag (1994) orientierte. Eine Besonderheit dieser Veranstaltung bestand darin, dass sie mit einer kompakten, nicht-technischen Einführung in die Merkmalslogik begann, in der Studierende Gelegenheit hatten, Modelle kleinster Grammatiken aus Styroporbällen und Draht zu konstruieren.

Darüber hinaus bot das SfS im Rahmen seiner Seminare zu mathematischen Methoden in der Sprachwissenschaft in unregelmäßigen Abständen Seminare an, welche die Eigenschaften der für die HPSG einschlägigen Merkmalslogiken zum Gegenstand hatten. Daneben wurden von Lehrenden des SfS drei Kurse zu diesem Thema an internationalen Sommerschulen unterrichtet. Auch hier lagen Vorlesungsskripte vor, deren Inhalte in das Web-basierte Seminar einflossen.

Am reichhaltigsten war das Material zur Einführung in Techniken der Grammatikimplementierung. Vorläufer war hier eine Reihe von Kursen in *Constraint-basierter Grammatikimplementierung*, die über mehrere Jahre an internationalen Sommerschulen und an der Universität Tübingen in erster Linie von W. Detmar Meurers und Erhard Hinrichs gehalten wurden. Diese Kurse steuerten eine erste Strukturierung der Lehrinhalte, eine Serie von stufenweise komplexeren Lehrgrammatiken, Ideen für geeignete Übungsaufgaben zur Grammatikimplementierung und reichhaltige Vorlesungsfolien bei.

Obwohl die Grammatikimplementierungskurse mit einer Ausnahme auf ConTroll (Götz, 2000), einem Vorläufersystem von TRALE, aufbauten, konnten doch viele Grammatiken und Aufgabenstellungen, die ursprünglich für ConTroll entworfen wurden, mit geringem Aufwand für die Verwendung im TRALE-System angepasst werden. TRALE selbst war in der letzten Phase eines von Erhard Hinrichs geleiteten HPSG-Projektes am Sonderforschungsbereich 340 (SFB 340) in den Jahren 1998 bis 2000 am SfS unter Federführung von Gerald Penn als ein Nachfolgesystem zu ConTroll und eine Erweiterung von ALE entwickelt worden, das möglichst große Nähe zur theoretischen HPSG mit computationeller Effizienz zu verbinden suchte.

2.2 Unterauftragnehmer

In einem Lernkontext müssen natürlicher Weise ganz andere Anforderungen an eine Grammatikimplementierungsplattform gestellt werden als in einem reinen Forschungskontext, in dem die Benutzerfreundlichkeit und in mancher Hinsicht auch die Robustheit von Programmen nicht notwendigerweise im Vordergrund steht. Sofern Benutzerhandbücher für Forschungssoftware überhaupt geschrieben werden, sind sie für ein Publikum gedacht, welches die wissenschaftlichen Grundlagen und den Zweck der Programme hervorragend kennt. In einem Web-basierten Kurs müssen dagegen an Lernsoftware hinsichtlich Benutzerfreundlichkeit, Robustheit, interaktiver Hilfefunktionen und einführender Handbücher besonders strenge Maßstäbe angelegt werden. Insbesondere gewinnt die Programmdokumentation für diejenigen Benutzer, die nicht Experten sind, einen hohen Stellenwert. Daraus resultiert notwendig ein anderer Aufbau der Benutzerhandbücher. Insofern musste eine umfangreiche Anpassung des TRALE-Systems für die Bedürfnisse des neuen

Benutzerkreises im neuen Verwendungskontext vorgenommen werden, die insbesondere die Robustheit, den Aufbau der Benutzerhandbücher und die Dokumentation betraf. Diese Arbeiten konnten in dem durch die Laufzeit des Projektes vorgegebenen Zeitrahmen nicht alleine in Tübingen durchgeführt werden.

Die natürlichen Unterauftragnehmer zur Weiterentwicklung der Software waren W. Detmar Meurers von der Ohio State University und Gerald Penn von der University of Toronto, da sie als ehemalige Mitarbeiter von Tübinger Projekten des SFB 340 die Autoren von Vorläuferkursen beziehungsweise der an die Bedürfnisse des Kurses anzupassenden Software waren. Die Projektarbeit konnte somit auch auf langjährige gemeinsame Arbeit aufbauen. Auf der anderen Seite stellte die transatlantische Konstellation der Zusammenarbeit eine weitere Herausforderung an die Projektplanung und insbesondere an den Informationsfluss innerhalb des Projektes dar.

3 Softwarefragen

Bei der Erstellung des Web-basierten Kurses *Grammar Formalisms and Parsing* standen der Natur der Sache gemäß über weite Strecken verschiedenste Softwarefragen im Vordergrund. Diese erstreckten sich von Betriebssystemfragen über die Programmierung in Java und das Verhalten von Javaprogrammen beim Aufruf von verschiedenen Browsern und auf verschiedenen Betriebssystemen sowie die Darstellung von Zeichensätzen in verschiedenen Browsern bis hin zum Verhalten der Programme in Netzwerken oder der Erstellung des Lehrmaterials in der vom MiLCA-Konsortium gewählten E-learning Plattform. Im Folgenden betrachten wir einige dieser Fragen etwas genauer.

3.1 Betriebssysteme, Standards und Internetsoftware

Die Entwicklung des Kurses stand im Spannungsfeld der Vor- und Nachteile verschiedener Betriebssysteme und der auf verschiedenen Betriebssystemen verfügbaren, für die unterschiedlichen Bestandteile des Kurses benötigten oder zumindest arbeitserleichternden Software. Eckpunkte waren dabei die Weiterentwicklung der Grammatikimplementierungsplattform TRALE und des grafischen Benutzerinterfaces GRISU unter Solaris (5.7 und 5.8) beziehungsweise (RedHat) Linux, da dies die Betriebssysteme der relevanten Netzwerke in Tübingen, Columbus und Toronto waren, auf denen sämtliche Vorarbeiten geleistet worden waren und auf denen jeweils die neusten Versionen des lizenzpflichtigen SICStus Prolog sowie anderer benötigter Compiler und Werkzeuge installiert waren.[2] Aufgrund der Arbeitsplatzausstattung war es somit notwendig, dass die Arbeiten unter Solaris und Linux durchgeführt wurden. Die Computerpools der Studierenden verfügten über eine entsprechende Ausstattung.

Auf der anderen Seite stand der Bedarf an multimedialen und vor allen Dingen interaktiven Inhalten des elektronischen Lernmaterials. Damit verbunden war die konsortiumsweite Entscheidung zugunsten der einheitlichen Verwendung der E-learning Plattform ILIAS[3] und die Erstellung des elektronischen Lehrbuchs auf XML-Basis. Beides wäre

[2]TRALE wird unter SICStus Prolog entwickelt. Informationen zu TRALE finden sich unter `http://www.sfs.uni-tuebingen.de/hpsg/sysen.html`.

[3]`http://www.ilias.uni-koeln.de/ios/index.html`

von verbreiteter Software alternativer Betriebssysteme besser unterstützt worden. Diesen Sachverhalt mögen zwei Beispiele illustrieren.

Zur Arbeit mit XML und XHTML (siehe auch Lobin et al. (2003)) gibt es unter Windows geeignete Entwicklungsumgebungen mit passenden Werkzeugen, welche für die Dokumentenbearbeitung nützlich sind und die Arbeit gerade mit Dokumenten der hier einschlägigen Größe erheblich erleichtern. Unter den uns zur Verfügung stehenden Betriebssystemen konnten wir keine derartigen Werkzeuge verwenden. Stattdessen mussten alle Arbeiten direkt am Sourcecode im Emacs-Editor oder mit dem viel schlichteren Editor der ILIAS-Plattform selbst durchgeführt werden. Diese Vorgehensweise war ungemein zeitraubend, insbesondere angesichts der Tatsache, dass verschiedene Optionen der Umsetzung grafischer und interaktiver Bestandteile in der Anfangsphase erst exploriert werden mussten. Alleine die Definition, Erstellung und Verwaltung dieser Dokumente beanspruchte nahezu die gesamte bezahlte Arbeitszeit der voll angestellten Mitarbeiterin des Projektes.

Entsprechendes gilt für Flash-Animationen, deren Verwendung vor allem für den theoretisch-sprachwissenschaftlichen Bestandteil des Kurses in größerem Umfang wünschenswert gewesen wäre und zu deren möglicher Verwendung es eine Reihe von Ideen gab. Die einzige Möglichkeit, Flash-Animationen zu erstellen, bestand jedoch auf privaten Rechnern, und von dieser Möglichkeit der Freizeitgestaltung wurde aus naheliegenden Gründen kaum Gebrauch gemacht. Infolgedessen wurde lediglich eine einzige Idee einer Flash-Animation umgesetzt.[4]

Ein weiterer großer Problembereich war die Verwendung von Javaprogrammen auf verschiedenen Betriebssystemen. Aufgrund der Tatsache, dass Java-Programme aus Browsern unter der Verwendung von weiteren Web-Ressourcen gestartet werden können, wurde diese Programmiersprache zur Implementierung eines neuen Lernprogramms gewählt, dem eine zentrale Position innerhalb der Kurskonzeption zukommt, indem es linguistische Theoriebildung, mathematische Grundlagen und Grammatikimplementierung miteinander verbindet und auf neue Weise visualisiert. Der Name des Programms lautet *Morph Moulder*, kurz MoMo. Es wurde federführend von Ekaterina Ovchinnikova geschrieben. Die wesentliche Idee von MoMo besteht darin, eine Klasse von formalen Sprachen bereitzustellen, mit denen sich HPSG-Grammatiken vollständig notieren lassen. Die Sprachen stellen syntaktisch eine echte Obermenge der Sprachen dar, welche TRALE zur Verfügung stellt. Im Unterschied zu Parsingsystemen kann MoMo jedoch die Bedeutung von Mengen von logischen Ausdrücken, die in Mengen von Merkmalsstrukturen besteht, grafisch darstellen. Logische Funktionen des Programms erlauben es, die Wohlgeformtheit von Ausdrücken und von Merkmalsstrukturen zu explorieren, die Bedeutung von Ausdrücken relativ zu Mengen von Merkmalsstrukturen zu erforschen, und letzten Endes die Bedeutung einfacher Grammatiken vollständig zu visualisieren.[5]

In unserem gegenwärtigen Zusammenhang ist nun die angebliche Plattformunabhängigkeit von Javaprogrammen von Interesse. Sie stellt zunächst fraglos eine Eigenschaft von Java dar, die seine Verwendung attraktiv macht. Wie sich aufgrund der drei verschiedenen Betriebssysteme am SfS und des Umstandes, dass einige Studierende MoMo zu Hause auf ihren Windows-PCs installieren wollten, recht schnell herausstellte, ist jedoch

[4]Es handelt sich dabei um eine interaktive Darstellung der gesamten Sortenhierarchie und der Appropriateness-Bedingungen der Signatur der Grammatik von Pollard and Sag, 1994.

[5]Zum Aufbau und zu den Funktionen von MoMo siehe Richter et al., 2002.

diese Plattformunabhängigkeit so lange eine lediglich theoretische Größe, bis man einige Erfahrungen hinsichtlich des unterschiedlichen Verhaltens von Java sowohl unter verschiedenen Betriebssystemen und zusammen mit verschiedenen Browsern als auch hinsichtlich der Unterschiede zwischen verschiedenen gängigen Versionen von Java gesammelt hat, die Studierende wahrscheinlich verwenden werden. Hier erwies es sich letzten Endes als unabdingbar, sich mit den Gegebenheiten auf allen gängigen Betriebssystemen vertraut zu machen und MoMo im Laufe seiner Entwicklung regelmäßig unter den verschiedensten Betriebssystem- und Versionsbedingungen zu testen, um die Lauffähigkeit zumindest unter den wahrscheinlich häufigsten Betriebsbedingungen sicherzustellen. Diese Vorgehensweise schien aus Gründen der Robustheit und Einsatzfähigkeit in der Studienpraxis unumgänglich, da die Verwendung der Software nicht zu rein technisch bedingten Schwierigkeiten auf Seiten der Studierenden führen darf.

Analoge Schwierigkeiten ergaben sich bei der Kodierung des elektronischen Lernmaterials in ILIAS, wobei das Grundproblem unabhängig von der hier verwendeten Lernplattform ist. Ursprung des Problems ist, dass Browser erst schrittweise im Laufe ihrer Entwicklung in der Lage sind, internationale Zeichensätze gemäß existierender Kodierungsnormen zu erkennen und visuell umzusetzen. Während alle Browser den Zeichensatz des lateinischen Alphabets beherrschen und in der Regel auch über Umlaute verfügen, werden die Fähigkeiten verschiedener Browser immer geringer, je weiter man sich zum Beispiel in der Verwendung mathematischer Zeichensätze vorwagt. Hier gibt es teilweise browserspezifische Lösungen, oder der Benutzer muss sich mit einer Einbindung der gewünschten Zeichen in Form von Bildern behelfen, die wiederum ihre eigenen Schwierigkeiten mit sich bringt. So haben Bilder beispielsweise eine statische Größe, und ihre exakte Anordnung im Textfluss ist wiederum browserabhängig. Gemeinsam sind all den Behelfsmöglichkeiten, dass sie keinesfalls so generell sein können wie eine Kodierung der fraglichen Zeichen gemäß der internationalen Normen.

Für den praktischen Einsatz des elektronischen Lernmaterials wäre es selbstverständlich völlig unsinnig gewesen, noch so wohlgemeinte internationale Normen zu befolgen, sofern kein einziger existierender und den Studierenden leicht zugänglicher Browser sie auch tatsächlich grafisch korrekt umzusetzen in der Lage ist. Es musste also für alle über das lateinische Alphabet hinausgehenden Zeichen (angefangen bei griechischen Buchstaben) stets erst überprüft werden, ob die Normen von Browsern bereits verstanden wurden oder noch nicht, und je nach Befund wurde über ihre Kodierung im elektronischen Lernmaterial entschieden.

Die geschilderte Situation resultierte darin, dass für sämtliche Kodierungsentscheidungen, welche die E-learning Plattform ILIAS betrafen, stets eine ganze Reihe von System- und Softwarekombinationen getestet wurde. Unsere getesteten Betriebssysteme waren MacIntosh OS, neuere Windowsversionen, Solaris 5.7 und 5.8 und neueste Versionen von RedHat Linux. Unter den Web-Browsern berücksichtigten wir die neusten Mozilla, verschiedene Netscapes, Internet Explorer und auf Linux den Konqueror. Wir konnten dadurch zwar nicht gewährleisten, dass unter allen Kombinationen stets eine optimale Darstellung und alle Funktionen verfügbar waren. Aber wir konnten doch immerhin die wichtigsten Grundlagen auf allen Plattformen garantieren und die Studierenden im Vorfeld beraten oder warnen. Der investierte Zeitaufwand wurde somit zumindest durch das Ausbleiben böser Überraschungen bei der Verwendung der Materialien in der Lehre belohnt.

3.2 Technische Grenzen

Trotz der im vorigen Abschnitt beschriebenen Maßnahmen gab es mehrere Punkte, an denen die Entwicklung des elektronischen Lehrmaterials aufgrund technischer Grenzen nicht soweit vorangetrieben werden konnte, wie es die ursprüngliche Projektplanung vorgesehen hatte.

Besonders herausstechend ist dabei auch unter dem Aspekt der späteren Pflege der Kursmaterialien die bereits erwähnte Frage der Kodierung mathematischer Symbole und anderer Sonderzeichen. Da gegenwärtige Browser die im Prinzip bereits existierenden Standards nicht oder nicht in ausreichendem Maße umsetzen, musste aufgrund praktischer Notwendigkeiten in der Lehre hier auf die standardisierte, einheitliche Kodierung verzichtet und stattdessen auf eine sehr aufwändige Lösung durch Einbindung von Bildern zurückgegriffen werden. Diese Lösung muss visuell unbefriedigend bleiben, zerstört die Einheitlichkeit der Kodierung und wird für die Pflege der Daten in Zukunft ein Problem darstellen, sofern nicht (finanzielle und personelle) Mittel gefunden werden, sie rückgängig zu machen, sobald fortgeschrittenere Browser verfügbar werden. Unsere Strategie war einstweilen die, all diejenigen Zeichen bereits gemäß internationaler Standards zu kodieren, die Mozilla in dieser Form erkennt und darstellen kann. Mozilla wurde zu unserem Standard, da er der in dieser Hinsicht am weitesten entwickelte Browser ist, der Studierenden leicht für alle einschlägigen Betriebssysteme zugänglich ist und dessen eventuell notwendig werdende Installation wir für zumutbar halten.

Um in Zukunft bei der Benutzung des Lernmaterials nicht auf die aufwändige Pflege und für Ergänzungen nicht auf Ressourcen zur Erweiterung des elektronischen Materials angewiesen zu sein, wird als Konsequenz aus den Entwicklungserfahrungen seit dem Sommersemester 2003 der Lerntext neben seiner ILIAS-basierten Fassung auch in Latex formatiert. Die Pflege dieser Version ist weniger personalintensiv und weniger anfällig für Veränderungen der Softwarebasis. Außerdem eignet sie sich besser zum Informationsaustausch mit Fachkollegen und kann gegebenenfalls durch Konvertierung ins PDF-Format und Einbindung von Links weitgehend webfähig gemacht werden.

Eine gewisse Beschränkung der eigentlich gewünschten Mobilität des Kurses resultiert aus der Verwendung des TRALE-Systems, da TRALE im Augenblick nur mit dem lizenzpflichtigen SICStus Prolog betrieben werden kann.[6] Studierende, die TRALE auf dem eigenen Rechner installieren möchten, sind somit gezwungen, selbst eine teure Lizenz zu erwerben. Eine in Erwägung gezogene servergestützte Lösung wurde aufgrund erheblicher Sicherheitsbedenken und der Größe der zu kommunizierenden Datenpakete verworfen. Sie hätte es den Kursteilnehmern erlaubt, TRALE grundsätzlich aus dem Browser heraus zu starten und auf einer zentralen Maschine zu benutzen. Bei Vorliegen einiger Voraussetzungen auf Seiten des Betriebssystems der Studierenden ist allerdings auch bei der gegenwärtigen Architektur eine Verwendung von TRALE über das Internet möglich: Einige Studierende mit schnellen Internetverbindungen konnten in der Vergangenheit bereits TRALE problemlos von zu Hause aus auf Universitätsrechnern verwenden. Unter günstigen Rahmenbedingungen ist eine Teilnahme am Kurs ausschließlich von einem universitätsexternen Rechner aus also prinzipiell möglich. Generell ist es jedoch im Augenblick so, dass jede am Kursangebot teilnehmende Universität in einem Computerpool SICStus Prolog installieren muss.

[6] Eine Anpassung von TRALE auf frei Verfügbare Prolog-Versionen ist vorgesehen, kann jedoch im Moment nicht geleistet werden.

4 Wachstumskurven

Ein Projekt zur Entwicklung eines virtuellen Unterrichtskurses wie das MiLCA-Modul *Grammar Formalisms and Parsing* hat kaum Vorlaufzeiten in der Entwicklung, sondern muss sich bald nach seinem Start in der Unterrichtspraxis bewähren. Zum ersten Mal wurde das Seminar rund ein dreiviertel Jahr nach Aufnahme der Arbeit gelehrt.

Bei der Planung des Kurses war zu berücksichtigen, dass verschiedene Faktoren ein Web-basiertes Kursangebot gleich zu Beginn des Projektes ausschlossen. Dazu zählten der Zeitbedarf für die Einarbeitung in die Funktionen und Verwendung der Lernplattform ILIAS, die notwendige Aufbereitung des Benutzerinterfaces und der Dokumentation von TRALE und die Programmierung von MoMo. Für die Planung der Kursentwicklung und den Einsatz des Materials in der Lehre über vier Semester hinweg (Sommersemester 2002 bis Wintersemester 2003/04) musste exakt geplant werden, zu welchem Zeitpunkt welche Bestandteile des Kurses in einem bestimmten Lehrformat benötigt würden und welchen Grad der selbstständigen Verwendung die Materialien ohne eine stets zur Hilfe und zur Erklärung bereitstehende Lehrkraft zulassen müssten. Die anvisierten Kursformate reichten vom computergestützten Präsenzseminar bis zum Web-basierten Training.

Der Zeitplan für die schrittweise Überführung des Lehrmaterials in elektronische Medien, die ein selbstgesteuertes Lernen unterstützen, war wiederum in Abhängigkeit zu sehen von der sehr knapp kalkulierten Finanzsituation des Projekts. Sie sah lediglich eine Mitarbeiterin für einen Teil der Laufzeit vor, sowie eine studentische Hilfskraft, was angesichts der notwendigen Maßnahmen keinen selbsttragenden Finanzrahmen bilden konnte. In den folgenden beiden Unterabschnitten beschäftigen wir uns mit den wichtigsten Aspekten der stufenweisen Erstellung eines tatsächlich Web-basierten Kurses während des laufenden Lehrbetriebs.

Bei allen Nachteilen, die durch die Parallelität von Einsatz und Entwicklung des Kurses entstanden sein mögen, sind aber auch stets die Vorteile zu berücksichtigen, die sich durch die sofortige Rückmeldung seitens der Studierenden einstellten. So manche Grundfunktion von MoMo verdankt ihr Dasein den früh festgestellten und geäußerten Bedürfnissen der Studierenden, die in einer Präsenzveranstaltung viel früher und frustrationsärmer ermittelt werden konnten und somit wirkliche Grundlagen des Designs des Interfaces beeinflussten. In der Praxis führte das anfänglich zur Entwicklung von einer Serie von Prototypen, die im Rhythmus der Seminarsitzungen entstanden. Dies vermittelte den Studierenden das Gefühl, dass sie in die Entwicklung eingebunden waren und dass ihren Kommentaren Gewicht beikam, was zur spürbaren zusätzlichen Motivation der Studierenden beitrug.

4.1 Vom Präsenzseminar zum Web-basierten Training

Die erste Lehrveranstaltung neun Monate nach Projektbeginn (Sommersemester 2002) war ein klassisches Präsenzseminar mit verstärktem Softwareeinsatz, also mit einem großen Anteil an Arbeit im Computerlabor. Hier konnte eine Neustrukturierung und Anpassung der zu Projektbeginn vorliegenden, reichhaltigen Lernmaterialien getestet werden. Die Entwicklungsschwerpunkte bei der Software orientierten sich am Seminarverlauf. Die neuen Tischvorlagen, die in diesem Kurs entstanden, dienten später als erste Bausteine des elektronischen Lehrbuchs.

Da das TRALE-System an sich bereits funktionsfähig war, wurden auf der Grammatikimplementierungsseite hinsichtlich des Einsatzes im Seminar für das Sommersemester 2002 keine Neuerungen angestrebt, sondern es wurden lediglich existierende Lehrgrammatiken von ConTroll-basierten Lehrveranstaltungen für TRALE adaptiert. Dadurch ergab sich die Möglichkeit, auf der Basis des alten Systems zu planen und seine grundlegende Überarbeitung durch Gerald Penn und W. Detmar Meurers vom laufenden Seminarbetrieb zu entkoppeln. Stattdessen lag der Schwerpunkt in dieser Phase hinsichtlich der laufenden Lehre auf der Entwicklung einsatzfähiger Prototypen von MoMo. Aufgrund des Kursformats als Präsenzseminar konnte in dieser Phase außerdem auf eine Dokumentation des Programms und auf Hilfefunktionen verzichtet werden. So gab es zu diesem Zeitpunkt keinerlei komfortable Eingabefunktionen. Auf diese Weise war es möglich, hinsichtlich der für die Lehre benötigten inhaltlichen Funktionalitäten der Software mit dem Fortgang des Seminars Schritt zu halten.

Die dem Präsenzseminar folgende vorlesungsfreien Zeit bot Gelegenheit, die gesammelten Erfahrungen auszuwerten, mit zusätzlichen Programmierkräften MoMo in einen robusten Zustand zu überführen und mit komfortablen, auf den Verwendungszweck zugeschnittenen Benutzerschnittstellen auszustatten, sowie die Überarbeitungen des TRALE-Systems und insbesondere seiner Benutzerschnittstelle soweit voranzutreiben, dass sowohl MoMo als auch TRALE ohne unmittelbar gegenwärtige und stets verfügbare Betreuung benutzbar wurden.

Die notwendig gedrängte Zeitplanung stellte im Wintersemester 2002/03 die Herausforderung, das elektronische Lehrbuch parallel zum laufenden Seminarbetrieb zu schreiben und in ein angemessenes Format zu überführen. Durch die Erfahrungen mit dem vorangegangenen Präsenzseminar konnte der Verlauf des Kurses exakt im Voraus geplant werden. Um eine realistische Chance zu haben, das elektronische Lehrmaterial in der zur Verfügung stehenden Zeit zu kreieren, wurde die erste tatsächlich computergestützt unterrichtete Version des Kurses von insgesamt drei Personen betreut.

Den Kurs im Wintersemester 2002/03 belegten Studierende der Universitäten Tübingen und Bonn, so dass ein Ausweichen auf das Format des Präsenzseminars in keiner Weise möglich war. Für zwei festgelegte Tage der Woche musste Kursmaterial im Umfang von jeweils zehn bis zwanzig Seiten produziert und in die ILIAS-Plattform integriert werden. Realisiert wurde dies durch die Verteilung der Arbeit vor Ort in Tübingen auf vier Personen: Einerseits wurde MoMo aufgrund der stets aktuellen Erfahrungen weiterentwickelt; eine Person schrieb das Lernmaterial im Latex-Format; eine weitere Person leistete technische Betreuung der Studierenden, besorgte die Umformatierung des Textes und vor allem der Graphiken und Links in ILIAS; und eine Person übernahm die Korrespondenz mit den Studierenden und die Durchsicht der wöchentlich zwei Mal eingereichten Hausaufgaben.

Die erste Veranstaltung, die in einer dem Zielformat angenäherten Form gehalten werden konnte, fiel auf das Sommersemester 2003. Zu diesem Zeitpunkt lag bereits eine in den relevanten Aspekten erheblich verbesserte Grammatikimplementierungsumgebung vor, und es gab benutzbare Versionen der Handbücher für TRALE und MoMo. Die Verzahnung von MoMo mit den elektronischen Kursmaterialien hatte begonnen, indem Übungsaufgaben nun direkt aus dem Lehrmaterial heraus zugänglich waren. Die Entwicklungsarbeit konzentrierte sich in dieser Phase darauf, das Material im Tempo des Seminarfortschritts auf der Basis der Erfahrungen des vorangegangenen Semesters gründlich zu überarbeiten. Ab diesem Zeitpunkt wurden die Kursmaterialien zudem synchron im PDF-Format und im elektronischen Format der ILIAS-Plattform weitergeführt.

Im Wintersemester 2003/04 sollen im abschließenden Schritt die Handbücher und Dokumentationen des TRALE-Systems mit dem bestehenden Lernmaterial enger verzahnt werden, die nahtlose Einbindung von MoMo in die Lernumgebung wird abgeschlossen und MoMo um eine interaktive Hilfefunktion erweitert, und neuere Funktionen von TRALE werden in die Lehre und das Lernmaterial integriert werden. MoMo wird dann nicht nur aus dem Browser heraus direkt mit den jeweiligen Übungsaufgaben des elektronischen Lehrbuchs heraus gestartet werden können, sondern die Texte, die das grafisch dargebotene unmittelbare Feedback des Programms erläutern, werden mit einem Stichwortglossar im Lernmaterial verlinkt sein, das zusätzliche Erklärungen und Begriffsdefinition bietet oder aus dem Lehrtext wiederholt. Außerdem ist daran gedacht, alternative Lernpfade für Studierende mit unterschiedlichen Vorkenntnissen anzulegen, um individuell getakteten Lernfortschritt zu ermöglichen.

4.2 Personalaufwand

Der Personalaufwand, der zur Umsetzung des Kurses getrieben werden musste, war von Beginn an hoch. Von entscheidender Bedeutung für die Einschätzung des tatsächlichen Aufwands ist unter dem Strich folgende Gegenüberstellung: Auf dem Papier verfügte das Projekt in Tübingen gemäß seinem Budget im Zeitraum von September 2001 bis einschließlich Juni 2003 über eine reguläre Mitarbeiterin, Beata Trawiński, sowie über eine studentische Hilfskraft, Ekaterina Ovchinnikova. Außerdem kommen die beiden Unterauftragnehmer in Columbus und Toronto hinzu. Tatsächlich kann man die Zahl der wesentlich zur Gestaltung des Kurses an seinen drei Standorten Beitragenden jedoch auf 16 Personen hochrechnen. Nach ihrem Beschäftigungsverhältnis aufgeschlüsselt handelt es sich dabei um zwei Professoren, vier Wissenschaftliche Mitarbeiter, einen Lehrstuhlassistenten, drei Studentische Hilfskräfte und sechs Doktoranden. Es ist klar, dass dieser Personaleinsatz nur durch beträchtliche technische und finanzielle Unterstützung von (projekt-) fremder Seite möglich war, welche die Gelder des Projektträgers ergänzte.

5 Bewertungen

5.1 Akzeptanz bei den Studierenden

Die Akzeptanz des Kurses bei den Studierenden war durchweg äußerst ermutigend. Die Teilnehmerzahl lag bei den bisherigen Angeboten konstant bei sieben, wobei die Zahl der anfänglichen Anmeldungen zu jedem Kurs stets deutlich höher war. Einige der Angemeldeten erschienen niemals im virtuellen Seminarbetrieb; ein Drittel bis zur Hälfte der ursprünglich Angemeldeten entschied sich regelmäßig in den ersten zwei bis drei Wochen, den Kurs abzubrechen. Dies entspricht durchaus den üblichen Erfahrungen bei Präsenzveranstaltungen, die nicht obligatorisch im Studienplan vorgeschrieben sind und stellt somit keine Besonderheit dieses Kurses dar.

Die Tatsache, dass das Kursmaterial auf Englisch vorliegt, erwies sich bei den Teilnehmern nie als erschwerender Faktor. Die meisten Studierenden machten nicht einmal von dem Angebot Gebrauch, Lösungen zu Übungsaufgaben auf Deutsch einzureichen, sondern formulierten sie in englischer Sprache. Weit entfernt davon, sich erschwerend auszuwirken, ermöglichte es die Verwendung des Englischen als Kurssprache einer Reihe von Austauschstudenten, deren Deutschkenntnisse nicht zur Teilnahme an einem deutschsprachigen Kurs ausgereicht hätten, den Kurs zu belegen.

In die gleiche Richtung weisen Erfahrungen mit der durch die Wahl der Sprache gegebenen Einsatzfähigkeit des Kursmaterials auf internationalen Lehrveranstaltungen und dem Austausch mit Fachkollegen. So wurden Teile des Lehrtextes, sowie kommentierte Lehrgrammatiken des Kurses im Oktober 2002 auf einer deutsch-israelischen Herbstschule eingesetzt, und MoMo erweist sich auf Kongressen als hilfreiches Werkzeug für illustrierende Beispiele. Die konsequente Verwendung des Englischen fördert somit die Nachhaltigkeit erheblich, ohne dass dabei der Einsatz in der Lehre erkennbar behindert würde.

Die Evaluation des Kurses durch die Teilnehmer ergab, dass insbesondere die individuelle Betreuung der Hausaufgaben geschätzt und als hilfreich empfunden wird. Im Vergleich zu traditionellen Präsenzseminaren im gleichen Themengebiet scheint zugleich eine größere Menge an Lehrstoff im gleichen Zeitraum behandelbar zu sein. Durchweg positiv beurteilt wurden die Gestaltung der WWW-Seiten und des Lernprogramms, die Aufbereitung des Inhalts und die Art der gestellten Aufgaben. Als besonders beliebt erwies sich dabei das MoMo-Programm zur Visualisierung und grafischen Exploration der logischen Grundlagen der HPSG. Allerdings sind alle Bewertungen notwendig vorläufiger Natur, da nicht ausgeschlossen werden kann, dass bisher vor allem besonders motivierte Studierende am Kurs teilnahmen. Die Bewertung des Kurses könnte sich ändern, sobald er als Pflichtkurs gelehrt wird, an dem alle Studierenden einer Fachrichtung gleichermaßen teilnehmen müssen.

5.2 Fazit: Vom Möglichen und vom Unmöglichen

Welches Fazit lässt sich aus der Summe der vielfältigen Erfahrungen mit *Grammar Formalisms and Parsing* für die Entwicklung Web-basierter Kursangebote ziehen?

Die unmittelbaren Projektresultate sind durchaus erfreulich: Die Akzeptanz des Kurses in der Form, in der er bisher unterrichtet werden konnte, ist sehr gut, und die Einsatzmöglichkeiten des Kursmaterials, sowohl der Texte als auch der Software, sind vielfältig. Kollegen innerhalb und außerhalb Deutschlands haben mehrfach darum gebeten, sie ihnen zur Ansicht und auch zur Verwendung zur Verfügung zu stellen. Daneben gab es Anfragen mehrerer Studierender auch von Universitäten, die nicht am MiLCA-Konsortium beteiligt sind, die sich nach den Möglichkeiten einer Teilnahme am Kurs erkundigten. Insgesamt ist es fraglos so, dass die Zahl der Studierenden, die von dem Kurs profitieren, durch sein Format mittelfristig erheblich höher sein wird, als wenn die Inhalte lediglich für ein traditionelles Präsenzseminar aufbereitet vorlägen.

Ein weiterer Effekt des Projekts besteht darin, dass TRALE durch seine Weiterentwicklung zur Lernsoftware zugleich als Forschungssoftware gefördert wurde, da das System mit der erheblich verbesserten Dokumentation und dem zugehörigen Lernmaterial auch für Experten erheblich zugänglicher und benutzungsfreundlicher geworden ist. Eine denkbare Verbindung zwischen dem Parsingsystem TRALE und MoMo als Software zur Manipulation und Visualisierung von Modellen von Grammatiken eröffnet darüber hinaus weiterführende Forschungsperspektiven. So rücken aufgrund der Erfahrungen mit MoMo Erweiterungen der Constraint-Sprache von TRALE ins Blickfeld, und zukünftige

Debuggingwerkzeuge könnten von in MoMo erstmals angewandten Techniken profitieren.

Die Durchführung des Projektes zu *Grammar Formalisms and Parsing* bestätigte die bereits im Vorfeld gehegte Vermutung, dass eine qualitativ angemessene Überführung von Kursmaterial aus Präsenzseminaren in eine Form, die für das Web-basierte Training geeignet ist, äußerst personal- und kostenintensiv ist. Die tatsächlichen Kosten für das Projekt konnten nur aufgebracht werden, weil über die Finanzierung durch den Projektträger hinaus in beträchtlichem Umfang weitere Mittel eingesetzt wurden. Ein derartiger Aufwand wiederum kann nur bei einem nachfolgenden breiten, vielfältigen und, wo möglich, internationalen Einsatz des Produktes und gegebenenfalls von Bestandteilen desselben gerechtfertigt werden. Die relativ geringe Studierendenzahl an einer einzigen Universität wäre selbst unter Berücksichtigung einer qualitativen Verbesserung der Lehre dem getriebenen Aufwand kaum angemessen.

Von entscheidender Bedeutung für den Zeitaufwand für die Erstellung des elektronischen Lernmaterials ist die Qualität der Softwarewerkzeuge und die Frage ihrer Integration in die oder ihrer Kompatibilität mit der eingesetzten Lernumgebung. In dieser Hinsicht war die Wahl von ILIAS 2 zusammen mit Solaris äußerst unglücklich, da ILIAS 2 keine der wünschenswerten Werkzeuge bereitstellte und sie auch unter den verwendeten Betriebssystemen den Entwicklern nicht zur Verfügung standen. Für die Planung vergleichbarer Projekte ist nachdrücklich zu empfehlen, im Vorfeld genau zu prüfen, welche Lernplattformen und Datenformate für die verfolgten Zwecke am geeignetsten sind und den Personalaufwand am effektivsten mindern helfen.

Bereits in naher Zukunft ist allerdings damit zu rechnen, dass durch die fortschreitende Entwicklung von Lernplattformen und Browsertechnologie viele der oben beschriebenen Probleme erheblich abgemildert oder ganz beseitigt werden. Es könnte also durchaus sein, dass in wenigen Jahren vergleichbare Resultate mit erheblich geringerem Personalaufwand verbunden sind. Dies ist jetzt bereits unter der Annahme absehbar, dass bereits vorhandene Standards der Dokumentenkodierung durch die verbreitetsten Internetbrowser unterstützt werden und Lernplattformen wie ILIAS diejenigen Werkzeuge zur Dokumentenbearbeitung zur Verfügung stellen, welche wünschenswert sind und zum jetzigen Zeitpunkt erst selektiv und für einzelne Plattformen und Betriebssysteme existieren.

6 Danksagungen

Der Autor ist einer Vielzahl von Personen zu Dank verpflichtet. Hervorzuheben sind die Mitglieder des Tübinger Teams, Beata Trawiński und Ekaterina Ovchinnikova, sowie Gianina Iordăchioaia, Holger Wunsch und Carmella Payne; und die Leiter der externen Softwareprojekte, W. Detmar Meurers und Gerald Penn. Lothar Lemnitzer half bei allen aufkommenden Problemen; Adam Przepiórkowski kommentierte ausführlich das Lehrmaterial; Erhard Hinrichs unterstützte das Teilprojekt finanziell aus Lehrstuhlmitteln.

Literatur

Bresnan, Joan (Ed.) (1982). *The Mental Representation of Grammatical Relations.* Cambridge, MA: MIT Press.

Daniels, Mike and Meurers, W. Detmar (2002). Improving the Efficiency of Parsing with

Discontinuous Constituents. In S. Wintner (Ed.), *Proceedings of NLULP'02: The 7th International Workshop on Natural Language Understanding and Logic Programming*, Number 92 in Datalogiske Skrifter, Copenhagen, pp. 49–68. Roskilde Universitetscenter.

Gazdar, Gerald, Klein, Ewan, Pullum, Geoffrey, and Sag, Ivan (1985). *Generalized Phrase Structure Grammar*. Harvard University Press, Cambridge, Mass.

Götz, Thilo (2000). *Feature Constraint Grammars*. Phil. dissertation, Eberhard-Karls-Universität Tübingen.

Hinrichs, Erhard W., Meurers, W. Detmar, and Wintner, Shuly (Eds.) (2003). *Linguistic Theory and Grammar Implementation*. In press.

Lobin, Henning, Stührenberg, Maik, and Rehm, Georg (2003). eLearning und offene Standards: Zum Einsatz XML-strukturierter Lernobjekte. In diesem Heft.

Meurers, W. Detmar, De Kuthy, Kordula, and Metcalf, Vanessa (2003). Modularity of grammatical constraints in HPSG-based grammar implementations. In *Proceedings of the ESSLLI '03 workshop "Ideas and Strategies for Multilingual Grammar Development"*, Vienna, Austria.

Meurers, W. Detmar, Penn, Gerald, and Richter, Frank (2002). A Web-based Instructional Platform for Constraint-Based Grammar Formalisms and Parsing. In D. Radev and C. Brew (Eds.), *Effective Tools and Methodologies for Teaching NLP and CL*, New Brunswick, NJ, pp. 18–25. The Association for Computational Linguistics. Proceedings of the Workshop held at the 40th Annual Meeting of the Association for Computational Linguistics. 7.–12. July 2002. Philadelphia, PA.

Penn, Gerald (2003a). AVM Description Compilation using Types as Modes. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL-03), Budapest, Hungary.*

Penn, Gerald (2003b). Efficient Transitive Closure of Sparse Matrices over Closed Semirings. In *Proceedings of the Third Workshop on Algebraic Methods in Language Processing (AMiLP-3), Verona, Italy, August 2003.*

Penn, Gerald and Haji-Abdolhosseini, Mohammad (2003). Topological Parsing. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL-03), Budapest, Hungary.*

Penn, Gerald and Hoetmer, Kenneth (2003). In Search of Epistemic Primitives in the English Resource Grammar. In *Proceedings of the 10th International Conference on Head-Driven Phrase Structure Grammar, East Lansing, Michigan.*

Penn, Gerald and Munteanu, Cosmin (2003). A Tabulation-Based Parsing Method that Reduces Copying. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-03), Sapporo, Japan.*

Pollard, Carl and Sag, Ivan A. (1994). *Head-Driven Phrase Structure Grammar*. University of Chicago Press.

Richter, Frank, Ovchinnikova, Ekaterina, Trawiński, Beata, and Meurers, W. Detmar (2002). Interactive Graphical Software for Teaching the Formal Foundations of Head-Driven Phrase Structure Grammar. In G. Jäger, P. Monachesi, G. Penn, and S. Wintner (Eds.), *Proceedings of Formal Grammar 2002*, pp. 137–148.