

# Dokumentation von Projektarbeiten mit dynamischen Inhalten mittels Web 2.0-Werkzeugen

Michael Hielscher  
Zentrum für Bildungsinformatik  
Pädagogische Hochschule Bern  
mail@michael-hielscher.de

Christian Wagenknecht  
Fachbereich Informatik  
Hochschule Zittau/Görlitz  
c.wagenknecht@hs-zigr.de

**Abstract:** Projektarbeiten spielen in der Ingenieurausbildung an Hochschulen eine wichtige Rolle und sind oft auch Teil der Leistungsbewertung. Von den Studierenden wird dabei im Allgemeinen erwartet, dass sie nicht nur eine schriftliche Dokumentation einreichen, sondern auch zusätzliche, im Laufe der Projekte entwickelte Materialien, wie etwa Quellcode von Computerprogrammen in der Informatik oder Datenvisualisierungen durch 3D-Modelle im Umfeld von Architektur und Gestaltung. Die Begutachtung solcher dynamischer Inhalte ist für traditionell papiergebundene Belege nur sehr schwer zu leisten, denn Erprobung bzw. Ausführung und Annotation erfordern umfangreiche und zeitaufwändige Zusatzmaßnahmen. Häufig werden diese dadurch erschwert, dass die individuellen Arbeits- bzw. Entwicklungsumgebungen der Studierenden und Lehrenden nicht übereinstimmen. Durch geeignete Anpassung von Web 2.0-Werkzeugen ist es möglich, solche dynamische Inhalte in die Dokumentation von Projektarbeiten zu integrieren und in webbasierter Form zur Verfügung zu stellen. Die Präsentation der Ergebnisse, einschließlich dynamischer Elemente, und deren Evaluation und Kommentierung durch die entsprechende Lehrperson, können dann in ein und demselben Dokument erfolgen. Anhand von zwei Beispielen aus dem Bereich der Informatikausbildung wird das Konzept der Dokumentation von Projektarbeiten mit dynamischen Inhalten vorgestellt.

## 1 Einleitung

An Hochschulen und Schulen der Sekundarstufe II werden oft Projektaufgaben gestellt, die eine umfangreiche, sich über mehrere Wochen erstreckende computergestützte Bearbeitung erfordern. Projektaufgaben können dabei sowohl als Einzel- als auch als Gruppenarbeit angelegt sein. Im Bereich der Ingenieurwissenschaften bestehen diese Aufgaben meist aus der Entwicklung eines Produkts und einer entsprechenden schriftlichen Dokumentation des Entstehungsprozesses. In der Informatik ist das Zielprodukt meist ein kleines Softwaresystem, in der Elektrotechnik zum Beispiel ein Schaltungsentwurf, in der Mathematik die Lösung einer Aufgabe in Form eines Skriptes für ein Computer-Algebra-System oder in der Chemie eine 3D-Darstellung eines komplexen Moleküls.

Die Bearbeitung solcher Projektaufgaben durch die Studierenden findet heute in der Regel in der privaten studentischen Umgebung statt. Noch vor der inhaltlichen Beurteilung der Arbeit durch den Lehrenden scheitert in vielen Fällen die Erprobung der dynamischen Elemente (Programme, Visualisierungen, Berechnungen) auf Grund von technischen Schwierigkeit-

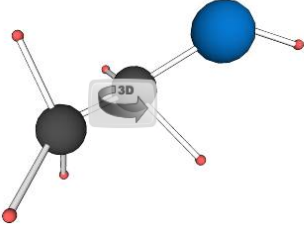
en. Häufig liegt das daran, dass die von den Studierenden und den Lehrenden verwendete Hard- und Software (Arbeitsumgebung) voneinander abweichen. Die z.B. im Rahmen einer Projektarbeit entwickelten Computerprogramme sind deshalb auf der Arbeitsplattform des Dozierenden oft nicht ohne aufwändiger Zusatzmaßnahmen lauffähig und damit überprüfbar. In der Regel lässt das Zeitbudget der Lehrpersonen eine intensive Analyse und Evaluation der Projektergebnisse auf Quelltextebene gar nicht zu. Im Fall der Informatik kann je nach Art des Softwareprojekts bereits die Vorbereitung zu einem groben Funktionstest sehr zeitaufwändig sein, wenn zunächst spezielle Softwarebibliotheken oder Server, welche die Studierenden für ihre Lösung verwendet haben, installiert bzw. konfiguriert werden müssen. Dies trifft aber nicht nur für die Informatik zu: Auch bei der Verwendung von Softwarewerkzeugen in anderen Fachbereichen wie Computer-Algebra-Systemen in der Mathematik oder generell Visualisierungswerkzeugen können Abhängigkeiten zu Hilfsdateien schnell zu Problemen bei der Sichtung der studentischen Projektergebnisse durch die Lehrenden führen. Die Dozierenden versuchen dann anhand des schriftlichen Belegteils das entwickelte System zu rekonstruieren, ohne das eigentliche Kernstück der Arbeit wirklich anzuschauen. Damit wird die für den Lernprozess der Studierenden erforderliche Auseinandersetzung und konstruktive Kritik der Arbeit behindert. Die Rückmeldungen an die Studierenden fallen dementsprechend wenig aussagekräftig aus. Sie beziehen sich nur auf die Aussagen in der schriftlichen Dokumentation und auf das sichtbare statische Endprodukt wie etwa die Darstellung eines 3D-Modells. Hinweise auf Schwachstellen im Programmcode einer Software, die sich oftmals aus dem Test von Extremfällen ergeben, oder zur Vereinfachung der Datenstruktur eines 3D-Modells bleiben aus. Dabei wären gerade solche Rückmeldungen zum Prozessverlauf in der Informatik, den Ingenieur- und Naturwissenschaften sehr wertvoll. Es bleibt nicht aus, dass sich die Studierenden auf diese Bewertungspraxis einstellen, und potenzielle Probleme bzw. bekannte Schwachstellen in der Lösung geschickt verbergen. In der Informatik zum Beispiel werden konzeptionelle Probleme bei der Softwareentwicklung von den Studierenden gern durch ein geschicktes Codekonstrukt umschifft, welches im schriftlichen Teil der Arbeit ganz bewusst undokumentiert bleibt.

Die durch unterschiedliche Arbeitsplattformen der Studierenden und Dozierenden verursachten technischen Probleme lassen sich bei der Nutzung geeigneter Web 2.0-Werkzeuge weitgehend vermeiden. Voraussetzung ist, dass dynamische Inhalte, wie die Verarbeitung von Programmcode oder Interpretation von Datenstrukturen für 3D-Modelle, in der Web-Plattform eingebettet und direkt im Browser ausgeführt werden können. Die Projektdokumentation ist nicht mehr mit der lokalen Arbeitsumgebung der Studierenden verknüpft, sondern wird beispielsweise auf einer Wiki-Plattform oder einem LMS abgelegt und kommentiert. Dies bietet verschiedene didaktische Mehrwerte: Die zentrale Ablage im Netz vereinfacht die Zusammenarbeit unter den Studierenden einer Projektgruppe und fördert das kollaborative Lernen. Dozierende können darüber hinaus jederzeit den aktuellen Arbeitsstand begutachten und kommentieren. Sie können Lernprozesse bekräftigen oder kritisieren. Gleichzeitig kann eine Projektgruppe die Fortschritte einer anderen Gruppen einsehen und sich an Diskussionen beteiligen. So können Studierende etwa bei Software-Entwicklungsprojekten direkt Einblick in den Programmcode bzw. die Resultate anderer Gruppen nehmen und sich gegenseitig unterstützen. Diese Option fördert die Kommunikation und soziale Kompetenzen der Studierenden. Wikis, wie auch Blogs, bieten di-

Feb 5 **Ethanol**
Alles Allgemein
RSS

**Aufgabenstellung:** Entwickeln Sie mit Hilfe von Google SketchUp ein dreidimensionales Modell des Ethanol Moleküls C<sub>2</sub>H<sub>6</sub>O und binden Sie Ihr Ergebnis hier ein. Beschreiben Sie anschließend zwei verschiedene Herstellungsverfahren. Wenden Sie ein Verfahren im Labor an und dokumentieren Sie Ihren Versuchsaufbau hier.

**Lösung:**  
3D-Modell des Moleküls:



Google 3D Warehouse

**Verfahren zur Herstellung:**  
Durch Destillation kann das Ethanol konzentriert werden. Auf diese Art gewonnene Getränke bezeichnet man als Spirituosen (z. B. Whisky, Cognac, Schnaps, Wodka oder Rum). Sie enthalten noch Aromen der verdampften Ausgangslösung, die zumeist ein Wein war. Liköre

**SEITEN**

- Eine Seite

**KATEGORIEN**

- Allgemein

**META-DATEN**

- Admin
- Abmelden
- Valid XHTML

Abbildung 1: Ethanol-Molekül und die Herstellung von Ethanol

verse Möglichkeiten zur Beobachtung der Aktivitäten, z.B. Watchlisten und RSS-Feeds, an. Durch die Verwendung solcher sog. Push-Verfahren lassen sich auch administrative Aufgaben effektiver bearbeiten.

Bislang gibt es jedoch keine befriedigende Möglichkeit, speziell für Lehrzwecke entwickelte dynamische Inhalte in Web 2.0-Plattformen einzubinden. Damit ist nicht etwa der Anhang einer Datei zum Download am Ende einer Wiki-Seite gemeint. Vielmehr geht es um die fließende Einbettung von dynamischen Inhalten, wie etwa von Java-Programmen, in eine statische Webseite. Ein solches Java-Programm sollte vom Betrachter der Seite direkt online ausgeführt und mit beliebigen Eingaben getestet werden können. Die angezeigten Resultate sind dann mit den Vorgabelösungen vergleichbar. Beinhaltet der dynamische Inhalt etwa eine Schaltungssimulation in der Elektrotechnik, sollte der Betrachter die Möglichkeit haben, die Simulation direkt in der Webseite auszuführen.

Abb. 1 zeigt eine fiktive Beispielaufgabe aus der Chemie. Es soll ein dynamisches dreidimensionales Modell eines Moleküls entwickelt und in eine Wordpress-Webseite eingefügt werden. Webbasierte Werkzeuge wie Google SketchUp eignen sich zwar besonders gut für die Einbettung von 3D-Modellen in Webseiten, bieten jedoch nicht die Funktionalität spezialisierter Molekülmodellierungssoftware für den Chemie-Unterricht.

In den folgenden Abschnitten wird die konkrete Realisierung eingebetteter dynamischer Inhalte in Wikis anhand zweier erprobter Beispiele aus dem Bereich der Informatikausbildung gezeigt. Die Beispiele stehen stellvertretend für andere Fachgebiete. Die zugrundeliegenden Technologien lassen sich auf viele andere Kontexte übertragen. Mit unserem Ansatz zeigen wir einen Weg auf, wie interaktive Elemente in Arbeitsumgebungen, wie etwa in Wikis, eingefügt werden können. Das mediendidaktische Potenzial von Wikis, siehe

z.B. in [Döb07], wird damit weiter ausgebaut.

## 2 Beispiel: Programming-Wiki

Programming-Wiki (vgl. [HW09]) ist eine Kombination aus einem herkömmlichen Wiki mit einer eingebetteten Entwicklungsumgebung, die keinerlei Installation zusätzlicher Software auf dem System des Nutzens erfordert. Dadurch kann ein Programmcode im Wiki nicht nur gelesen, sondern aktiv verändert, gespeichert und ausgeführt werden. Die Verarbeitung des Programmcodes findet dabei ausschließlich auf dem Client (im Webbrowser des Nutzens) statt. Das Programming-Wiki unterstützt verschiedene Programmiersprachen wie Java, JavaScript, Pascal, Prolog, Scheme oder auch SQL.

In Abb. 2 ist eine Programming-Wiki-Seite zu sehen. Im unteren Bereich erkennt man das eingebettete dynamische Element. Durch einen Klick auf die Schaltfläche „ausführen“ werden die darüber angegebenen (und editierbaren) SQL-Befehle abgearbeitet und das Ergebnis unterhalb der Schaltfläche ausgegeben (siehe [PWE10]). Im Beispiel wird dazu eine vollwertige SQL-Datenbank (HSQLDB) mit vollem Funktionsumfang dynamisch im Browser des Betrachters erzeugt und verwendet, ohne andere Nutzer bei ihrer Arbeit zu beeinträchtigen.

An der Hochschule Zittau/Görlitz nutzen die Studierenden in der Lehrveranstaltung „Algorithmen und Komplexität“ ein Programming-Wiki, um in Kleingruppen von 2-3 Studierenden ein Thema, wie probabilistische Algorithmen oder systematische Suche, in Form eines interaktiven Wiki-Arbeitsblatts für ihre Kommilitonen zu erschließen. In der Lehrveranstaltung stellt jeweils eine Gruppe ihr Thema vor und in einer anschließenden Übungsphase bearbeiten die anderen Studierenden das von dieser Gruppe angefertigte Arbeitsblatt im Programming-Wiki, ganz nach dem didaktischen Konzept des Lernen durch Lehren (vgl. [GW07]). In einem solchen Arbeitsblatt finden sich neben Texten nicht nur interaktive Programmabschnitte sondern auch mathematische Formeln, Abbildungen und eingebundene Videos. Als Programmiersprache wird hier vorwiegend die funktionale Sprache Scheme eingesetzt. Die von den Studierenden formulierten Übungsaufgaben sind vielfältig auch im Hinblick auf den Schwierigkeitsgrad: Sie reichen vom Lesen eines Quellcodes, über dessen Vervollständigung bis hin zur Entwicklung eigener Programme (vgl. [AuK09]).

Durch die Integration der Programming-Wiki-Technologie wird für das betrachtete Lehrgebiet ein signifikanter Mehrwert erzielt.

- Die Studierenden erwerben Kenntnisse und Erfahrungen mit der Entwicklung von Software in Kleingruppen, wie später in der Praxis unter Nutzung von Versionierungssystemen, wie Subversion oder Mercurial benötigt werden. Das Wiki unterstützt diese Arbeitsform naturgemäß und ohne zusätzliche Bedienanforderungen.
- Die Dokumentation der Projektarbeiten auf einer Webseite oder auf Papier und die separaten Abgabe der entwickelten Programme führte früher häufig dazu, dass die Studierenden ihre Arbeit aufteilten. Eine Teilgruppe kümmerte sich primär um die Dokumentation und eine andere Teilgruppe um die Entwicklung der Programme.



**Navigation**

- Startseite
- Letzte Änderungen
- Hilfe

**Suche**

**Werkzeuge**

- Links auf diese Seite
- Änderungen an verlinkten Seiten
- Hochladen
- Spezialseiten

Chat zur Seite anzeigen


Seite
Diskussion
Bearbeiten
Versionen/Autoren

**Inhaltsverzeichnis** [Verbergen]

- 1 Was ist passiert
- 2 Die Hinweise
- 3 Die Datenbank
- 4 Die Aufgabe

**Was ist passiert**

Ein Gängstertrio hat am 23.08.2009 die Central Bank in New York ausgeräumt. Es waren Profis wie der Polizeichef bestätigte: "Hier waren ganz klar Kenner mit Erfahrung am Werk. Sie haben den Coup sicher schon über Wochen im Vorfeld geplant."



**Die Hinweise**

Von den Augenzeugen haben wir nur zwei Hinweise auf die drei männliche Täter erhalten:

1. Hinweis:  
Ein Täter bedrohte die Angestellten und wurde von einem Komplizen „Lie“ gerufen. Er soll mindestens 1'80 groß gewesen sein, zwischen 20 und 30 Jahre alt und trug einen auffallend chinesischen Kleidungsstil.

2. Hinweis:  
Eine Augenzeugin erinnert sich an das Fluchtfahrzeug. Es ist ein blauer oder schwarzer BMW gewesen sagt sie und das Kennzeichen soll auf 557 geendet haben.

**Die Datenbank**

Als FBI-Ermittler und Datenbank-Spezialist bist du für diesen Fall zuständig. Du hast bereits 100 Personen herausgesucht, die als Täter in Frage kommen könnten. Du hast Zugriff auf folgende Tabellen der FBI-Datenbank:

Die Spalten Vater, Mutter, Fahrzeughalter und Person sind Fremdschlüssel auf die Spalte ID der Tabelle Einwohner.

Hier kannst du einen Blick in die Datenbank werfen. Es werden jeweils 10 Datensätze angezeigt. Öffne dazu die Anzeige unterhalb dieses Textes und klicke dann auf ausführen.

Ersten 10 Datensätze verstecken

```

1 SELECT * FROM Einwohner AS Ausschnitt LIMIT 10;
2 SELECT * FROM Vorstrafen AS Ausschnitt LIMIT 10;
3 SELECT * FROM KFZ AS Ausschnitt LIMIT 10;
4 SELECT * FROM Straftatfel AS Ausschnitt LIMIT 10;
5

```

ausführen

**Die Aufgabe**

Abbildung 2: Programming-Wiki am Beispiel Datenbanken mit SQL

Mit der Verwendung des Programming-Wiki scheint diese Aufteilung nicht mehr statt zu finden, da entwickelte Beiträge nun unmittelbar am gleichen Ort dokumentiert werden.

- Die Studierenden entwickeln eine beachtliche Motivation, die in einen selbst auferlegten Qualitätsanspruch mündet. Studierende kommentieren und modifizieren Beiträge und Programmentwürfe von Kommilitonen und es findet ein reger Austausch statt.

Programming-Wiki lässt sich nicht nur auf Hochschulstufe einsetzen. ZUM-Wiki (vgl. [ZUM08]) richtet sich an allgemein bildende Schulen und weist derzeit mehr als 1500 angemeldete Benutzer auf, die Unterrichtsmaterialien zu praktisch allen Fächern der Sekundarstufe 1 und 2 zusammentragen. Seit Herbst 2009 verwendet das ZUM-Wiki ebenfalls die Programming-Wiki-Erweiterungen. An mehreren Gymnasien werden derzeit erste Erfahrungen zum Einsatz im Schulunterricht gesammelt.

Aus Sicht der Informatik lassen sich mit einem Programming-Wiki besonders Grundlagen des Programmierens und algorithmische Konzepte vermitteln. Die Einstiegshürde für Anfänger ist damit sowohl administrativ (Softwareinstallation) als auch bedientechnisch (verglichen mit komplexen Werkzeugen wie Eclipse oder VisualStudio) gering. So ist es beispielsweise möglich, an einem webbasierten Prolog-Schnupperkurs ganz ohne Softwareinstallation teilzunehmen. Für umfangreiche Softwareprojekte wird natürlich der Umstieg auf eine vollwertige Entwicklungsumgebung empfohlen.

### **3 Beispiel: Compiler-Wiki mit AtoCC**

In der Lehrveranstaltung „Abstrakte Denktechniken und theoretische Informatik“ an der Hochschule Zittau/Görlitz müssen die Studierenden in Kleingruppen einen eigenen Compiler entwickeln. Die Quell- und Zielsprache werden dabei vorgegeben, lassen jedoch einen gewissen Interpretationsspielraum zu. Dabei geht es vorzugsweise um Übersetzungsaufgaben mit visueller oder akustischer Ausgabe in der Zielsprache.

In der Lehrveranstaltung wird die Lern- und Arbeitsumgebung AtoCC (vgl. [HW07]) eingesetzt, um Inhalte der theoretischen Informatik, insbesondere formale Sprachen und Automatentheorie, durch praktische Anwendung im automatisierten Compilerbau zu konkretisieren und zu vertiefen.

Während der Planung und Bearbeitung des Auftrags dokumentieren die Studierenden ihre Fortschritte, Überlegungen und Probleme in einem sogenannten Compiler-Wiki. Bei diesem Wiki handelt es sich um ein Wiki, das die Einbettung von mit AtoCC entwickelten Compilern ermöglicht. Ein solcher Compiler kann von den Studierenden ins Wiki hochgeladen werden. Dieser Vorgang erfordert keinen nennenswerten Bedienungsaufwand und kann von jedem registrierten Benutzer ohne Zusatzkenntnisse ausgeführt werden. Auf beliebigen Wiki-Seiten kann der Compiler anschließend direkt online verwendet werden. So steht die Projektarbeit anderen Studierenden, den Dozierenden aber auch weiteren Personen in produktiver Form zur Verfügung.

Gegenüber dem weiter oben vorgestellten Programming-Wiki werden hier komplexe Java-

Programme auf dem lokalen System in einer Lernumgebung entwickelt und anschließend das Endprodukt in der Web 2.0-Umgebung eingebettet. Die im Rahmen dieser Lehrveranstaltung entwickelten Compiler können somit ohne eine lokale Installation von der Lehrperson begutachtet und mit verschiedenen Eingaben getestet werden. Das Compiler-Wiki vereinheitlicht das Ein- und Ausgabeverhalten, wodurch die gleichen Eingabeprogramme und Testszenarien für alle Compiler in gleicher Form verwendet werden können. Dies vereinfacht den administrativen Aufwand für die Begutachtung und Rückmeldung durch die Lehrperson erheblich. Obwohl die Lösungen im Wiki für alle einsehbar und damit auch kopierbar sind, können diese von anderen Gruppen nur schwer unrechtmäßig übernommen werden. Zum einen lässt sich über die Wiki-Versionsverwaltung schnell herausfinden, welche Gruppe die jeweilige Lösung zuerst entwickelt hat und zum anderen würde ein solcher Betrugsversuch, wenn nicht durch die Lehrperson, schnell von den anderen Gruppen erkannt werden.

In Abb. 3 ist ein Teil einer Lösung von Studierenden gezeigt. In dieser konkreten Belegaufgabe sollte ein  $PGN \rightarrow FEN$ -Compiler entwickelt werden. Es handelt sich dabei um zwei verschiedene Sprachen für Schachnotationen (vgl. [Com10]).

Die Erfahrungen mit Compiler-Wiki zeigen, dass die Dokumentation der Compilerentwicklung und -nutzung von den Studierenden wesentlich besser strukturiert wurde als das in traditionellen Papierbelegen früher der Fall war. Die Motivation, das selbst erzeugte Produkt nun auch anderen unmittelbar, d.h. ohne Zusatzmaßnahmen, zeit- und ortsunabhängig zum Erproben präsentieren zu können, setzt beachtliche Kräfte frei. Auch für die mehr oder weniger beständige Dokumentation eigener Studienergebnisse ist dies wertvoll. Dabei geht es nicht nur um einen Archivierungsaspekt, sondern auch um die Möglichkeit, in neuen Projektkontexten auf frühere Teillösungen zurückgreifen zu können.

#### **4 Interaktive Web 2.0-Werkzeuge als Systemtyp**

Die eingebetteten dynamischen Inhalte müssen für eine Webansicht vorgängig aufbereitet werden. Viele Softwarewerkzeuge bieten dies heute bereits an. Tools wie Google SketchUp können mit geringem Aufwand in Webseiten eingebunden werden. Der Hersteller des Computer-Algebra-Systems Maple entwickelte zu diesem Zweck MapleNet, welches Maple-Worksheets direkt im Browser darstellen kann. Bei MapleNet fehlt jedoch eine geeignete Integration in Plattformen wie Wikis oder LMS. In unseren beschriebenen Beispielen, nämlich dem Programming-Wiki und dem Compiler-Wiki, ist eine Einbettung der entsprechenden Umgebungen relativ einfach möglich. Als Basis wird ein MediaWiki-System (vgl. [Med08]) verwendet. MediaWiki ist eine der größten freien Wiki-Engines und bildet auch die Grundlage der Wikipedia. Unser Ansatz ist jedoch nicht auf Wikis beschränkt, sondern lässt sich beispielsweise auch auf Blogs oder komplexere Lernplattformen übertragen. Dazu ist eine geringfügige Anpassung der jeweiligen Server-Software nötig. In der Regel bieten Web 2.0-Werkzeuge eine Plugin/Extension-Schnittstelle an, über die dynamische Inhalte eingebettet werden können.

Im konkreten Fall des Compiler-Wikis wurde eine in der Programmiersprache PHP geschriebene

Admin Eigene Diskussion Einstellungen Beobachtungsliste Eigene Beiträge Abmelden

llb08 Gruppe 2 Seite Diskussion Bearbeiten Versionen/Autoren Löschen Verschieben Schützen Beobachten

**Navigation**

- Startseite
- Was ist ein Wiki?
- Hilfe zur Formatierung
- Compiler hochladen
- Canvas Beschreibung
- Turtle Beschreibung
- Letzte Änderungen

**Suche**

Seite Suchen

**Werkzeuge**

- Links auf diese Seite
- Hochladen
- Spezialseiten
- Druckversion
- Permanentlink
- PDF Version
- Chat zur Seite anzeigen

**Inhaltsverzeichnis (Verbergen)**

- 1 Gruppenmitglieder
- 2 Die Sprache PGN
  - 2.1 Definition & Beispiel
    - 2.1.1 Definition
    - 2.1.2 Beispiel
  - 2.2 Planung des Übersetzungsprozesses
  - 2.3 Grammatik für PGN
    - 2.3.1 Allgemein
    - 2.3.2 Regeln
  - 2.4 Festlegung der Token
    - 2.4.1 Allgemein
    - 2.4.2 Tokendefinition
- 3 Automatisierte Parsergenerierung
  - 3.1 Reduzierte Grammatik
  - 3.2 Scannergenerator und Scannertest
  - 3.3 Parsergenerator und Parsertest
- 4 Compiler
  - 4.1 Zielsprache FEN
    - 4.1.1 Allgemein
    - 4.1.2 Aufbau
  - 4.2 S-Ausdrücke
    - 4.2.1 Aufbau der Regeln
    - 4.2.2 Semantische Regeln
  - 4.3 Compiler Beispiel
- 5 Fazit
- 6 Links
- 7 Dateien

**Ausgeblendete Teile**

**Regeln:**

- Angabe der Spielfiguren mit einem großgeschriebenen Anfangsbuchstaben
- Bauern werden in der Regel nicht mit ihrem Anfangsbuchstaben angegeben, können auch damit beginnen
- Linie werden mit den Kleinbuchstaben von a bis h angegeben
- Angabe der Reihe mit 1 bis 8
- Anfangsposition der Spielfiguren in den Reihen 1,2 und 7,8
- Jeder Zug einer Figur wird mit dem Anfangsbuchstaben und dem Ankunftsfield beschrieben
- wenn 2 Figuren der gleichen Farbe auf ein gleiches Ankunftsfield ziehen wollen, dann:
  - wenn Sie in der gleichen Reihe stehen, wird der Anfangsbuchstabe der Figur, die Herkunftsreihe sowie das Ankunftsfield notiert
  - wenn Sie in der gleichen Linie stehen, wird der Anfangsbuchstabe der Figur, die Herkunftsreihe sowie das Ankunftsfield notiert
  - wenn Sie aus unterschiedlichen Reihen und Linien kommen, wird dem ersten Punkt Vorrang gegeben
- beim Schlagen wird zwischen dem Anfangsbuchstaben der Spielfigur und dem Ankunftsfield ein x geschrieben
- wenn ein Bauer schlägt, wird die Reihe des Ausgangsfield gefolgt von einem x und dem Ankunftsfield notiert
- bei einer Bauernumwandlung wird der Bauernzug angegeben, gefolgt von einem = und dem Anfangsbuchstaben der neuen Figur

**Beispiel** [\[Bearbeiten\]](#)

```
[Event "IBM Kasparov vs. Deep Blue Rematch"]
[Site "New York, NY USA"]
[Date "1997.05.11"]
[Round "6"]
[White "Deep Blue"]
[Black "Kasparov, Garry"]
[Opening "Caro-Kann: 4...Nd7"]
[ECO "B17"]
[Result "1-0"]

1.e4 c6 2.d4 d5 3.Nc3 dxe4 4.Nxe4 Nd7 5.Ng5 Ngf6 6.Bd3 e6 7.N1f3 h6
8.Nxe6 Qe7 9.O-O fxe6 10.Bg6+ Kd8 {Kasparov schuetzelt kurz den Kopf}
11.Bf4 b5 12.a4 Bb7 13.Re1 Nd5 14.Bg3 Kc8 15.axb5 cxb5 16.Qd3 Bc6
17.Bf5 exf5 18.Rxe7 Bxe7 19.c4 1-0
```

**Compiler Beispiel** [\[Bearbeiten\]](#)

```
x| 1 [Event "IBM Kasparov vs. Deep Blue Rematch"]
2 [Site "New York, NY USA"]
3 [Date "1997.05.11"]
4 [Round "6"]
5 [White "Deep Blue"]
6 [Black "Kasparov, Garry"]
7 [Opening "Caro-Kann: 4...Nd7"]
8 [ECO "B17"]
9 [Result "1-0"]
10 1.e4 c6 2.d4 d5 1-0
11
```

**speichern & ausführen**

```
> rnbqkbnr/pppppppp/8/8/4P3/8/PPPP1PPP/RNBQKBNR b - e3 0 1
rnbqkbnr/pp1ppppp/2p5/8/4P3/8/PPPP1PPP/RNBQKBNR w - - 0 1
rnbqkbnr/pp1ppppp/2p5/8/3PP3/8/PPPP2PPP/RNBQKBNR b - d3 0 2
rnbqkbnr/pp2pppp/2p5/3p4/3PP3/8/PPPP2PPP/RNBQKBNR w - d6 0 2
```

Schachbrett

Abbildung 3: Auszug einer Belegaufgabe von Studierenden im Compiler-Wiki



Erweiterung für MediaWiki entwickelt. Das Wiki verarbeitet eine Wikiseite, indem der darin enthaltene WikiCode in HTML transformiert wird. Durch eine Erweiterung lassen sich die gewünschten Ausgaben für ganz bestimmte Schlüsselwörter definieren: Wird etwa in einer Wiki-Seite das Tag `<uploadcompiler>` verwendet, erzeugt die Compiler-Wiki-Erweiterung ein HTML-Formular, über das ein in AtoCC entwickelter Compiler hochgeladen werden kann.

Die Entwicklung derartiger Erweiterungen ist stark an der jeweiligen Server-Software ausgerichtet. Eine MediaWiki-Erweiterung lässt sich beispielsweise nicht einfach in eine andere Wiki-Umgebung oder eine Lernplattform wie Moodle integrieren. Erweiterungen werden deshalb häufig nur von Entwicklern erstellt, die mit der jeweiligen Plattform besonders gut vertraut sind. Plugins für Lehrinhalte findet man jedoch eher selten, da die Zielgruppe verglichen mit dem Nutzerkreis eines YouTube-Plugins klein ausfällt.

Allgemein kann ein Web 2.0-Anwendung bezüglich dynamischer Inhalte in drei Gruppen unterteilt werden:

- Eine normale Web 2.0-Anwendung ohne spezielle dynamische Inhalte;
- Eine Web 2.0-Anwendung, welche vorgegebene dynamische Inhalte verwendet, mit denen die Lernenden interagieren können (z.B. Programming-Wiki);
- Eine Web 2.0-Anwendung, welche es den Lernenden erlaubt, selbst dynamische Inhalte einzustellen, mit denen anschließend alle Nutzer interagieren können (z.B. Compiler-Wiki).

## **5 Erfahrungen und Ausblick**

Bei der Bearbeitung der vorgestellten Projektarbeiten gingen die Studierenden sehr unterschiedlich vor. Einige Gruppen begannen sehr früh mit der Nutzung des Wikis, nahmen zunächst eine Gruppeneinteilung und eine grobe Inhaltsplanung vor und arbeiteten über den vorgegebenen Zeitraum hinweg an der Vervollständigung der Beiträge im Programming-Wiki bzw. Compiler-Wiki. Aus der Versionsgeschichte ließ sich bei diesen Gruppen gut erkennen, welche Teile von welchem Gruppenmitglied entwickelt wurden. Andere Gruppen erarbeiteten die Inhalte zunächst in einer traditionellen Umgebung (z.B. Word) und übertrugen diese erst zum Schluss ins Wiki. Die Endfassung wurde häufig von einer Person aus dem Team koordiniert. Die Zuordnung der Beiträge zu den einzelnen Gruppenmitgliedern war somit schwierig. Um die Vorteile der Dokumentation von Projektarbeiten in Web 2.0-Werkzeugen wirklich ausnutzen zu können, sollten deshalb bereits im Vorfeld feste Termine vorgegeben werden, an denen bestimmte Zwischenstände erreicht und dokumentiert sein müssen. Um bei Gruppenarbeiten die Anteile der einzelnen Mitglieder besser zuordnen zu können, wird eine persönliche Anmeldung mit einem eigenen Benutzernamen dringend empfohlen.

Am Ende der Lehrveranstaltungen wurde jeweils eine formfreie formative Evaluation durchgeführt, bei der die Studierenden positive und negative Aspekte der Vorlesung zusam-

mentragen sollten. Die praktischen Übungen in den Wiki-Umgebungen wurden dabei mehrfach als besonders positiv und motivierend angegeben. Wir vermuten, dass durch die Präsentation der eigenen Arbeit in einem Wiki ähnliche soziale Bedürfnisse zur Selbstdarstellung angesprochen werden, wie es etwa bei der Nutzung von Web 2.0-Plattformen wie YouTube oder Facebook der Fall ist. Das verwendete Programming-Wiki könnte durchaus als eine Art „Programming-YouTube“ bezeichnet werden, in dem die Studierenden ihre Arbeit sowohl Lehrpersonen, Freunden, Eltern aber auch anderen Gruppen zur Schau stellen.

Die Entwicklung von Erweiterungen für die verschiedenen Web 2.0-Plattformen ist nicht trivial und zeitaufwändig. Unsere zukünftigen Bestrebungen liegen deshalb in der Entwicklung einer einheitlichen, plattformunabhängigen Schnittstelle für derartige interaktiver Elemente, um deren Herstellung zu vereinfachen.

## Literatur

- [AuK09] Programming-Wiki, studentische Arbeiten aus dem Sommersemester 09, Juli 2009.  
<http://www.michael-hielscher.de/wiki/index.php/AuK>.
- [Com10] Compiler-Wiki, studentische Arbeiten aus dem Wintersemester 09/10, Februar 2010.  
<http://www.michael-hielscher.de/compilerwiki/>.
- [Döb07] Beat Döbeli. Wiki und die starken Potenziale - Unterrichten mit Wikis als virtuellen Wandtafeln. *Zeitschrift Computer und Unterricht*, Web 2.0 und Schule Nr 66:39–41, 2007.
- [GW07] Joachim Grzega and Franz Waldherr. Lernen durch Lehren (LdL) in technischen und anderen Fächern an Fachhochschulen. *Projektseminare für Lehrende in technischen Fächern.*, Schriftreihe DiNa, Ausg. 11:1–17, 2007.
- [HW07] Michael Hielscher and Christian Wagenknecht. AtoCC didaktischer Ort und erste Erfahrungen. *Didaktik der Informatik in Theorie und Praxis*, INFOS'07:159–170, 2007.
- [HW09] Michael Hielscher and Christian Wagenknecht. Programming-Wiki: Online programmieren und kommentieren. *Zukunft braucht Herkunft. 13. GI-Fachtagung Informatik und Schule*, INFOS'09:281–292, 2009.
- [Med08] MediaWiki, Oktober 2008.  
<http://www.mediawiki.org>.
- [PWE10] Programming-Wiki, Februar 2010.  
<http://www.michael-hielscher.de/wiki/>.
- [ZUM08] ZUM Wiki, Oktober 2008.  
<http://wiki.zum.de/>.