

<JACK:lin> – Linguistische Module für das E-Assessment mit JACK

Tim Kocher

Ulrike Haß

Bernhard Schröder

Germanistik/Linguistik // Universität Duisburg-Essen

{tim.kocher, ulrike.hass, bernhard.schroeder}@uni-due.de

Abstract

In unserem Demo-Beitrag stellen wir Module für linguistische E-Klausuren und Übungen mit automatischem Feedback im Test- und Prüfungssystem JACK vor.

1 Rahmenbedingungen

In den linguistischen Fachanteilen der philologischen Fächer sind Vorlesungsgrößen von mehreren hundert Hörerinnen und Hörern keine Seltenheit. Modulprüfungen mit entsprechenden Teilnehmerzahlen bedeuten einen hohen Organisations- und Korrekturaufwand. Papierklausuren bedeuten für die Korrektur einen hohen zeitlichen Aufwand, der i. d. R. auf mehrere Personen verteilt werden muss. Kleinere schriftliche Übungen sind mit dem Wegfall von Übungen bzw. Tutorien unter diesen Bedingungen höchstens stichprobenartig durchführbar, ein individuelles Feedback ist nicht möglich.

Um den Korrekturaufwand zu reduzieren und die Zeit zwischen Test bzw. Klausur und Rückmeldung der Ergebnisse an die Studierenden zu reduzieren, bietet sich der Umstieg auf elektronische Lösungen – E-Assessment – an. Vorteile sind, je nach Aufgabentyp, automatisierte Korrekturen von E-Klausuren einerseits, sowie um Hinweise erweiterbare Übungsaufgaben mit unmittelbarem Feedback andererseits. Für Tests mit automatisiertem Feedback und E-Klausuren werden verschiedene Systeme angeboten, die sich hinsichtlich ihrer Möglichkeiten und Stärken deutlich unterscheiden. Nach der Prüfung von Alternativen arbeiten wir derzeit mit einer Kombination der Systeme Moodle und JACK. Während mit Moodle v. a. Lernmaterialien bereitgestellt werden, lassen sich mit JACK Aufgaben für beide o. g. Szenarien erzeugen.

2.1 Die Lösung mit <JACK:lin>

Moodle einerseits hat den Vorteil, dass es als Lernplattform einen Online-Kursraum bietet, in welchem Literatur (verschiedene Textdateiformate), ergänzt durch unterschiedliche multimediale Inhalte (Video-/Audio-Dateien, Hyperlinks) einem definierten Nutzerkreis, i. d. R. Studierende eines Seminars bzw. einer Vorlesung, zur Verfügung gestellt werden kann. Darüber hinaus sind neben Befragungen auch Tests mit unterschiedlichen Aufgabentypen durchführbar (darunter Lückentext, Multiple Choice, Zuordnung). In unserem speziellen Einsatzkontext haben sowohl rechtliche als auch technische Bedenken uns von einer Nutzung als E-Klausur-System Abstand nehmen lassen.

Mit *JACK* steht andererseits ein E-Assessment-System zur Verfügung, das für Prüfungen genutzt werden kann und bereits an verschiedenen Universitäten entsprechend genutzt wird (Striwe/Goedicke 2013). Der ursprünglich intendierte Anwendungsbereich von *JACK* waren die MINT-Fächer, insbesondere die Informatik und die Mathematik. Typische Aufgabenarten sind entsprechend Programmierübungen (erst in Java, mittlerweile auch in C++ (Striwe et al. 2008)) sowie formelbasierte Aufgaben zur Mathematik (ermöglicht durch die Einbindung von LaTeX). Obwohl *JACK* also nicht ursprünglich für sprachwissenschaftlich orientierte AnwenderInnen konzipiert war, scheint es uns als Entwicklungsplattform für eine linguistische E-Klausur geeignet. Zwei Gründe sind dabei für unsere Bewertung ausschlaggebend:

1. *JACK* ist ein nach wie vor in der Entwicklung begriffenes System, das ständig verbessert und erweitert wird. In enger Zusammenarbeit mit den EntwicklerInnen kann der Funktionsumfang stetig angepasst werden.
2. Die Aufgabenerstellung ist ein modularisiertes System, bei dem über eine webgestützte graphische Benutzeroberfläche XML-Dateien zu (ggf.

aufeinander aufbauenden) Aufgaben verknüpft werden. Da das eigentliche Aufgabendesign innerhalb der XML-Dateien erfolgt, lassen sich – innerhalb der durch den Aufgabentyp gesetzten Grenzen – eine Vielzahl spezifisch sprachwissenschaftlicher Aufgaben erstellen. Dies ist von besonderem Interesse, da die zu erstellende Klausur eine große Bandbreite linguistischer Themengebiete abzudecken hat: Von Semiotik über Phonetik und Phonologie, Morphologie über Syntax zu Semantik und Pragmatik. So lässt sich z. B. die Darstellung relevanter IPA-Zeichen über Unicode-Referenzen in XML in den Aufgaben realisieren. Durch die Annotationsmöglichkeiten, die XML bietet, lässt sich ein Pool unterschiedlicher Satzgliedanalysen in Form XML-annotierter Sätze hinterlegen, aus welchem JACK dann randomisierte Aufgaben zur Satzgliedzerlegung und -klassifikation in Form eines speziellen Multiple-Choice-Aufgabentyps generieren kann.

Kurzfristig beschränken wir uns auf die Aufgabentypen Multiple Choice unter Einschluss des speziellen Subtyps Satzgliedanalyse und Lückentext, mittelfristig werden wir die Implementierung von Aufgaben zur Manipulation von Baumstrukturen anstreben. Für Aufgaben dieses Typs gibt es auch außerhalb der Linguistik, z. B. in der Informatik, Bedarf.

Ein weiterer Vorteil von JACK ist, dass sich mit JACK erstellte Aufgaben in die Moodle-Kurse einbetten lassen. Moodle ist bereits so etabliert, dass diese Einbettung die Akzeptanz von JACK seitens der Studierenden deutlich erhöhen wird. Wir könnten also unter Beibehaltung der gewohnten Lernplattform den Studierenden mit JACK ein Tool zur selbstständigen Lernfortschrittsevaluation an die Hand geben. In dieser Hinsicht ist JACK Moodle ebenfalls überlegen, da sich die Aufgaben in Schleifen anordnen lassen, die bei fehlerhaften Antworten (ggf. nach Informationsgehalt gestaffelt) Hinweise anzeigen, mit deren Hilfe die Aufgabe erneut bearbeitet werden kann.

2.2 Warum JACK?

Zwei alternative E-Assessment-Systeme wurden von uns getestet: Moodle und L-Plus. Wegen der zu geringen Detailtiefe der Protokollierung wird an unserer Universität aus rechtlichen Gründen derzeit von der Durchführung von Prüfungen mit Moodle abgeraten. Von L-Plus konnten wir uns im Rahmen eines mehrstündigen Lehrgangs einen Eindruck verschaffen. Die Eigenschaften, die unserer Ansicht nach für JACK sprechen, sind

1. die Einbindungsmöglichkeit von XHTML-Elementen: Tabellen, Formeln, Unicode;
2. flexiblere Auswertungsmöglichkeiten und
3. die XML-basierte Programmierbarkeit der Aufgaben.

Es ergibt sich in Summe ein Vorteil zugunsten von JACK, da die Aufgaben(-Typen) leichter variiert werden können und JACK die programm-basierte Generierung komplexer Spezifikationen ermöglicht. Letzteres ist für uns besonders für den fachspezifisch relevanten Aufgabentyp „Satzgliedanalyse“ interessant, da gerade dieser Aufgabentyp von den alternativen Programmen nicht oder nur mit hohem Aufwand überhaupt realisiert werden kann.

Literatur

Michael Striwe, Michael Goedicke and Moritz Balz. 2008. *Computer Aided Assessments and Programming Exercises with JACK*. Technical report 28, ICB, University of Duisburg-Essen.

Michael Striwe and Michael Goedicke. 2013. JACK revisited: Scaling up in multiple dimensions. *Proceedings of Eighth European Conference on Technology Enhanced Learning (EC-TEL)*, Paphos, Cyprus, 635-636.