

# Intelligenter Support für Übungen und den Übungsbetrieb (ITS)

## Studienkontext: Prolog und Prolog-Kurse in Osnabrück

Prolog ist eine relativ junge Programmiersprache, die Anfang der 80er Jahre entwickelt wurde, um Probleme der maschinellen Verarbeitung natürlicher Sprache theoretisch fundiert behandeln zu können. Dazu wurde auf ein Verfahren aus dem Bereich des *automatischen Beweisens* zurückgegriffen, die SLD-Resolution über Horn-Klauseln. Im Kern ist ein Prolog-Interpreter also ein automatischer Beweiser für eine Teilmenge der Prädikatenlogik erster Stufe. Dadurch unterscheidet Prolog sich grundsätzlich von anderen, prozeduralen, funktionalen oder objektorientierten Programmiersprachen: Im Vordergrund steht die deklarative Beschreibung eines Problems im Gegensatz zur expliziten Spezifikation des Lösungsweges.

Im Rahmen der Studiengänge Computerlinguistik und Künstliche Intelligenz sowie Cognitive Science ist ein regelmäßig durchgeführter Kurs „Logisches Programmieren“ zentraler Bestandteil des Grundstudiums. Dabei soll nicht lediglich die Fähigkeit vermittelt werden, syntaktisch korrekte Prolog-Programme zu verfassen, sondern ebenfalls eine Reihe von Techniken, die notwendig sind, um die Struktur von Problemen abzubilden.

Die Kurse werden als vierstündige Vorlesung durchgeführt und setzen keine Programmierkenntnisse oder besonderes mathematisches Wissen voraus. Wie in natur- und ingenieurwissenschaftlichen Veranstaltungen üblich, erfolgt die Leistungskontrolle über Klausuren und wöchentlich zu bearbeitende Übungsaufgaben. Zur Betreuung sowie Vor- und Nachbereitung dieser Aufgaben gibt es wöchentlich zusätzlich zweistündige Übungen, die von den studentischen Tutoren betreut werden, die auch die Korrektur und Kommentierung der abgegebenen Aufgaben übernehmen.

Der Osnabrücker Prolog-Kurs orientiert sich in seiner Struktur an gängigen Lehrbüchern wie z.B. Clocksin/Mellish (1994) oder Bratko (2000) und geht in seinem späteren Verlauf speziell auf Methoden und Techniken aus der Künstlichen Intelligenz (Suchverfahren) und Computerlinguistik (Grammatiken und Parsing) ein. Das hat Auswirkungen auf die Exportierbarkeit eines Kurses, der hier in seiner Gesamtheit nicht einfach als „Einführung in Prolog“

verstanden werden kann, obwohl eine solche Orientierung eine typische Anwendungsdomäne von Prolog darstellt und daher nicht ungewöhnlich ist. Es wird nicht ein einzelnes Lehrbuch zu Grunde gelegt, sondern ein über mehrere Jahre entstandenes Skript (Gust/Tischler o.J.), das zuletzt im Rahmen des Osnabrücker „Virtueller Campus“-Teilprojektes vollständig überarbeitet (Tschorn 2001) und zusätzlich ins Englische übersetzt wurde.

## Didaktische Einordnung

Der Kurs - und damit auch die vorhandenen Übungsaufgaben - lässt sich in drei Blöcke gliedern:

1. Einführung. Die Prolog-Syntax wird vorgestellt und die Konzepte „Fakten“ und „Regeln“ sowie die zentralen Datenstrukturen „Atome“ und „Listen“ eingeführt. Dieser Teil umfasst ca. vier Wochen, also vier Übungsblätter mit je drei bis vier Aufgaben. An einfachen Beispielen werden die grundlegenden Programmier- und Verarbeitungskonzepte in Prolog, hauptsächlich Rekursion, Unifikation und Backtracking geübt.
2. Vertiefung. Im zweiten Block wird das erworbene Anwendungswissen vertieft und mit theoretischem Hintergrund gefestigt. Hierzu zählt das Verhältnis von Prädikatenlogik und Prolog, genauere Kenntnisse über das Beweisverfahren von Prolog und umfangreichere Aufgaben, die die Kombination mehrerer Techniken verlangen.
3. Anwendung und Spezialisierung. In diesem Block verlieren die Aufgaben ihren reinen Übungscharakter. Sie werden komplexer und können sich auch als Teilaufgaben definiert über mehrere Blätter erstrecken. Die Hinwendung zu spezifisch computerlinguistischen Fragestellungen oder solchen aus der Künstlichen Intelligenz findet sich hauptsächlich in diesem Modul, das auch mit anderen Inhalten gefüllt werden kann, ohne dass die ersten beiden Teile geändert werden müssten.

Dieser Kurzüberblick zeigt, dass auch für andere Kurskonzeptionen Teile des Osnabrücker Skripts und der Aufgaben verwendet werden können, insbesondere die grundlegenden Übungen des ersten Teils. Dabei ist aber als Grundausrichtung zu beachten, dass es sich um einen Programmierkurs handelt, der umfassende und praktische Programmierfähigkeiten vermitteln möchte, nicht so sehr um einen Wissensrepräsentationskurs, der stärkeres Gewicht auf Darstellung und Aufbereitung von Sachverhalten legt. Der erfolgreiche Einsatz von Skript und Aufgaben im Rahmen eines vierwöchigen Prologkurses in Hannover hat gezeigt, dass für ähnlich ausgerichtete Kurse leicht Teilmengen

Vorabversion! Erschienen als: *Christoph Peylo, Claus Rollinger, Tobias Thelen: "Intelligenter Support für Übungen und den Übungsbetrieb"*, als Unterkapitel von Hauenschild, Christa; Neijdl, Wolfgang; Rollinger, Claus; Wagner, Erwin; Womser-Hacker, Christa: "Der Entwicklungsverbund: Sprache - Wissen - Information". In: Wagner, Erwin und Kindt, Michael (Hrsg.): "Virtueller Campus: Strategien - Szenarien - Studium". Münster: Waxmann, S. 462-506. 2001

definierbar sind, die einzelne inhaltliche Fäden von der Einführung über die Vertiefung bis zur Anwendung verfolgen.

## Lernumgebung: Die Zugänge zum VC-Prolog-System

Das VC-Prolog-System (s. Peylo et al. 1999, 2000) ist ein Client-Server-System, das als universeller Rahmen für internetbasierte Lehr- und Lernumgebungen mit integrierter Verwaltung eines Übungsbetriebes dienen kann. Das System bietet für Studierende und Kursleiter eigene Arbeitsumgebungen, die über WWW-Browser zugänglich und entsprechend den unterschiedlichen Erfordernissen angepasst sind. Lehrenden bietet das System eine komplette Kursverwaltung, einschließlich Benutzer- und Aufgabenverwaltung sowie einer Korrekturumgebung. Studierenden präsentiert sich das System als Netz-basiertes intelligentes Lehr-/Lernsystem.

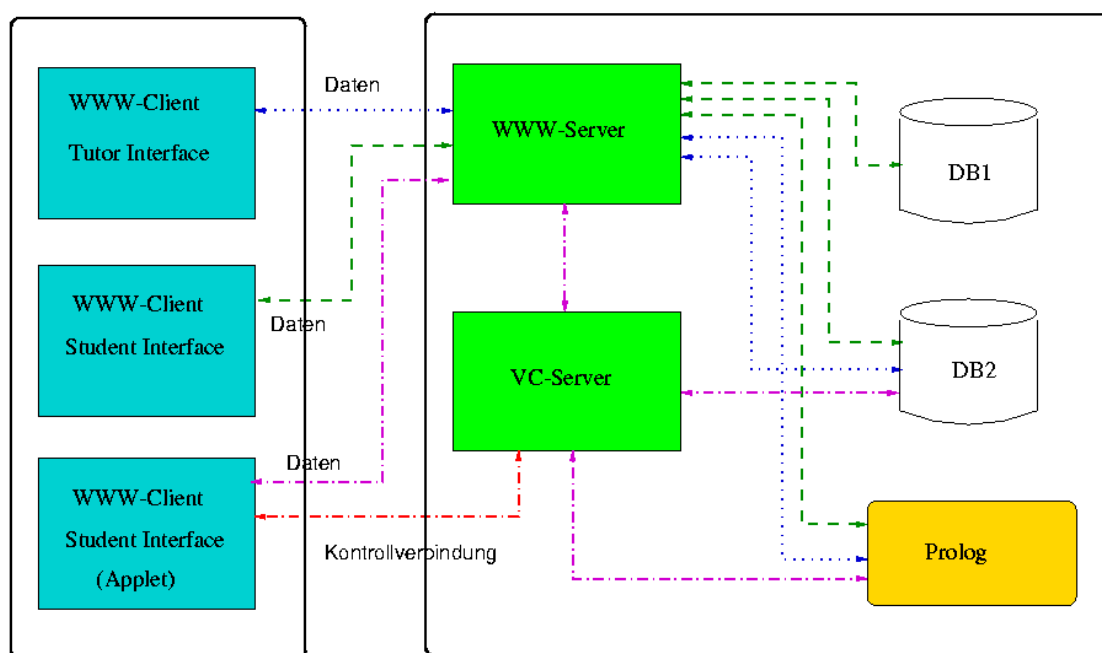


Abbildung 1: Die verschiedenen Schnittstellen des VC-Prolog-Systems

Durch den Server wird jedem Benutzer, der über das Internet auf ihn zugreift, ein eigener Applikationsprozess (Prolog) zur Verfügung gestellt. Darüber hinaus verwaltet er über eine Datenbank die Daten für den Lehr- und Übungsbetrieb. Der Zugang zu den verschiedenen Bereichen ist über personen- und gruppenspezifische Zugriffsrechte reglementiert. Durch das Zusammenspiel von Client und Server kann der Lernende auf die entfernt liegenden Daten wie auf Teile des eigenen Dateisystems zugreifen.

Studierende haben die Möglichkeit über zwei verschiedene Schnittstellen am VC-System zu arbeiten (siehe Abbildung 1): entweder über ein Java-Applet, das durch einen Java-fähigen Browser bzw. SUN's Appletviewer benutzt werden kann, oder eine CGI-ähnliche Schnittstelle, die auf der Java-Servlet Technologie beruht. Das Applet besitzt Basisfähigkeiten eines Texteditors und bietet somit komfortables Arbeiten beim Erstellen von Prologcode. Außerdem ist über das Applet die Fehlerkorrekturkomponente und ein graphischer Debugger zugänglich, die den Lerner bei der Lösung seiner Aufgaben unterstützen. Der Lerner kann einen Lösungsansatz an den Prolog-Online-Tutor schicken (PLOT, s. Bönner et al. 1999) und erhält Hilfestellungen, die ihm bei der weiteren Lösung der Aufgabe helfen sollten. Die Analyse basiert auf dem Ansatz, die vom Lerner produzierte Lösung mit einer oder mehreren korrekten Lösungen, sogenannten Musterlösungen, zu vergleichen. Die Fehlerbehandlung interpretiert die von den einzelnen Analysemodulen gefundenen Fehler und präsentiert sie mit entsprechenden Erklärungen dem Benutzer. Weiterhin werden während der Analyse des Programmcodes Informationen über die Struktur des Programmes generiert und an ein Modul weitergeleitet, das daraus den Ablauf des Programms graphisch veranschaulichen kann. Dies erleichtert dem Benutzer das Verständnis für Ablauf und Arbeitsweise des Prolog-Beweislers.

Allerdings ist das Applet auf eine ständige Verbindung mit dem VC-Server angewiesen. Das Arbeiten daran setzt daher eine persistente Internetverbindung voraus. Für Studierende, die über eine Modem- oder ISDN-Verbindung mit dem System arbeiten wollen, ist diese Schnittstelle ungeeignet, da sie für die Zeit der gesamten Sitzung online sein müssten. Aus diesem Grund ist die Servlet-Schnittstelle eingerichtet worden, die zwar wesentlich weniger Unterstützung bietet, dafür aber *dial-up on demand* des Betriebssystems unterstützt.

Die Schnittstelle für Tutoren basiert auf Java-Servlets. Dies hat den Vorteil, dass keinerlei Vorgaben bezüglich des verwendeten Webbrowsers gemacht werden müssen, da die auf Benutzerseite dargestellten Ausgaben reines HTML sind.

## **Die Tutorenarbeitsumgebung**

Die Arbeitsumgebung für Tutoren soll die Kursleiter und betreuenden Hilfskräfte bei allen Aufgaben im Rahmen eines Kurses unterstützen. Wesentliche Bereiche sind hierbei:

**Benutzer- und Gruppenverwaltung:** Zu Beginn des Semesters müssen die Benutzer ein Login erhalten, um an dem System arbeiten zu können.

Studierende können sich in Arbeitsgruppen zusammenschließen, die Aufgaben gemeinsam bearbeiten. Das System bietet Formulare zum Eingeben von Benutzerdaten, die dabei gleich in Arbeitsgruppen zusammengefaßt werden können. Auch das Wechseln von Studierenden in andere Arbeitsgruppen, das Löschen vorhandener oder die Bildung neuer Gruppen kann über das System durchgeführt werden.

**Aufgabenverwaltung:** Für den Übungsbetrieb müssen Übungsaufgaben erstellt, exemplarische Lösungen (Musterlösungen) angefertigt und zu erreichende Punktzahlen festgelegt werden. Hierzu werden seitens des Systems geeignete Werkzeuge zur Verfügung gestellt. Beispielsweise kann bei dem Erstellen von Musterlösungen ein Veröffentlichungsdatum angegeben werden, ab dem die Lösung von den Studenten eingesehen werden kann.

**Korrekturumgebung:** Die Bewertung der Aufgaben für den Leistungsnachweis erfolgt durch menschliche Tutoren. Um die Korrektur zu erleichtern, kann eine abgegebene Lösung gleich mit vordefinierten Testanfragen in Prolog überprüft werden. Auf diese Weise können Lösungen rasch und effizient getestet werden.

**Class monitoring:** Anhand des Abschneidens aller Gruppen bei der Bearbeitung der Übungsaufgaben kann ein Profil des Kurses erstellt werden.

Für jede der genannten Teilaufgaben gibt es HTML-Seiten mit Formularen, über die diese Arbeiten erledigt werden können. Die einzelnen Seiten sind von einer zentralen, durch Passwort geschützten Einstiegsseite aus zugänglich.

## **Die Studentearbeitsumgebung**

Netzbasierte Lehrsysteme haben im Gegensatz zu den traditionellen Lehr/Lernsystemen in der Regel das Lehrmaterial integriert (vgl. Brusilovski 1999). Das gesamte Kursmaterial ist im Netz publiziert und über die Einstiegsseiten des Kurses allgemein zugänglich gemacht. Das Kursmaterial beinhaltet das Prolog-Skript der Veranstaltung (darüber hinaus ein paar Kapitel eines adaptiven Skriptes), das Benutzerhandbuch zum verwendeten Prologdialekt (SWI-Prolog), sowie Verweise auf weitere im Netz zugängliche Prolog-Ressourcen und Literatur zu Prolog.

Die Applet- bzw. Servlet-Schnittstellen für einen Kurs sind über Passwörter geschützt. Beide Schnittstellen bieten den Lernenden folgende Funktionalität:

- Einsehen der korrigierten Übungsaufgaben, d.h. der Kommentare des Korrektors;

Vorabversion! Erschienen als: *Christoph Peylo, Claus Rollinger, Tobias Thelen: "Intelligenter Support für Übungen und den Übungsbetrieb"*, als Unterkapitel von Hauenschild, Christa; Neijdl, Wolfgang; Rollinger, Claus; Wagner, Erwin; Womser-Hacker, Christa: "Der Entwicklungsverbund: Sprache - Wissen - Information". In: Wagner, Erwin und Kindt, Michael (Hrsg.): "Virtueller Campus: Strategien - Szenarien - Studium". Münster: Waxmann, S. 462-506. 2001

- Einsicht in die Liste der freigegebenen Übungsaufgaben;
- Möglichkeit der Bearbeitung von Aufgaben und einer Schnittstelle zu Prolog.

Die Einbettung des Kurses in eine Web-basierte Umgebung bietet den Lernenden einige Vorteile. So ist es nicht notwendig, Prolog auf dem eigenen Rechner zu installieren. Auch kann ein Lernender während der Bearbeitung einer Übungsaufgaben im Skript der Veranstaltung oder dem Manual der entsprechenden Programmiersprache „nachblättern“, ohne seine Arbeitsumgebung verlassen zu müssen.

## Austausch und Erfahrungen: Nutzung des VC-Systems

Im Sommersemester 2001 arbeiten im Rahmen eines Prologkurses ca. 70 Studierende an dem System. Daneben verfügen insgesamt 80 externe Benutzer über ein Login in das System. Über die Analyse von Server-Logdateien lässt sich die Nutzung des Systems anhand der Protokollierung von Anfragen nachvollziehen. Für den Monat Mai ergibt sich dabei folgendes Bild: Es gab 125.160 Anfragen aus 50 verschiedenen Ländern auf die VC-Seiten.

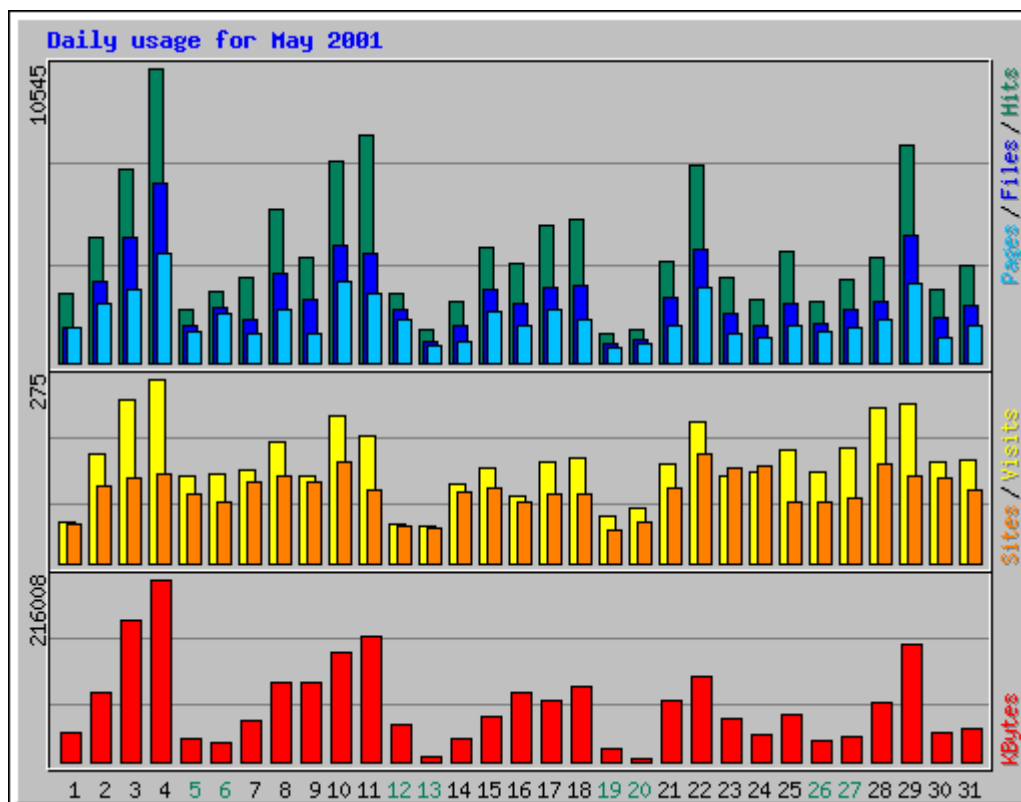


Abbildung 2: Zugriffe an einzelnen Tagen



Die am häufigsten angeforderten Seiten liegen in den Arbeitsbereichen der Studierenden und Tutoren. Dabei gab es im studentischen Bereich über 10.000 Anfragen auf Seiten, mit denen an Prolog gearbeitet wird. Darunter entfallen fast 1500 auf Sitzungen mit dem Applet-Interface. Auf Seiten aus der Arbeitsumgebung der Tutoren wurde mehr als 5000 mal zugegriffen. Das Gros der restlichen Anfragen verteilt sich auf die weiteren Seiten des Kursbereichs, insbesondere das Skript.

Die Serverlogs lassen auch Rückschlüsse über das Arbeitsverhalten der Benutzer zu. Demnach wird an Werktagen deutlich mehr als an Wochenenden auf das System zugegriffen (siehe Abbildung 2) und es erfolgen mehr Zugriffe über die Mittagsstunden (siehe Abbildung 3). Letzteres hängt auch mit dem Belegungsplan der zugänglichen Rechnerarbeitsräume zusammen. Die meisten Zugriffe erfolgen aus der Domäne des Universitätsnetzes und der Computerarbeitsräume. Da diese auch für Veranstaltungen genutzt werden, wird offensichtlich häufig die Mittagspause für Arbeiten an den Programmieraufgaben genutzt.

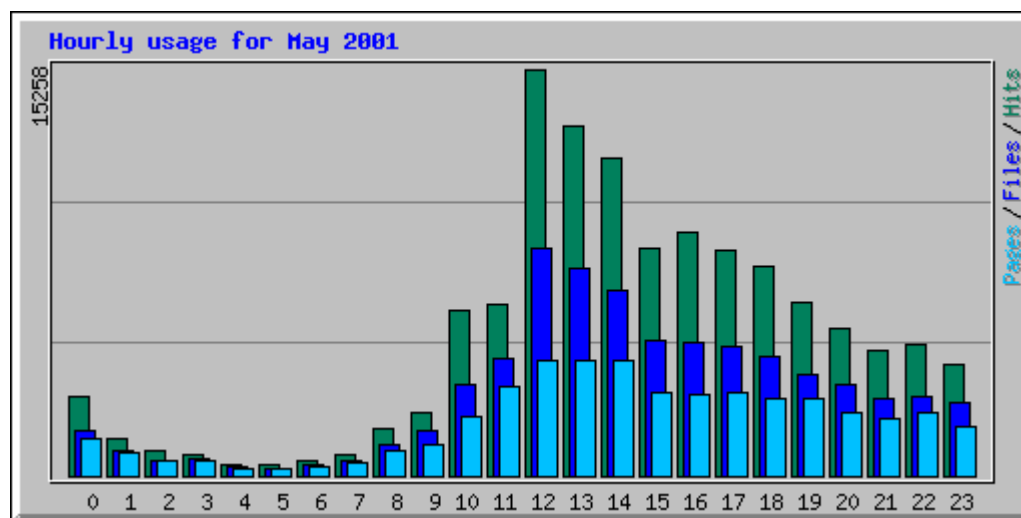


Abbildung 3: Verteilung der Zugriffe über den Tag

Ein grundsätzlich ähnliches Bild entsteht bei der Auswertung der Zugriffszahlen während des Einsatzes in Hannover im Januar/Februar 2001. Allerdings ergeben sich Unterschiede aus dem Stellenwert, den das Bearbeiten der Aufgaben innerhalb der Kurse einnimmt. In Osnabrück ist ein erfolgreiches Bearbeiten der Aufgaben Voraussetzung für die Klausurzulassung, in Hannover sind die Übungen ein freiwilliges Angebot, den klausurrelevanten Stoff vorlesungsbegleitend zu üben. Neben dem Effekt, das die durchschnittlich pro Woche aufgewendete Zeit von Teilnehmern des Osnabrücker Kurses sehr viel

höher ist, als die von Teilnehmern des Hannoveraner Kurses, stellen sich bei verpflichtenden Übungen ungleich höhere Anforderungen an Verfügbarkeit und Stabilität des Systems sowie an den personellen Aufwand zur kursbetreuenden Wartung und technischen Beratung. Vergleichbare Erfahrungen wurden auch beim Einsatz des Systems an der Universität Hildesheim im Rahmen eines Prologkurses mit ebenfalls nichtverpflichtender Übungsteilnahme im Wintersemester 1999/2000 gemacht.

Tabelle 1 stellt den tatsächlich angefallenen zeitlichen Aufwand für den Kurs in Hannover dem geschätzten Zeitaufwand für einen entsprechenden vierwöchigen Kurs mit verpflichtender Übungsteilnahme gegenüber. Durch die angegebenen Zeiten wird die gesamte technische Infrastruktur inkl. Wartung auf Serverseite sowie die inhaltliche Vorbereitung abgedeckt, nicht jedoch die inhaltliche Betreuung des Kurses, also etwa die Korrektur der abgegebenen Lösungen.

Kurs mit 20 Teilnehmern über vier Wochen	
8h	8h Vorbereitungen, Bedarfsermittlung
8h	8h Entwurf eines Kompaktkurses als Auswahl aus Skript und Aufgaben
8h	8h Einweisung vor Ort
2h	12h Einrichten eines Kurses auf vorhandenem/zusätzlichem Server
8h	8h Verfügbarmachung von Aufgabenblättern und Musterlösungen
4h	40h Wartung, Bearbeitung von Anfragen (1h/Woche bzw. 2h/Tag)
38h	Gesamtaufwand bei freiwilliger Übungsteilnahme
84h	Gesamtaufwand bei verpflichtender Übungsteilnahme

**Tabelle 1: Zeitaufwand für den Export eines Kurses mit Übungsumgebung**

Neben der grundlegenden Funktionalität für die Kursverwaltung und das Bearbeiten und Abgeben von Übungsaufgaben bietet das System die Möglichkeit, bereits beim Durcharbeiten des Skripts weiterführende Erklärungen zu erhalten und kurze Verständnisfragen zu beantworten. Aufgrund dieser Information wird ein Benutzermodell erstellt, das individuelle Unterstützung eines Lernenden ermöglichen soll. Dieses Angebot wird aber nur von einem Teil der Lernenden genutzt. 45 Studierende des Kurses waren überhaupt einmal in dem Skript angemeldet und nur 14 bearbeiteten die Verständnisfragen oder forderten zusätzliche Erklärungen. Die Ergebnisse bei der Bearbeitung der Übungsaufgaben zeigt, dass die Studenten, die dieses zusätzliche Angebot wahrnehmen zu den erfolgreichereren Studenten des Kurses gehören.

Das positive Bild, das sich durch Auswertung der Logfiles ergibt, wird durch die Rückmeldung der Tutoren bestätigt. Das System wird von Studierenden und



Vorabversion! Erschienen als: *Christoph Peylo, Claus Rollinger, Tobias Thelen: "Intelligenter Support für Übungen und den Übungsbetrieb"*, als Unterkapitel von Hauenschild, Christa; Neijdl, Wolfgang; Rollinger, Claus; Wagner, Erwin; Womser-Hacker, Christa: "Der Entwicklungsverbund: Sprache - Wissen - Information". In: Wagner, Erwin und Kindt, Michael (Hrsg.): "Virtueller Campus: Strategien - Szenarien - Studium". Münster: Waxmann, S. 462-506. 2001

Tutoren im Kurs genutzt und der Möglichkeit selbst ein Prologsystem zu installieren vorgezogen.

## Literatur

Bönnen, Carsten; Gerstenhöfer, Joachim; Hartmann, Norman; Holzwarth, Adrian; Kober, Katharina; Lenschow, Norbert; Meißner, Bastian (1999): „PLOT: Prolog-Online-Tutor“. Osnabrück: Institut für Semantische Informationsverarbeitung.

Bratko, Ivan (2000): „Prolog Programming for Artificial Intelligence“. New York: Longman.

Brusilovski, Peter (1999): „Adaptive and Intelligent Technologies for Web-based Education“. In: *Künstliche Intelligenz*, 4/1999: 19-26.

Clocksin, William; Mellish, Christopher (1994): „Programming in Prolog“. Heidelberg, Berlin: Springer.

Gust, Helmar; Tischler, Armin: „Einführung in Prolog mit g\_log“. Osnabrück: Institut für Semantische Informationsverarbeitung. Verschiedene Überarbeitungen, 1989-1999.

Peylo, Christoph; Teiken, Wilfried; Rollinger, Claus; Gust, Helmar (1999): „Der VC-Prolog-Tutor: Eine Internet-basierte Lernumgebung“. In: *Künstliche Intelligenz*, 4/1999: 32-36.

Peylo, Christoph; Thelen, Tobias; Rollinger, Claus; Gust, Helmar (2000): „A web-based intelligent educational system for prolog“. In: “Proceedings of the International Workshop on Adaptive and Intelligent Web-based Educational Systems held in Conjunction with ITS 2000 in Montreal, Canada” Osnabrück: Technical report of the Institute for Semantic Information Processing. S. 85-96

Tschorn, Patrick (2001): „Prolog – Programming in Logic“. Osnabrück: Institut für Semantische Informationsverarbeitung. <http://virtcampus.cl-ki.uni-osnabrueck.de/prolog/skript.html>