

Erfahrungen bei der Vermittlung algorithmischer Grundstrukturen im Informatikunterricht der Realschule mit einem Robotersystem

Bernhard Wiesner, Torsten Brinda

Didaktik der Informatik
Universität Erlangen-Nürnberg
Martensstr. 3
91058 Erlangen
{wiesner,brinda}@informatik.uni-erlangen.de

Abstract: Informatische Unterrichtsinhalte lassen sich erfolgreicher vermitteln, wenn Anwendungsbezüge zur Erfahrungswelt und zum Interessensfeld der Lernenden geschaffen werden. In diesem Zusammenhang richtet sich das Interesse auf einfache Robotersysteme als Unterrichtsmedium und als Unterrichtsgegenstand. Vorgestellt werden die Ergebnisse einer Fallstudie, die exemplarisch untersucht, inwieweit ein Robotersystem als Unterrichtsmedium die Vermittlung von Informatik Inhalten der Sekundarstufe I unterstützen kann.

1 Motivation

Der Algorithmusbegriff gilt als fundamentale Idee der Informatik. Dementsprechend ist es ein Ziel des Informatikunterrichts, bei Schülerinnen und Schülern Verständnis für das Entwickeln von Algorithmen zur Lösung von Problemen bzw. Problemklassen zu wecken [SS04], [Hu04]. Dieses Vorhaben wurde in der Vergangenheit unter anderem über das Modellieren einfacher Abläufe und anschließendes Implementieren in einer Programmiersprache umgesetzt. Es zeigte sich dabei oft, dass das Erlernen der dazu erforderlichen Programmiersprache einen Großteil der zur Verfügung stehenden Zeit verschlang, ebenso wie die Einarbeitung in die notwendige Programmierumgebung. Darüber hinaus ging die Motivation der Schülerinnen und Schüler für das Entwerfen, Codieren und Fehlersuchen im Lauf der Zeit innerhalb der Klasse erheblich auseinander. Während auf der einen Seite Lernende mit den Programmierwerkzeugen motiviert arbeiteten und über den Unterricht hinaus kreativ mit der neuen Technik umgingen, kam eine große Gruppe über den rein reproduzierenden Umgang mit dem Thema kaum hinaus.

Eine Möglichkeit solche Schwierigkeiten zu vermeiden besteht darin, algorithmisierende Vorgehensweisen in andere Kontexte zu verlegen. Strukturierte Handlungsvorschriften werden in verschiedenen Unterrichtsfächern entwickelt, beispielsweise im Mathematikunterricht der Sekundarstufe I bei der Determinantenberechnung von linearen Gleichungssystemen (8. Jgst) oder im Informatikunterricht beim Umgang mit Tabellenkalku-

lationsprogrammen. Die Berechnungsvorschriften beschränken sich hierbei jedoch auf das Berechnen von Formelwerten. Ein Verständnis für die Strukturierung von Abläufen wird damit noch nicht erreicht.

Ein weiterer Ansatz verfolgt den Gedanken, statt Standard-Programmiersprachen „Minilanguages“ im Unterricht zu verwenden, die mit auf das Wesentliche beschränkter Syntax und Entwicklungsumgebung bereitgestellt werden. Damit erwartet man ein grundlegendes Verständnis der Schülerinnen und Schüler für die Entwicklung von Algorithmen und andererseits eine Verminderung der Misserfolge beim Implementieren. Konkret handelt es sich dabei um Umgebungen, die Variationen der Grundidee des programmierbaren, virtuellen Roboters „Karel“ [Pa81] darstellen, z. B. [Fr04]. Vorteile sind die einfache Inbetriebnahme des Systems und geringe bzw. keine Kosten, nachteilig die Tatsache, dass es sich dabei um abstrakte Simulationen mit reinem Übungscharakter handelt.

Für die Realschule, die bei der Auswahl der Unterrichtsmedien und -methoden grundsätzlich Praxisbezug und Handlungsorientierung anstrebt (z. B. [BUK01]), erscheinen Lösungen besonders wertvoll, bei denen reale Gegenstände als Medien den Lernprozess unterstützen. In diesem Zusammenhang sind Robotersysteme von besonderem Interesse, speziell solche, die nicht nur Eigenschaften und Verhaltensweisen realer technischer Geräte repräsentieren, sondern auch eine didaktisch gut aufbereitete Entwicklungsumgebung mitbringen.

2 Robotersysteme als Unterrichtsmedien

2.1 Überblick

Mit der Verwendung von Robotersystemen im schulischen Umfeld werden unterschiedliche Ziele verfolgt. Zum einen sollen sie dazu beitragen, Jugendlichen einen Zugang zur Technik in ganz allgemeinem Sinn zu verschaffen. Dieser Grundgedanke manifestiert sich in unterschiedlicher Ausprägung. Abhängig von Alter und Interessenlage werden Lösungen mit Baukastensystemen¹, mit festen Selbstbausätzen² bis hin zu kompletten Selbstbauten [PPA06] favorisiert. Dabei geht es außer dem Konstruieren eines technischen Geräts, unter Umständen auch mit viel Feinwerkarbeit, vor allem um das Ergründen der Möglichkeiten, die Roboter generell bieten können. Die Schülerinnen und Schüler sollen erkennen, wie Roboter mit Sensoren und Aktoren und einer programmierten Steuerung in Wechselwirkung zu ihrer Umwelt treten.

Eine wesentliche Erweiterung dieses Leitgedankens ist in den vielfältigen Formen von Roboterwettbewerben zu finden. Wettbewerbe fördern über den Umgang mit technischem Gerät hinaus zusätzliche Erziehungsziele wie Teamfähigkeit und Wettbewerbs-

¹ Z. B. LEGO Education – Mindstorms, <http://www1.lego.com/eng/education/mindstorms/default.asp>;
Fischer Technik RoboPro, <http://www.fischertechnik.de/de/computing/index.html> (beide zuletzt geprüft am 26.01.2007).

² Z. B. DLR School_Lab – Robotik: Experimente, http://www.dlr.de/schoollab/desktopdefault.aspx/tabid-1991/2841_read-4404/ (zuletzt geprüft am 26.01.2007); Conrad Electronic GmbH: Hauptkatalog 2006/2007, Hirschau; S. 1060.

verhalten [Kr05]. Sie sind allerdings keine spezifische Domäne der Robotik, sondern in vielen Disziplinen zu finden.

In der Hochschullehre werden Robotersysteme immer häufiger eingesetzt, um die Verflechtung von Technik und Informatik am konkreten Beispiel zu vermitteln und exemplarisch Projektarbeiten in fachübergreifenden Teams zu erstellen [KQ04, S. 3f] sowie um Studierenden Anwendungsfälle für das Modellieren und Implementieren in höheren Sprachen zu geben [Fa00]. Zum letztgenannten Zweck werden sie auch im Informatikunterricht der Sekundarstufe II verwendet. Motivation für den Einsatz von Robotersystemen in der Sekundarstufe I ist die Förderung naturwissenschaftlich-technischer Bildung, beispielsweise in Verbindung mit dem Projekt „Roberta“ (vgl. [Te05]), wobei meist Mädchen als Zielgruppe im Vordergrund stehen, und die Vermittlung informatischer Inhalte im Rahmen des regulären Informatikunterrichts (z. B. [DR01], [Mü02]).

2.2 Umsetzung didaktischer Prinzipien

Grundsätzlich unterstützt das Arbeiten mit Robotersystemen eine Reihe von didaktischen Prinzipien guten Unterrichts [Sa03, S. 3ff].

- **Situiertheit:** Die Lernenden gehen unmittelbar mit dem Roboter als Gegenstand der Aufgabe um und sind dadurch stets mit einem konkreten Fall befasst. Die hochmotivierende Wirkung wird mehrfach bestätigt. Insbesondere gelingt es damit Schülerinnen wie Schüler gleichermaßen anzusprechen [MRH00, S. 4], [Mü05, S. 147].
- **Selbststeuerung:** Die Modellierungs- und Implementierungsaufgaben lassen sich beim Arbeiten mit Robotersystemen oft so gestalten, dass die Lernenden ihre Produkte, ihre Vorgehensweise und ihr Arbeitstempo selbst bestimmen. Sie können aufgrund des System-Feedbacks ihren eigenen Arbeitsfortschritt feststellen und eignen sich das grundlegende Prinzip „Aus (Programmier-)Fehlern lernen“ an [MRH00, S. 4].
- **Soziale Einbettung:** Die Lernenden erarbeiten den überwiegenden Teil der Lerninhalte kooperativ in ihren Teams und tauschen während des Testens ihrer Entwürfe Erfahrungen mit anderen Teams aus. Die Lehrperson ist überwiegend beratend tätig.
- **Vielfältige Anschlussmöglichkeiten:** Der direkte Anwendungsbezug von Modellierung und Programmierung sowie der Umgang mit den Geräten sprechen verschiedene Sinne an und berücksichtigen vielfältige Lernertypen. Das problemorientierte Lernen ermöglicht die Entwicklung nützlicher Lernstrategien. Die Lernenden erwerben Wissen und wenden es gleichzeitig an.

Es ist unmittelbar ersichtlich, dass die genannten Vorteile nur bei einer soliden didaktisch-methodischen Unterrichtskonzeption zum Tragen kommen.

2.3 Robotersysteme für den Informatikunterricht

An Robotersysteme für den Informatikunterricht sind andere Anforderungen als beispielsweise an Geräte zu stellen, die in Wettbewerben erfolgreich sein sollen. Schnelligkeit und Präzision der Mechanik sind weniger entscheidend als eine gut strukturierte Programmiermethodik und eine Realisierung mit vertretbarem Zeit- und Organisationsaufwand. Gerade dieser Punkt stellt immer wieder ein Hemmnis für den Einsatz im Pflichtunterricht dar.

Aus didaktischen wie auch finanziellen Gründen ist es vorteilhaft, wenn das System in mehreren Jahrgangsstufen eingesetzt werden kann. Es sollte als Werkzeug den Schülerinnen und Schülern vertraut sein und im Idealfall eine Vielzahl von Lernzielen in unterschiedlichen Altersgruppen unterstützen.

Das Angebot an entsprechenden Systemen ist gegenwärtig klein. Für den Unterricht im Klassenverbund scheiden Geräte aus, zu deren Erstellung Lötarbeiten an kleinen Elektronikbauteilen notwendig sind. Die Bewältigung solcher Arbeiten ist nicht Unterrichtsziel der Sekundarstufe I. Ganz abgesehen davon wäre der organisatorische Aufwand nicht vertretbar. Andererseits müssen die Geräte von den Lernenden auf einfache Weise programmierbar sein. Damit kommen manche Fertigsysteme nicht in Betracht, die lediglich Robotereigenschaften demonstrieren oder nur fernsteuerbar sind.

Zu den Baukastensystemen zählen die Robotersysteme LEGO Mindstorms Robotic Invention System (RIS) und dessen Nachfolger LEGO Mindstorms NXT, die nach dem LEGO-Prinzip durch einfaches Zusammenstecken den Bau von Modellen mit Roboter-eigenschaften erlauben. Damit bleibt einerseits das Selbstbauprinzip mit seinen oben genannten didaktischen Vorteilen erhalten, andererseits bleiben den Lernenden mechanisch aufwändige Konstruktionen und Lötarbeiten erspart.

LEGO Mindstorms RIS lässt sich visuell mittels RCX-Code programmieren. Die grafischen Symbole der Programmierumgebung repräsentieren stets vollständige algorithmische Grundelemente wie etwa Wiederholung oder bedingte Anweisung. Entsprechend der gestellten Aufgabe ordnen die Lernenden die notwendigen Symbole auf der Oberfläche an und stellen gegebenenfalls erforderliche Parameter in geeigneter Weise ein, beispielsweise Motordrehrichtungen. Damit können Syntaxfehler, wie sie aus der Skriptprogrammierung bekannt sind, etwa fehlende Zeichen, nicht entstehen. Die Programme sind grundsätzlich lauffähig und die Lernenden können sich auf die semantischen Probleme ihrer Konstruktion konzentrieren. Ein zusätzlicher Vorteil der grafischen Programmierung ist die Ähnlichkeit der Programme mit Struktogrammen. Diese Eigenschaften lassen RIS für den Einsatz in der Sekundarstufe I als geeignet erscheinen.

Weiterführende Möglichkeiten der Programmierung bieten RoboLab³, das auf LabVIEW basiert, und NQC (Not Quite C) [Ba03]. Nach entsprechender Änderung der Roboter-Firmware lässt sich das RIS-System auch in Java, FORTH, ADA, C oder LISP programmieren [Fa00, S. 148ff], [Ba03, S. 363ff], [K104].

³ LEGO Education Mindstorms: LMS Concept http://www.lego.com/eng/education/mindstorms/home.asp?pagename=lmfsc&l2id=4_4 (zuletzt geprüft am 28.01.2007)

2.4 Vermittelbare Informatik-Kompetenzen

Roboter sind Informatiksysteme, die dazu geeignet sind, um mit ihnen Problemlösekompetenzen im weitesten Sinn zu erwerben. Gegenüber reinen Softwareprodukten zeichnen sie die mechanisch-technischen Komponenten aus, die auf jugendliche Lernende attraktiv und motivierend wirken (vgl. [MRH00, S. 4]).

Programmierbare Robotersysteme sind dazu geeignet, die Prozesskette Analysieren, Modellieren, Implementieren und Reflektieren handlungsorientiert durchzuführen, wobei sich dies zielgruppenspezifisch variieren lässt: In der Sekundarstufe I beispielsweise analysieren Lernende Aktionen, welche die Roboter ausführen sollen, zerlegen sie in elementare Handlungsschritte und gelangen damit auf natürliche Weise zur Modellierung von Abläufen. Weiterhin konnte bereits exemplarisch gezeigt werden, wie sich objektorientiertes Modellieren [DR01], ereignisgesteuerte Abläufe und zustandsorientiertes Modellieren [RNH04] mit Robotersystemen als Medien verknüpfen lassen.

In Verbindung mit der grafischen Programmiersprache RCX-Code bzw. dem Nachfolger NXT Software erscheint das Mindstorms-System speziell zur Einführung des Algorithmusbegriffs geeignet. Die Lernenden analysieren die gegebene Aufgabe, die ihr Roboter bewältigen soll, entwickeln und beschreiben Lösungen, zerlegen den geplanten Ablauf in elementare Schritte, stellen ihn in Diagrammform dar und generieren Programme aus algorithmischen Grundbausteinen. Anschließend testen sie, korrigieren Fehler und beurteilen ihre Produkte.

Zusätzlich gewinnen die Schülerinnen und Schüler bei der Arbeit mit den Robotern eine Reihe weiterer Kompetenzen. Dazu zählen die Fähigkeit zum Entwurf von Systemen, Komponenten und Prozessen, die vorgegebene Eigenschaften besitzen, die Fähigkeit zur effektiver Kooperation und Kommunikation im Team sowie die Fähigkeit, mit neuartigen Geräten und Softwareanwendungen umzugehen (vgl. [KQ04, S. 4]).

2.5 Probleme und offene Fragen

Es ist denkbar, dass der angestrebte Lernerfolg bei der Verwendung von Robotersystemen durch Randbedingungen beeinträchtigt wird, die der Einsatz von Baukästen im Unterricht mit sich bringt. Hier ist vor allem der erhöhte Zeitbedarf zu nennen, den das Bauen der Roboter und das Einarbeiten in ihre Programmierung erfordern.

Aufgrund ihrer Konstruktion sind Roboter, die durch Zusammenstecken einzelner LEGO-Steine gebaut wurden, mechanisch anfällig. Es ist die Frage zu stellen, in welchem Maß das Erhalten eines verlässlichen Betriebszustands sowie das immer erforderliche Anpassen von Sensorwerten und Motordrehzeiten den Unterricht behindern [Ku04, S. 7], [FM02, S. 13].

Ferner ist ungewiss, ob sich die hohe Motivation, mit der Schülerinnen und Schüler das Thema „Roboter“ beginnen, über eine längere Unterrichtssequenz erhalten lässt [PMT03, S. 5] und inwieweit Roboter als Medien die Aufmerksamkeit der Lernenden dominieren und damit die eigentlichen Lernziele zu sehr in den Hintergrund drängen. Um zu diesen Fragen erste Antworten zu erhalten, wurde an einer bayerischen Realschule nachfolgende Fallstudie durchgeführt.

3 Fallstudie

3.1 Organisatorisches Konzept

Informatik ist an der Realschule in Bayern ein Wahlpflichtfach mit einem Stundenmaß von 2 Wochenstunden von der 8. bis zur 10. Jahrgangsstufe. Als Variante wird in einem seit 2001 dauernden Modellversuch das Wahlpflichtfach Informationstechnologie gelehrt, das je nach Wahlpflichtfächergruppe Inhalte aus Informatik, Textverarbeitung, Technischem Zeichnen (CAD) und Rechnungswesen enthält. Für den IT-Unterricht wird eine Klasse üblicherweise in zwei Gruppen geteilt. Die unterrichtete Gruppe bestand aus 4 Schülerinnen und 10 Schülern, die für die Unterrichtssequenz 7 Zweiergruppen zugeordnet wurden, da insgesamt 7 Arbeitsplätze mit je einem Rechner und je einem Mindstorms-Baukasten zur Verfügung standen.

Im Rahmen eines studienbegleitenden fachdidaktischen Praktikums wurde die Fallstudie in der Zeit von Oktober 2006 bis Januar 2007 an einer Realschule im Raum Erlangen-Nürnberg zusammen mit 3 Informatik-Lehramtsstudierenden durchgeführt. Die Unterrichtssequenz umfasste 9 Unterrichtsstunden (Doppelstunden) in einer 9. Jahrgangsstufe. Die Unterrichtsstunden wurden gemeinsam mit den Studierenden in einem Begleitseminar konzipiert und von jeweils einem Studierenden im wöchentlichen Wechsel gehalten. Die anderen Praktikumsmitglieder, die Autoren und eine weitere Lehrkraft beobachteten den Unterricht als Hospitierende. Bei der Evaluation im Anschluss an den jeweiligen Unterricht wurden Verbesserungsvorschläge entwickelt, die im nachfolgenden Erfahrungsbericht eingearbeitet sind.

Zusätzlich zur Beobachtung und Dokumentation des Unterrichtsgeschehens wurden die Leistungen der Schülerinnen und Schüler erfasst. Zu Beginn der dritten Unterrichtseinheit fand eine schriftliche Lernerfolgskontrolle im Umfang von 10 Minuten statt. Am Ende der Reihe bearbeiteten die Lernenden schriftliche Aufgaben im Rahmen eines Übungszirkels. Die Beobachtungsergebnisse sowie die Ergebnisse der Lernerfolgskontrollen wurden für die Auswertung verwendet.

3.2 Inhaltliche und methodische Konzeption

Die Ziele der Unterrichtssequenz orientierten sich am Lehrplan. Die Schülerinnen und Schüler sollten wissen, was Roboter sind und verstehen, dass ihr Verhalten von einem Programm gesteuert wird und dass Sensoren den Programmablauf beeinflussen. Sie sollten die algorithmischen Strukturen Sequenz, Wiederholung, Verzweigung kennen und im Zusammenhang mit dem benutzten Roboterfahrzeug zur Lösung von Aufgaben verwenden können. Ferner sollten sie den Algorithmusbegriff als genau definierte Handlungsvorschrift zur Lösung eines Problems oder einer bestimmten Art von Problemen kennen und einfache Aktionen in Struktogrammform beschreiben können.

Die Inhalte und ihre Verteilung auf die Unterrichtseinheiten (UE) werden in Tabelle 1 stichwortartig wiedergegeben. Eine vollständige Dokumentation der Reihe ist unter <http://ddi.informatik.uni-erlangen.de/Lehre/WS200607/ddifapra/material/> zu finden.

1	Einsatzbereiche und Eigenschaften von Robotern; Definition des Roboterbegriffs; Präsentation von LEGO-Robotern; Bau des Roboterfahrzeugs; Ausführung eines vorgegebenen Programms und Anfertigung einer genauen Beschreibung des Bewegungsablaufs
2	Simulation eines Roboters in Rollenspielform zur Einführung der Begriffe „Anweisung“ und „Befehlssatz“; Definition des Begriffs „Sequenz“; Einführung in die Programmierung mit RCX-Code; Erstellung eines Programms mit rein sequentieller Struktur; Einführung der Wiederholung als verkürzendes Verfahren
3	Simulation eines Roboters in Rollenspielform zur Einführung der bedingten Anweisung; Erweiterung des Rollenspiels in mehreren Varianten; Einbau eines Lichtsensors in das Roboterfahrzeug; Modellierung und Programmierung der Aufgabenvarianten aus dem Rollenspiel; Einführung des Begriffs „bedingte Anweisung“; Erweiterung der Aufgabe um ein Lenkmanöver beim Unterschreiten eines bestimmten Lichtwerts; Einführung des Begriffs „Verzweigung“
4	Aufgabe, den Roboter so lange jeweils 1 Sekunde geradeaus fahren zulassen, wie der Messwert des Lichtsensors einen hellen Untergrund meldet; Erklärung der Verknüpfung von Sensormesswerten mit der Wiederholungsanweisung in RCX-Code; Erklärung der „bedingten Wiederholung“; Vergleich dieses Verfahrens mit der ähnlich lautenden Aufgabe aus der Vorstunde
5	„Übersetzung“ des RCX-Programms aus der Vorstunde in eine umgangssprachliche Beschreibung; Erzeugung eines Struktogramms zum Programm aus der Vorstunde; Definition des Begriffs Struktogramm und seiner Elemente; Erzeugung eines Struktogramms zu einer umgangssprachlichen Beschreibung einer Folge von Roboterbewegungen; Realisierung mittels RCX-Code
6, 7	Wiederholung und Festigung der Lernziele in Form eines Übungszirkels
8	Aufgabenstellung zum Erreichen eines Labyrinth-Ausgangs; Diskussion mehrerer Lösungsverfahren; vergleichende Abschätzung des Zeitaufwands; Definition des Algorithmusbegriffs; Analyse und Detailplanung eines Lösungsverfahrens; Programmierung und Test; Demonstration der Lösungen mit Wettbewerbscharakter
9	Erstellung einer schriftlichen Zusammenfassung der Unterrichtssequenz in Form von bearbeiteten, ausführlich kommentierten Aufgaben durch die Schüler

Tabelle 1: Unterrichtseinheiten und Lerninhalte

Das Bauen und Programmieren eines Roboters wird häufig, wie auch im Lehrplan vorgeschlagen, als Projektarbeit im Informatikunterricht durchgeführt. Dabei planen, bauen und programmieren die einzelnen Schülerteams ihre Geräte, wobei ihnen ein verhältnismäßig großer gestalterischer Freiraum gewährt wird. Die Lernenden arbeiten über einen längeren Zeitraum selbstständig, dokumentieren ihre Arbeit und präsentieren am Ende ihre individuellen Ergebnisse. Da in dieser Unterrichtssequenz jedoch bestimmte, fest umrissene Inhalte vermittelt werden sollten, wurden die Phasen der Gruppenarbeit kürzer gefasst und in andere Unterrichtsformen, wie Lehrerpräsentation und Unterrichtsgespräch, eingebettet.

Die Vermittlung neuer Lerninhalte erfolgte, indem den Lernenden zunächst Aufgaben gestellt wurden, zu deren Lösung neue Kenntnisse oder neue Fähigkeiten erforderlich waren. Im Plenum oder in den Teams wurden dann Lösungsansätze gesucht und diskutiert. Anschließend machten sich die Schülerinnen und Schüler im Team an die Bewältigung der Aufgaben. Die Lehrperson leistete bei Gruppen, die längere Zeit vergeblich an der Lösung arbeiteten, Hilfestellung. Im Anschluss daran wurden jeweils die neuen Erkenntnisse schriftlich festgehalten. Diese Methode erlaubte den Lernenden, in gewissem Umfang selbst aktiv neue Inhalte zu erarbeiten (vgl. auch [HNR06, S. 92]).

Zur Festigung der Lerninhalte wurde ein Übungszirkel gestaltet (Stunde 6 und 7). Die Schüler hatten insgesamt sechs Aufgaben, die sich jeweils auf unterschiedliche Themen der Unterrichtssequenz bezogen, sowohl schriftlich als auch am Rechner zu bearbeiten (siehe <http://ddi.informatik.uni-erlangen.de/Lehre/WS200607/ddifapra/material/>). Dieses Verfahren diente der Förderung des didaktischen Prinzips der Selbststeuerung, was die Reihenfolge, das Arbeitstempo und auch in gewissem Umfang die Auswahl der Aufgaben anging.

4 Ergebnisse und Erfahrungen

4.1 Formales Beschreiben von Abläufen, Ablaufdiagramme

In der ersten Stunde wurde von den Schülerinnen und Schülern die detaillierte Beschreibung einer vom Roboter ausgeführten Aktion erwartet. Die Bewegungen des Roboterfahrzeugs waren klar voneinander zu unterscheiden und wurden meist streng sequentiell ausgeführt. Der Detaillierungsgrad der Beschreibungen fiel allerdings sehr unterschiedlich aus. Eine Drehbewegung des Fahrzeugs wurde beispielsweise so beschrieben: „Die linken Räder drehen sich kurz rückwärts und die rechten Räder drehen sich kurz vorwärts“, bis hin zu „Der Roboter fährt nach links.“ Dies trat nicht mehr auf, sobald die Lernenden die Programmierumgebung und die Symbole des RCX-Codes kennen gelernt hatten, da dann der Detaillierungsgrad der Anweisungssymbole übernommen wurde.

Um deutlich zu machen, dass ein Roboter nur eine beschränkte und fest definierte Menge an Methoden hat, erwies sich die Simulation in Form eines Rollenspiels als sehr hilfreich: Einem Schüler wurde die Rolle eines Roboterfahrzeugs zugewiesen. Ein zweiter Schüler bekam die Rolle des Anweisungsgebers. Er musste unter Beschränkung auf die zur Verfügung stehenden Methoden, die „Befehlssatz des Roboters“ genannt wurden, den Roboter leiten. Die gestellte Aufgabe lautete etwa: Der Roboter steht zwischen den Tischen des Klassenzimmers und soll bis zur Tür gehen. Der Befehlssatz ist beschränkt auf „Gehe einen Schritt vorwärts“, „Drehe dich um 90 Grad nach rechts“, „Drehe dich um 90 Grad nach links“. Der Spielverlauf zeigte, dass sich sowohl Anweisungsgeber als auch Roboter nicht immer an den vorgegebenen Befehlssatz hielten. Dem wurde entgegengesteuert, indem ein dritter Schüler die Einhaltung der Formulierungen überwachte. Er war mit einer Hupe ausgestattet und intervenierte, sobald eine nicht vorhandene Methode angefordert wurde.

Die Darstellung von Abläufen mit Diagrammen wurde erst behandelt, als die Lernenden einen ausreichenden Überblick über die algorithmischen Grundelemente und die entsprechenden RCX-Icons gewonnen hatten. Die verwendete Programmierumgebung erwies sich an dieser Stelle als vorteilhaft, da die grafische Programmdarstellung große Ähnlichkeit mit Struktogrammen aufwies. Daraus folgte zum Ersten die Entscheidung für die Verwendung von Struktogrammen als formale Darstellung für die Aufgaben der Roboter, zum Zweiten die Nutzung der Programm-Analyse als einführende Methode. Auf die Modellierung mit Hilfe von Aktivitätsdiagrammen wurde verzichtet, da die Transferanforderungen beim Übergang vom Diagramm zum Programmcode wesentlich größer sind.

Unter Hinweis auf eine produktunabhängige grafische Generalisierung erfolgte der Übergang von der grafischen Darstellung im RCX-Code zu einer ersten Version des Struktogramms. Einfache Anweisungen und bedingte Anweisungen (Verzweigungen) bereiteten dabei keinerlei Schwierigkeiten, da ihre Darstellungen fast direkt aus RCX in ein Struktogramm übertragen werden konnten. Wiederholungen dagegen erforderten eine gewisse Umgestaltung, die mit dem Hinweis auf Darstellungsnormen (DIN 66261) eingeführt wurde.

4.2 Anwendung der algorithmischen Grundelemente

Die erste Programmieraufgabe für das Roboterfahrzeug und Hinführung zum Begriff der Sequenz war das Umfahren eines Quadrats von ca. 30 cm Kantenlänge. Nach der Vorbereitung durch das Rollenspiel (vgl. Tab. 1, UE 2) gelang den Schülerinnen und Schülern die Zerlegung der Aufgabe in die einzelnen Anweisungen ohne Probleme. Die Aufgabe bereitete auch gleichzeitig den nächsten Lerninhalt vor, die Wiederholung als algorithmisches Grundelement. Hierzu wurde das bisher erstellte Programm untersucht und festgestellt, dass sich einzelne Teile der Sequenz, nämlich die Geradeausfahrt und anschließende Drehung genau viermal wiederholen. Nachdem die Wiederholungsanweisung im RCX-Code identifiziert war, wurde die Aufgabe damit neu gelöst. Der daraus resultierende Vorteil war für die Schüler unmittelbar einsichtig: Eventuell nötige Änderungen an den Fahr-Anweisungen oder Drehungen müssen bei Anwendung der Wiederholungsstruktur nur einmal für alle vier Quadratseiten vorgenommen werden.

Nach der Erklärung des Symbols für die bedingte Anweisung programmierten die Teams die Abläufe des Rollenspiels selbstständig im RCX-Code (vgl. Tab. 1, UE 3). Es erwies sich als wichtig, die Abfolge der gespielten Varianten genau einzuhalten. Teams, die mit den komplexen Varianten begonnen hatten, verloren den Zusammenhang zwischen dem Ablauf der Aufgabe und der hieraus folgenden notwendigen Anordnung der Programm-elemente.

Bei der Einführung der bedingten Wiederholung (vgl. Tab. 1, UE 4) wurde auf die Unterscheidung zwischen Wiederholung mit Anfangs- und Endebedingung verzichtet. In Musterlösungen wurde ausschließlich die Wiederholung mit Anfangsbedingung verwendet.

4.3 Hard- und Software

Das Bauen nach einer genauen Anleitung erwies sich als vorteilhaft, da damit die Funktionssicherheit erreicht wurde, die im Unterrichtsbetrieb erforderlich ist (vgl. [Ku04, S. 12]). Da die Produkte alle den gleichen Fahrzeugaufbau besaßen, mussten gemeinsam entwickelte Programmkomponenten nicht an individuelle Konstruktionen angepasst werden.

Zum Bau der Fahrzeuge wurden an die Gruppen genau die Bauteile herausgegeben, die für die Konstruktion nötig waren. Das vorherige Abzählen der Bauteile bedeutete zwar einen erhöhten Vorbereitungsaufwand, führte aber zu einer geordneten Konstruktionsphase, da die Schülerinnen und Schüler die benötigten Bauteile nicht aus einer großen Menge heraussuchen mussten und keine Bauteile verloren gingen. Die Fahrzeuge blieben den einzelnen Teams während der gesamten Unterrichtssequenz zugeordnet.

Die Programmierumgebung bietet mehrere Symbole, die recht ähnlich aussehen und daher für den Anfänger in ihrer Funktionalität schlecht unterscheidbar sind. So gibt es beispielsweise das Symbol „Motor(en) ein“ und das Symbol „Motor(en) ein für x Sekunden“. Im zweiten Fall wird die Programmausführung für die Zeit x angehalten, im ersten Fall nicht, was zu unerwarteten Reaktionen führt. Hier wäre es vorteilhaft, die zur Verfügung stehenden Symbole per Voreinstellung einschränken zu können.

Es zeigte sich, dass in der Basisausstattung des RIS-Systems aufgrund fehlender Rotationssensoren nur Zeitdauern und Motorleistungen für den Betrieb der Motoren angegeben werden können, nicht etwa Umdrehungszahlen. Damit wurde das Programmieren von Drehungen um feste Winkel zu einem Geduldsspiel, da vor allem die Reibung der Räder auf dem Boden schlecht kalkulierbar war. Dieser Nachteil ist in der Nachfolgeverson Mindstorms NXT behoben.

Mit der Verwendung paralleler Programmstränge lassen sich im verwendeten System Reaktionen auf Sensormeldungen eleganter als bisher beschrieben realisieren. Dabei wird eine Bedingung zu einem Sensorwert formuliert, bei deren Eintreten das Programm im Parallelstrang (z. B. Lenkmanöver) Priorität vor dem Programm im Hauptstrang (z. B. Geradeausfahrt) erhält. Anschließend kehrt die Ausführung wieder zum Hauptprogramm zurück. Da sich die beschriebene Unterrichtssequenz aber ausschließlich mit der Behandlung grundlegender algorithmischer Strukturen befasste, wurde diese Methode nicht verwendet. Auch hier würde es die Unterrichtspraxis erleichtern, wenn sich die entsprechenden Symbole, wie schon oben geschildert, ausblenden ließen.

4.4 Unterrichtsmethoden

Die Simulation von Handlungsabläufen in Form von Rollenspielen hat sich als ausgesprochen hilfreich erwiesen. Das schrittweise „Durchspielen“ der Algorithmen brachte das notwendige Verständnis für die genaue Abfolge der Anweisungen und deckte dabei Verständnisschwierigkeiten der Lernenden auf, die der Lehrperson beim Formulieren der Aufgaben zum Teil nicht bewusst waren.

Erfahrungsgemäß besteht bei Gruppenarbeit, besonders bei größeren Gruppen, die Gefahr, dass sich Einzelne aus der gemeinsamen Bearbeitung „ausklinken“. Dies war im

vorliegenden Fall nicht zu beobachten. Die Schülerinnen und Schüler arbeiteten in Zweierteams. Zudem wechselte die Unterrichtsform innerhalb einer Stunde mehrfach zwischen Gruppenarbeit und Unterrichtsgespräch, in das alle Schülerinnen und Schüler gleichermaßen eingebunden wurden.

Der gegen Ende durchgeführte Übungszirkel erwies sich als wirksame Methode zur Festigung aller Lerninhalte. Die Teams arbeiteten selbstständig und mit wachsendem Engagement am Aufgabenmaterial. Die Gruppenarbeit ermöglichte es der Lehrperson, während des Unterrichts die Teams bei Fragen zu beraten und ihre Lösungen zu beurteilen.

5 Fazit und Ausblick

Mit der Fallstudie konnte am Thema „Algorithmische Grundstrukturen“ exemplarisch gezeigt werden, wie die Verwendung von Robotersystemen die Vermittlung informatischer Inhalte im Unterricht der Realschule unterstützt. Mit dem Hinzufügen dieses „begreifbaren“ Unterrichtsmediums werden unterschiedliche Lerntypen gefördert und ein Leitgedanke der Realschule umgesetzt. Gleichzeitig wird ein Praxisbezug für die zu vermittelnden informatischen Sachverhalte geschaffen, der die Auseinandersetzung der Lernenden mit der Thematik deutlich intensiviert. Das verwendete System zeichnet sich zusätzlich dadurch aus, dass es die Entwicklung des Modellierungsprozesses, im dargestellten Fall die Modellierung von Abläufen, sehr einfach macht. Es konnte gezeigt werden, dass sich Bau und Verwendung der Roboter durch das in Kap. 4.3 dargestellte Verfahren problemlos in den Unterrichtsbetrieb einfügen. Die in Kap. 2.5 genannten mechanischen Schwächen traten auch hier zutage, wurden aber von den Lernenden nicht als hinderlich empfunden. Das Interesse der Schülerinnen und Schüler am Thema wurde von den Hospitierenden über den gesamten Zeitraum als sehr hoch beurteilt. Ein Nachlassen war nicht zu beobachten.

Die Ergebnisse der Lernerfolgskontrollen zeigten ein Bild, das den sonstigen Leistungen der Klassengruppe entsprach. Der Mehrwert des geschilderten Verfahrens ist zum großen Teil darin zu sehen, dass sich auch Schülerinnen und Schüler zur Mitarbeit motivieren ließen, die dem Fach Informatik nach Aussagen ihres Informatiklehrers bisher eher ablehnend gegenüberstanden. Im Übungszirkel zeigte sich, dass die Lernenden auch einfache Algorithmen darstellen konnten, die nichts mit den Robotersystemen zu tun hatten, mithin also ein weiterführendes Verständnis für die Thematik entwickelten, das Transferleistungen ermöglicht.

Davon ausgehend erscheint es sinnvoll, die Erprobung des Konzepts weiterzuführen und auszudehnen auf die in Kap. 2.4 genannten Themenbereiche. Voraussetzung für ein erfolgreiches Arbeiten ist sicher eine didaktisch wie technisch ausgereifte Programmierumgebung. Es wäre allerdings verkehrt anzunehmen, dass allein das Benutzen des Robotersystems bereits den Lernerfolg herbeiführt. Erforderlich sind ein klar strukturiertes inhaltliches Konzept und ein methodisch gut durchdachter Unterrichtsablauf, in den die Roboterpraxis eingebunden ist.

Literaturverzeichnis

- [Ba03] Baum, D.: Definitive guide to LEGO MINDSTORMS. Apress, Berkeley Calif., 2003.
- [BUK01] Bayerisches Staatsministeriums für Unterricht und Kultus (Hrsg.): Lehrplan für die sechststufige Realschule in Bayern, Maß, München, 2001.
- [DR01] Dietzel, R.; Rinkens, T.: Eine Einführung in die Objektorientierung mit Lego Mindstorms Robotern. In: Keil-Slawik, R.; Magenheimer, J. (Hrsg.): Informatikunterricht und Medienbildung, INFOS 2001. Köllen, Bonn, 2001; S. 193–199.
- [Fa00] Fagin, B. S.: Using Ada-based robotics to teach computer science. In: Proceedings of the 5th Annual Conference on Innovation and Technology in Computer Science Education. ACM Press, New York, 2000; pp. 148–151.
- [FM02] Fagin, B. S.; Merkle, L. D.: Quantitative Analysis of the Effects of Robots on Introductory Computer Science Education. In: JERIC 2 (2002) 4; pp. 1–18.
- [Fr04] Freiberger, U.: Robot Karol, 2004,
URL: <http://www.schule.bayern.de/karol/> (zuletzt geprüft am 22.01.2007).
- [HNR06] Hartmann, W.; Näf, M.; Reichert, R.: Informatikunterricht planen und durchführen. Springer, Berlin, 2006.
- [Hu04] Hubwieser, P.: Didaktik der Informatik. Berlin, Springer, 2004.
- [KQ04] Kitts, C.; Quinn, N.: An interdisciplinary field robotics program for undergraduate computer science and engineering education. In: JERIC 4 (2004) 2; Art. 3.
- [Kl04] Klassner, F.: Enhancing lisp instruction with RCXLisp and robotics. In: Proceedings of the thirty-fifth Technical Symposium on Computer Science Education. ACM Press, New York, 2004; pp. 214–218.
- [Kr05] Kraetzschmar, G. K.: Motivation for Free - Zum Motivationsfaktor von internationalen RoboCupJunior-Wettbewerben. In: Cremers, A. B.; Manthey, R.; Martini, P.; Steinhage, V. (Hrsg.): INFORMATIK 2005 - Informatik LIVE! Band 1. Köllen, Bonn, 2005; S. 148–152.
- [Ku04] Kumar, A. N.: Three Years of Using Robots in an Artificial Intelligence Course - Lessons Learned. In: JERIC 4 (2004) 3; Art. 1.
- [MRH00] Magenheimer, J.; Reinsch, T.; Hirsch, M.: Zugänge zur Informatik mit Mindstorms. In: LOG IN 20 (2000) 2; S. 34–46.
- [Mü02] Müller, W.: Algorithmik mit dem LEGO-Roboter, 2002, URL: <http://www.lehrer-online.de/url/lego-roboter> (zuletzt geprüft am 24.01.2007).
- [Mü05] Müllerburg, M.; Börding, J.; Theidig, G.; Petersen, U.: Informatikausbildung, Roboter und Mädchen. In: Cremers, A. B.; Manthey, R.; Martini, P.; Steinhage, V. (Hrsg.): INFORMATIK 2005 - Informatik LIVE! Band 1. Köllen, Bonn, 2005; S. 143–147.
- [Pa81] Pattis, R. E.: Karel the Robot - A Gentle Introduction to the Art of Programming. 1st Edition, Wiley, New York, 1981.
- [PMT03] Petersen, U.; Müllerburg, M.; Theidig, G.: Girls and Robots - A Promising Alliance. URL: <http://alex.ais.fraunhofer.de/zeno/web?action=content&journal=16919&rootid=15465> (zuletzt geprüft am 28.01.2007)
- [PPA06] Professur für Prozessautomatisierung der TU Chemnitz (Hrsg.): Roboking 2007 Wettbewerbsdokumentation, 2006, URL: http://www.tu-chemnitz.de/etit/proaut/rk/fileadmin/user_upload/downloads/RoboKing2007_Wettbewerbsdoku.pdf (zuletzt geprüft am 22.01.2007).
- [RNH04] Reichert, R.; Nievergelt, J.; Hartmann, W.: Programmieren mit Kara. Ein spielerischer Zugang zur Informatik. Springer, Berlin, 2004.
- [Sa03] Sacher, W.: Neue Medien - neuer Unterricht?. Schulpädagogische Untersuchungen Bd. 19, Nürnberg, 2003.
- [SS04] Schubert, S.; Schwill, S.: Didaktik der Informatik. Spektrum, Berlin, 2004.
- [Te05] Tempelhoff, A.: Robotik in der Sekundarstufe I. In: LOG IN 25 (2005) 134; S. 23–29.