

## **Zugänge zur Informatik mit Mindstorms (1) - mit Lego Robotern in der Sek I spielerisch ein Informatiksystem entdecken und gestalten**

Die am MIT (Massachusetts Institute of Technology) in Kooperation mit Lego entwickelten Mindstorms Roboter sind mittlerweile dabei, weltweit zahlreiche Kinderzimmer zu erobern<sup>1</sup>. Im Internet findet man eine Vielzahl von Webseiten, auf denen 'Lego Experten' eine Fülle von Informationen zur Programmierung und zum Bau der Roboter anbieten. Der 'Spielzeug-Roboter' übt offensichtlich eine faszinierende Anziehungskraft auf Kinder und z.T. auch auf Erwachsene aus und motiviert zum spielerischen Umgang mit elementaren Fragen von Mechanik, Algorithmik, Steuerung und Automatisierung. Es gibt mittlerweile spezifische, zum Teil visuelle Programmierumgebungen für den elektronischen Baustein (RCX) des Roboters und Programmierschnittstellen zu gängigen Programmiersprachen, die eine Einbindung in professionelle Entwicklungsumgebungen ermöglichen. Kann man angesichts dieser Rahmenbedingungen Lego Mindstorms nutzen, um Schülerinnen und Schülern im Informatikunterricht 'ernsthaft' mit informatischen Problemstellungen zu befassen, ohne ihnen dabei durch übertriebene Didaktisierung den Spaß am Umgang mit den Robotern zu verderben? Ist in der Schule mit Mindstorms ein spielerischer Zugang zur Informatik möglich?

Der vorliegende Beitrag will u.a. diesen Fragen nachgehen. In einem ersten Teil werden wesentliche technische Elemente der Mindstorms Roboter vorgestellt, didaktische Forschungsfragen im Zusammenhang mit der Verwendung der Bausätze im Unterricht behandelt und die Grundzüge einer Unterrichtseinheit mit Mindstorms Robotern für den Informatikunterricht der Sek I skizziert. Hierzu wurde eine auf Visual Basic und NQC (Not Quite C) aufsetzende Software (RCC: Roboter Control Center) entwickelt, die einigen Anforderungen für den Unterrichtseinsatz in der Sek I besser Rechnung tragen soll als die standardmäßig zur Verfügung stehenden Programme und Entwicklungsumgebungen. Wesentliche, didaktisch begründete Funktionen von RCC und Grundzüge der Programmstruktur, soweit deren Verständnis für eine sinnvolle Nutzung erforderlich ist, sind ebenfalls Gegenstand des ersten Teils des Artikels. Abgeschlossen wird er mit einer kurzen Darstellung erster Praxiserfahrungen mit Schülerinnen und Schülern.

In einem zweiten Teil werden einige ausgewählte Programmiersprachen und Entwicklungsumgebungen vorgestellt, die eine Schnittstelle zum RCX bieten. Sie werden hinsichtlich ihrer Funktionalität und ihrer didaktischen Möglichkeiten für den Informatikunterricht untersucht. Außerdem soll der potenzielle Beitrag der Mindstorms für den Anfangsunterricht in der Sek II thematisiert werden.

Die hier vorgestellten Überlegungen basieren zu großen Teilen auf Ergebnissen, die von Lehrenden und Studierenden in didaktischen Seminaren während der letzten drei Semester in der AG 'Didaktik der Informatik' an der Universität Paderborn gemeinsam erarbeitet wurden, sowie auf damit assoziierten Vorträgen, Examensarbeiten und didaktischen Forschungsbemühungen.

### **Lego Mindstorms - Technik und Marktübersicht**

Lego Mindstorms wurden im Herbst 1998 erstmals in den USA einer breiten Öffentlichkeit vorgestellt. Ein Jahr später waren die ersten Sets auch in Deutschland erhältlich. Die Fachschaft Mathematik/Informatik der Universität Paderborn importierte bereits im Dezember 1998 aus England vier Mindstorms Sets, die später durch den Erwerb von Zusatzkästen (Robo Sports) und weiteren Sensoren ergänzt wurden. So konnten wir, durch eine enge Kooperation mit der Studierendenvertretung, seit mehr als eineinhalb Jahren Erfahrungen sammeln.

Heute sind auf dem Markt drei verschiedene Mindstorms-Versionen mit diversen Erweiterungsmöglichkeiten erhältlich:

- ?? Robotics Discovery Set
- ?? Droid Developer Kit
- ?? Robotics Invention System

Aus unserer Sicht kann man in der Schule nur das Robotics Invention System (RIS) sinnvoll einsetzen. Alle Baukästen haben als Zielgruppe den Heimmarkt und sollen Lego auch für die Kinder der Computerspielergeneration interessant machen. Der Schulmarkt ist für Lego nur von sekundärem Interesse und wird von der Tochtergesellschaft Lego Dacta betreut.

---

<sup>1</sup> Dies gilt angesichts der ökonomischen und sozialen Situation auf dem Globus leider kaum für Entwicklungsländer.

Das Droid Developer Kit ist Teil des Star Wars Merchandising Konzepts. Es ermöglicht den Bau eines R2-D2 Droiden, der schon Luke Skywalker und Prinzessin Amidala rettete. Allerdings besteht der Kern, der 'Micro Scout', nur aus einem eingebauten Motor, einem eingebauten Lichtsensor und sieben vorgegebenen Programmen.

Zielgruppe des Robotics Discovery Sets sind Kinder, die keinen Computer haben. Den Roboterkernel bildet der 'Scout'. Dabei handelt es sich um einen großen blauen Legostein der einen Toshiba Mikroprozessor mit 32k ROM und 1k RAM enthält. Über ein Display lässt sich der Scout direkt programmieren. Lego bezeichnet den Scout als kleinen Bruder des RCX, der Stein des Robotics Invention System (RIS).



Abb. 1: Der RCX Baustein, Herzstück der Mindstorms Roboter

Die größten Möglichkeiten bietet das RIS mit dem gelben RCX-Baustein. Der RCX-Baustein wurde von Seymour A. Pappert entwickelt. Für die Entwicklung erhielt Pappert am MIT finanzielle Unterstützung der Firma Lego. Der RCX basiert auf einem Hitachi HD6433292 Prozessor mit 32k RAM. Es stehen jeweils drei Ein- und Ausgänge zur Verfügung. Dem RIS-Kasten liegen zwei Motoren, zwei Tastsensoren und ein Lichtsensor bei. Weitere Sensoren sind in den Erweiterungssets enthalten oder können einzeln über LEGO Dacta bestellt werden.

Für die Schule bieten sich unseres Erachtens nur die speziell zusammengestellten LEGO Dacta Baukästen oder der RIS-Baukasten mit dem Robo Sports Erweiterungsset an. Robo Sports enthält einen dritten Motor, der den Freiheitsgrad der Modelle erheblich vergrößert. Im Einzelfall kann das Ultimate Accessory Kit (Fernbedienung, Lampe, Dreh- und Rotationssensor) interessant sein - jedoch sind die Teile auch einzeln erhältlich.

### **Programmierschnittstellen**

Zwei Phasen müssen durchlaufen werden, um den Robotern Leben einzuhauchen: Als erstes müssen sie mit den vorhandenen Lego Technik Kenntnissen physikalisch zusammengesteckt werden. Anschließend muss man ein Programm für den Ablauf erstellen. Die Programme werden auf einem PC programmiert und über eine Infrarotschnittstelle auf den RCX-Baustein überspielt. Der Baustein kann fünf verschiedene Programme abspeichern und autark verarbeiten.

Für die Erstellung der Programme auf dem PC gibt es verschiedene Entwicklungsumgebungen:

?? Beigelegt ist dem Kasten das RIS 1.5. Man merkt der Software deutlich an, dass sie fürs Kinderzimmer entwickelt wurde. Mit ihr ist es einfach möglich, Programme zu erstellen. Kinder brauchen keine Syntaxkenntnisse und stecken die einzelnen Programmelemente wie Legosteine zusammen. Neben den traditionellen Kontrollstrukturen ermöglicht diese Software sehr schnell einen Einstieg in die Programmierung nebenläufiger Prozesse.

?? Weniger verspielt und speziell für den Unterricht konzipiert ist die Software RoboLab von Lego Dacta. RoboLab basiert auf der Software LabVIEW von National Instruments (Texas, USA). LabVIEW ist eine professionelle grafische Steuerungssoftware, die von Ingenieuren und Wissenschaftlern eingesetzt wird: 1997 benutzte die NASA bei der Programmierung der Marslandefähre Sojourner teilweise die LabVIEW Software. Von Lego Dacta ist für RoboLab auch eine englische Unterrichtsreihe erhältlich.

- ?? Neben diesen beiden offiziellen Varianten bietet Lego noch das Software Developer Kit (SDK) an. Über das ActiveX-Control 'Spirit.ocx' kann man von beliebigen Programmiersprachen (Visual Basic, Delphi, C++, Java) den RCX-Baustein ansprechen. Den Zusatz 'beliebig' muss man relativieren: Die entsprechenden Programmiersprachen müssen selbstverständlich ActiveX unterstützen. Leider ist die Originalversion des SDKs sehr kryptisch. Seit Mai 2000 ist die Pre-Alpha-Version des SDKs auf den Internetseiten von Lego Mindstorms verfügbar. Lego Script ähnelt C und soll den Kern der neuen RIS-Version bilden.
- ?? Im Rahmen eines Seminars haben wir das Roboter Control Center unter Visual Basic entwickelt, auf das wir in unserem Unterrichtsentwurf genauer eingehen werden.
- ?? Neben diesen Möglichkeiten gibt es im Internet noch zahlreiche kostenlose Programmierumgebungen. Am Interessantesten scheint die Software NQC zu sein, zu der bereits Publikationen (z.B. Knudsen 1999) erschienen sind. Ein großer Vorteil von NQC ist, dass es nicht nur unter Windows, sondern auch unter Linux oder Mac OS läuft. Darüber hinaus gibt es noch zahlreiche Skriptsprachen, wie zum Beispiel BOT-Code.

### **Didaktische Fragestellungen**

Nachdem in einem Seminar im Wintersemester 1999/2000 vor allem didaktisch-methodische Fragen des Unterrichtseinsatzes der Mindstorms im Mittelpunkt der Diskussion standen, haben wir in dem im Sommersemester 2000 veranstalteten Seminar verschiedene Entwicklungsumgebungen untersucht und deren Schultauglichkeit getestet. In den beiden Seminaren wurde ein breites Spektrum von interessanten Gesichtspunkten zur unterrichtlichen Verwendung der Mindstorms Roboter diskutiert. Hier eine Auswahl von Fragestellungen und eine knappe Zusammenfassung der im Seminar erarbeiteten Ergebnisse:

#### ***Bedeutung des Themas 'Roboter' für den Informatikunterricht in der Sek I***

Die Beschäftigung mit Robotern hat Tradition im Informatikunterricht der Sekundarstufe I. Das in den Lehrplänen vieler Bundesländer für die Sekundarstufe I empfohlene Thema 'Messen, Steuern, Regeln' wurde auch bisher schon oft mit Robotern gestaltet. Schwerpunkte der unterrichtlichen Auseinandersetzung mit dem Thema sollten hauptsächlich sein:

- Einsatz von Robotern und Automatisierung in der Produktion;
- die damit verbundenen gesellschaftlichen Auswirkungen;
- elementare Algorithmen zum Steuern und Regeln;
- grundlegende Prinzipien digitaler Informationsverarbeitung und -übermittlung;
- elementare Aspekte der Prozessdatenverarbeitung.

Schülerinnen und Schüler sollten sich diesen Themenstellungen altersadäquat aus der Perspektive von Anwendern und Nutzern in unterschiedlichsten Teilbereichen des gesellschaftlichen Lebens, aus der Perspektive der Betroffenen im Hinblick auf die persönlichen und gesellschaftlichen Folgen von Automatisierungsprozessen, aber auch in der Rolle von Gestaltern derartiger technischer Systeme nähern. Welche 'Technik' stand im Unterricht für derartige Konzepte zur Verfügung?

Neben verschiedenen 'didaktischen Modellen zur Steuerung' hatten diverse Hersteller (Fischertechnik, Lego u.a.) für diese Aufgabenstellungen Roboterbausätze entwickelt, die über ein Interface fest mit dem Computer verbunden waren. Mittels einer mehr oder weniger komfortablen Software-Schnittstelle oder per kryptischem Basic oder Assembler Code konnte der Roboter über das Interface angesprochen werden. Auch bewegliche Systeme, die allerdings ihre Steuerbefehle über ein Kabel von dem auf dem PC ablaufenden Programm erhielten, konnten konstruiert werden.

Mit Mindstorms ist nun einiges anders. Die Mindstorms Roboter sind nicht fest mit dem Interface verdrahtet, sondern fungieren als quasi autonome Systeme, die den Programmcode zu ihrer Steuerung als komplettes Paket vom Computer per Infrarotschnittstelle erhalten. Die Programmierung erfolgt über eine auf dem Computer installierte Software, die die elementaren Steuerbefehle des auf dem elektronischen Baustein des Mindstorms Roboter (RCX) befindlichen Betriebssystems (Firmware) aufruft. Nach erfolgter Übertragung kann das Programm vom RCX abgearbeitet werden, und der Roboter führt im idealen Falle die ihm zugeordneten Aktionen aus. Damit können im Unterricht die prinzipiellen Funktionen eines autonomen Systems erkundet und gestaltet werden. Aufgaben, wie 'einem Hindernis ausweichen', 'bewegen im Labyrinth' oder 'finden eines Zielobjekts' sind nun im Unterricht mit den Mindstorms lösbar. Sie fördern ein grundsätzliches Verständnis für ähnliche Aufgabenstellun-

gen, wie etwa in der Produktion oder in der Raumfahrt, wo lange Signallaufzeiten z.B. bei der Marsmission des 'pathfinder' die Unabhängigkeit von leistungsfähigen stationären Großrechnern und damit den Einsatz partiell autonomer 'eingebetteter Systeme' erfordern. Neben dem Problemkomplex 'eingebettete Systeme' sind mit den Mindstorms Robotern auch parallele Prozesse, realisiert mit Multitasking, und Probleme der Kommunikation und Kooperation zwischen autonomen eingebetteten Systemen im Unterricht prinzipiell behandelbar. Insgesamt kann man feststellen, dass sich die in den Informatiklehrplänen angesprochenen Themenbereiche und Zielsetzungen mit den Mindstorms Robotern sicherlich realisieren lassen und dass es darüber hinaus noch weitere Vertiefungsmöglichkeiten gibt.

### ***Motivationale und mediale Komponenten***

Die Mindstorms Roboter besitzen besonders für den Informatikunterricht in der Sek I aber auch für den Anfangsunterricht in der Sek II eine hohe motivationale und mediale Qualität. Sie genügen in ihrer medialen Funktion im Unterricht und hinsichtlich der mit ihnen erschließbaren Themenbereiche einerseits Kriterien wie Altersgemäßheit, Bezug zum Erfahrungshorizont der Schülerinnen und Schüler, Wissenschaftsorientierung oder Zukunftsbedeutung. Andererseits repräsentieren sie in den Erfahrungen vieler Schülerinnen und Schüler auch einen am Spiel orientierten außerschulischen Freizeitkontext, so dass der Informatikunterricht von den damit verbundenen motivationalen Effekten profitieren kann. Voraussetzung dafür ist, dass entdeckendes und selbstgesteuertes Lernen, wie es im außerschulischen Bereich im Umgang mit 'Spielzeug' üblich ist, im Informatikunterricht nicht durch belehrende und frontale Unterrichtsformen, die die Schülerinnen und Schüler zu weitgehender Passivität verurteilt, dominiert wird. Schülerinnen und Schülern sollte auch im Informatikunterricht, bei allem Bemühen, ihnen informatische Grundkonzepte zu vermitteln, der Spaß am problemlösenden Tüfteln und Ausprobieren sowie damit letztlich die Freude am spielerischen Umgang mit dem Material nicht genommen werden.

Die mediale Qualität der Mindstorms für den Informatikunterricht liegt u.a. auch darin begründet, dass Schüler eine unmittelbare visuelle Rückkopplung hinsichtlich ihrer theoretischen und praktischen Problemlösungsversuche erhalten. In den Mindstorms haben sich ihre problemlösenden Ideen materialisiert, und in dem Verhalten der Roboter können sie auch Misserfolg oder Erfolg und Effizienz ihrer Bemühungen ablesen.

### ***'Info-mecha-tronik'***

Am Beispiel der Mindstorms lässt sich auch im Unterricht verdeutlichen, dass Informatik mehr beinhaltet als das Codieren von Algorithmen in einer Programmiersprache. Ähnlich wie bei der Genese der Fachwissenschaft Informatik, wo Ingenieurwissenschaften, Mathematik, Elektro- und Kommunikationstechnik einen entscheidenden Einfluss hatten, kann bei den Mindstorms Robotern das Zusammenspiel verschiedener technischer und informatischer Konzepte beobachtet werden. Nur wenn alle Komponenten des Roboters aufeinander abgestimmt sind, kann er die ihm zugedachte Aufgabe zufriedenstellend lösen: Ausgehend von der Problemstellung bedarf es einer Modellierung, die schließlich auf der Softwareseite des Informatiksystems 'Roboter' zu problemadäquaten Datenstrukturen und Algorithmen führt. Deren Effizienz hängt jedoch entscheidend von der mechanischen Konstruktion des Roboters und dem wirkungsvollen Zusammenspiel von Sensoren und Aktoren ab. Software und die zugehörige Hardware erscheinen hier als wesentliche Elemente von Steuer- und Regelprozessen. Softwareentwicklung steht bei diesem eingebetteten System in entscheidender Wechselwirkung mit der gewählten Mechanik und repräsentiert auf anschauliche Weise ein wichtiges Kooperationsfeld von Informatik und Ingenieurwissenschaften. Hinzu kommen weitere Betrachtungsperspektiven, die am Beispiel der Mindstorms Roboter weitere informatische Problemstellungen und die fachübergreifende Kooperation verschiedener fachwissenschaftlicher Methoden erschließen.

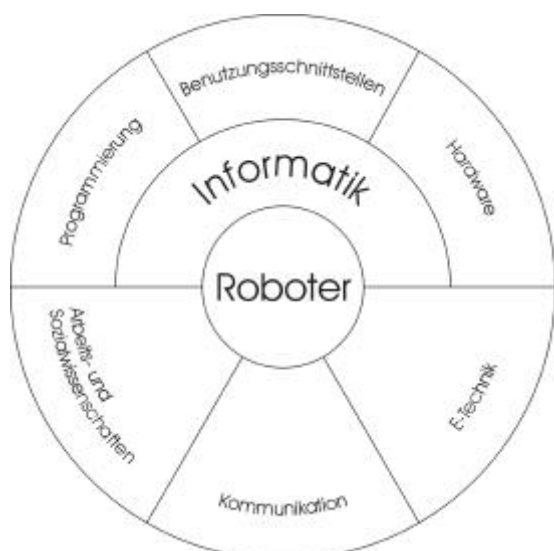


Abb. 2: Interdisziplinäre Sichtweisen auf Roboter

Die Codierung der Algorithmen in einer Programmiersprache und deren Test führt innerhalb der Informatik zu Fragen der Wahl einer geeigneten Programmiersprache nebst zugehöriger Entwicklungsumgebung und zum Problem der Gestaltung von Benutzungsschnittstellen. Die Übermittlung der Daten vom Computer auf den Softwarebaustein RCX eröffnet die Möglichkeit zur Erörterung von elementaren Hardwarestrukturen, von Grundprinzipien digitaler Kommunikation und fundamentalen elektronischen Konzepten zu ihrer Realisierung. In der optischen Gestaltung der Roboter können jenseits ihrer Funktionalität Fragen des Designs zu Problemen visueller Kommunikation führen. Ganz zu schweigen von dem breiten Feld der Anwendungen und Auswirkungen des Einsatzes von Robotern und eingebetteten Systemen in verschiedenen Bereichen gesellschaftlichen Lebens, die zu wesentlichen Problemstellungen von Arbeits- und Sozialwissenschaften führen. Das Medium Roboter bietet also reichhaltigstes Material für unterrichtliche Zugänge auch im Informatikunterricht und ist bestens geeignet, eindimensionale Blickwinkel auf die Informatik zu verhindern und die hohe Interdisziplinarität des Faches herauszuarbeiten. Natürlich meint das nicht, dass alle Perspektiven innerhalb einer Unterrichtsreihe zu thematisieren seien, sondern hier sollte, wie auch in anderen Fällen und Fächern, auf das bewährte Prinzip des Spiralcurriculums zurückgegriffen werden.

### **Modellieren im Informatikunterricht**

Schon im Informatikunterricht der Sek I sollte verengten Sichtweisen auf das Fach vorgebeugt werden. Hierzu gehört unserer Meinung nach die Vermittlung der Einsicht, dass für die Gestaltung von Informatiksystemen die vorhergehende Modellierung des Systems eine wesentliche Voraussetzung bildet. Unter Informatiksystemen verstehen wir hierbei eine Einheit von Soft- und Hardware, die sie umgebende Technik sowie ggf. das assoziierte soziale Handlungssystem. Modellierung findet ihre Grenzen bei der Umsetzung zur Systemgestaltung, wenn technische Disfunktionalitäten im Zusammenspiel mit übergeordneten Systemen nicht intendiertes Systemverhalten zeigen oder vor allem, wenn bei der Neuorganisation von assoziierten sozialen Handlungssystemen menschliche Verhaltensdispositionen das Gesamtsystem in nichtdeterministischer Weise beeinflussen.<sup>2</sup>

Die 'Modellierungstechniken', die im Anfangsunterricht in der Sek I zur Verfügung stehen, sind naturgemäß begrenzt, da es zunächst an den notwendigen fachwissenschaftlichen Vorkenntnissen fehlt. Dennoch scheinen uns vielversprechende Zugänge möglich, wenn mit Papiermodellen der Roboter, die die wesentlichen Funktionen von Motoren, Sensoren und Mechanik abbilden, in Abhängigkeit von der Aufgabenstellung das notwendige Zusammenspiel von Technik der Roboter, der technischen Gestaltung des Umfeldes und den daraus resultierenden Anforderungen an die Software mit Schülerinnen und Schülern mit Papier und Bleistift im Unterricht erarbeitet werden. Modellieren eines Informatiksystems wird durch diese Vorgehensweise als wichtige Aufgabenstellung der Informatik in die

<sup>2</sup> vgl. Magenheim 2000  
vgl. ebenso Ropohl 1999

Wahrnehmung der Schülerinnen und Schüler gerückt. Modellieren wird als Prozess wahrgenommen, der die wechselseitig verzahnten Bereiche von 'Technik des Roboters', 'technische Struktur des Einsatzumfeldes' und 'Software' zu beschreiben und gestalten versucht und der seine Fortsetzung sowohl in der Entwicklung von Algorithmen, deren Codierung und Implementierung, als auch in der Konstruktion der Mechanik finden muss.

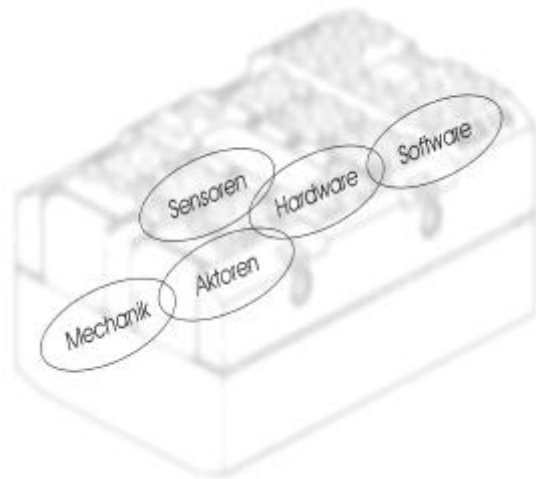


Abb. 3: Mindstorms Roboter als Eingebettete Systeme

Damit sind wesentliche Aspekte des Modellierens und der Gestaltung von Informatiksystemen angesprochen, die später, wenn es in einem anderen Kontext etwa um den Einsatz von Industrierobotern und den daraus resultierenden Folgen geht, um den wichtigen Bereich der assoziierten sozialen Handlungssysteme erweitert werden können. Jugendliche erfahren auf diese Weise vor allem auch, dass es bei technischen Gestaltungsprozessen oft auch Alternativen gibt. Schülerinnen und Schüler zu befähigen, technische und assoziierte soziale Strukturen nicht als unabwendbar hinzunehmen, sondern nach Alternativen zu fragen und sie im Hinblick auf Zielkonflikte kritisch zu bewerten, kann in diesem Zusammenhang ein wichtiger Beitrag des Informatikunterrichts zur Allgemeinbildung sein.

#### **Restriktionen beim Unterrichtseinsatz: Kleine Bauteile, große Lerngruppen**

So wünschenswert es wegen einer integrativen Betrachtungsweise von Technik und von eingebetteten Systemen auch sein mag, dass die Schülerinnen und Schüler die Montage der Mindstorms Roboter selbst vornehmen, um so klarer haben uns auch unsere unterrichtspraktischen Erfahrungen gezeigt, dass dies nur mit einem erheblichen Betreuungsaufwand und am Besten in Kleingruppen zu realisieren ist. Zudem ist ein zusätzlicher Zeitbedarf von mehreren Stunden bei der Unterrichtsplanung zu berücksichtigen. Arbeiten mehrere Gruppen parallel, kommt es auch leicht zur Vermischung der vielen kleinen Bauteile, die dann für einen nächsten Durchgang aufwendig wieder auseinander sortiert werden müssen. Es ergeben sich aber andererseits interessante Einblicke in technische Konstruktionsprinzipien (z.B. Kraftübertragung, Getriebe, Drehmomente, ...), die sonst im Unterricht nicht erfahrbar gemacht werden können und viele spannende Situationen zum selbstentdeckenden Lernen bieten. Was tun? Mögliche Auswege aus dem Dilemma: Montage in Kleingruppen im Rahmen einer AG am Nachmittag vornehmen oder auf vormontierte Robotertypen zurückgreifen, deren Funktion erkundet werden kann. Eine digitale Konstruktionsdatenbank, die die wesentlichen Konstruktionsprinzipien der Roboter visuell beschreibt und evtl. auch kurze Videosequenzen des Robotermodells in Aktion zeigt, könnte in diesem Zusammenhang eingesetzt werden<sup>3</sup>.

<sup>3</sup> Wir haben hierzu eine kleine Datenbank mit der Software MLCAD erstellt.



Abb. 4: Beispiele für Konstruktionszeichnungen

Außerdem kann bei vorgefertigten Roboter-Typen auch leichter eine Beziehung zu 'virtuellen Robotern' im Rahmen einer objektorientierten Vorgehensweise hergestellt werden (s.u.).

#### ***Vermittlung Informatischer Prinzipien***

Im informatischen Kernbereich gibt es neben der Einführung in die Idee der Modellierung eine Reihe weiterer wichtiger Zugänge zur Informatik, die, wie teilweise schon oben erläutert, mit den Mindstorms Robotern im Unterricht erarbeitet werden können: Einführung einfacher Kontrollstrukturen bei der Erarbeitung von Steuerungsalgorithmen; Erarbeitung elementarer Algorithmen; Gestaltung von Benutzungsoberflächen, etwa bei der Entwicklung von Software zur direkten Steuerung der Mindstorms; Codierung und Digitalisierung von Daten bei der Datenübertragung zwischen PC und RCX bzw. bei der Kooperation zweier Roboter; Skalierung von analogen Messwerten zur algorithmischen Weiterverarbeitung; Erkennen des Prinzips eingebetteter Systeme und der Wechselwirkung zwischen Software und Hardware sowie der zugehörigen Mechanik. Mit fortgeschrittenen Lerngruppen bieten sich auch vertiefende Betrachtungen an, etwa über die Effizienz der Steuerungsalgorithmen, die effiziente Nutzung von Speicherplatz des RCX-Bausteins, um umfangreichere Programme platzieren zu können; über das Konzept der Parallelverarbeitung, etwa bei der scheinbar simultanen Abfragen von Sensordaten und gleichzeitiger Steuerung der Motoren - realisiert mit Multitasking. Auch die Strategie zyklischer Vorgehensmodelle der Softwareentwicklung kann Schülerinnen und Schülern mit den Mindstorms nahe gebracht werden, z.B. mit der schnellen Entwicklung eines Roboter-Prototypen, der zunächst nur eine simple Lösungsstrategie - möglicherweise sogar nur für ein Teilproblem beherrscht. Der Prototyp wird durch stete Verfeinerung seiner Fähigkeiten, die durch zyklisch wiederkehrende Phasen praktischer Anwendung, deren Evaluation sowie daraus resultierender Veränderung der Algorithmen erfolgt, soweit verbessert, bis er schließlich die gestellte Aufgabe zufriedenstellend löst. Die Schüler evaluieren damit ihre eigene Arbeit und bemessen sie an der eingangs aufgestellten Anforderungsdefinition. Sie nehmen hierbei wechselnd die Rollen von Entwicklern und Auftraggebern ein und simulieren dergestalt den interaktiven kommunikativen Prozess, ohne den professionelle Softwareentwicklung nicht möglich ist. Auch hier bieten die Mindstorms Roboter wieder eine breite Palette von Sichten auf die Informatik und ihre Methoden, die die Vielschichtigkeit des Faches erschließen können.

#### ***Einübung objektorientierter Sichtweisen***

Natürlich kann es nicht Aufgabe des Informatikunterrichts in der Sek I sein, das Konzept objektorientierten Modellierens und Programmierens in all seinen Facetten zu vermitteln. Die fachwissenschaftlich begründete Anwendung derartiger Strategien, die mit Begriffen wie Wiederverwendbarkeit, Projektkomplexität und Softwareentwicklung im Team beschrieben werden, treffen hier nicht oder nur in geringem Maße zu. Trotzdem sehen auch Lehrpläne für den Informatikunterricht in der Sek I eine Einführung in fundamentale Konzepte der Objektorientierung vor. Neben lernpsychologischen Argumenten, die eine Identifizierung von Objekten in der Umwelt als günstige Voraussetzung für den Einstieg in objektorientierte Denkweisen ansehen oder die eine Förderung von vernetztem Denken im Zusammenhang mit der Objektorientierung<sup>4</sup> postulieren, können auch einige fachdidaktische Überlegungen die Auseinandersetzung mit der Thematik in der Sek I für angebracht erscheinen lassen. Zugänge zur Idee des Modellierens lassen sich durchaus sinnvoll mit objektorientierten Sichtweisen verbinden. Ein Papiermodell des Roboters mit wesentlichen Aktoren und Sensoren kann in Beziehung zu den Anforderungen des Umfelds gebracht und als erster Entwurf einer Klasse 'Roboter' mit assoziierten Klassen (Motor, Sensor) und zugehörigen Methoden (fahre\_vor, fahre\_zurueck, melde\_messwert, etc.) angesehen werden. Basierend auf dieser abstrakten und zugleich am Papiermodell veranschaulichten 'Klassendefinition' können dann konkrete Roboter mit spezifischen Aufgabenstellungen und z.T. unterschiedlichen Methoden abgeleitet werden. Mittels einer geeigneten Programmiersprache und einer zugehörigen Entwicklungsumgebung lässt sich dieses Konzept auch als Klasse (virtueller Roboter) auf dem Rechner implementieren. Das einmal entworfene Grundkonzept des Roboters lässt sich für unterschiedliche Aufgabenstellungen spezifizieren und damit wiederverwerten. Außerdem kann an dem einfachen Beispiel auch das Zusammenspiel verschiedener Objekte verfolgt werden. Schön wäre es in diesem Zusammenhang, wenn im Unterricht ohne große 'Umbaumaßnahmen' zu jedem virtuellen Roboter auf Softwarebasis auch ein reales Mindstorms Modell zur Verfügung stände, an dem man die augenscheinliche 'Korrektheit' des Softwareentwurfs zumindest empirisch überprüfen könnte. Mit einem solchen Konzept kann schon im Informatikunterricht der Sek I ein Grundverständnis für objektorientierte Sichtweisen in der Informatik gefördert und eine vertiefende Auseinandersetzung mit der Thematik in der Sek II vorbereitet werden. Es ist sicher auch für den Anfangsunterricht in der Sek II geeignet.

### **Programmierung, Sprachkonzepte und Entwicklungsumgebungen**

Da wir dieser Thematik im zweiten Teil des Artikels breiteren Raum schenken wollen, hier nur einige knappe Bemerkungen und ein vergleichender Überblick über die von uns untersuchten Sprachkonzepte und Entwicklungsumgebungen:

<b>Sprache</b>	<b>Textuell/visuell</b>	<b>Eigene Entwicklungsumgebung</b>	<b>Unterstützung von OO-Sichtweisen</b>	<b>Betriebssystem</b>
Robotics Invention System 1.5	visuell	ja	-	Win / Mac
Robo Lab 1.5	visuell	ja	-	Win / Mac
Not Quite C	textuell	nein <sup>1)</sup>	-	Win / Mac / Linux
Spirit.ocx	textuell	nein <sup>2)</sup>	o	Win
Visual Basic	textuell/ visuell	ja	o	Win
Delphi	textuell / visuell	ja	o	Win
Roboter Control Center	textuell / visuell	ja	o	Win
Lego Script	textuell	nein	o	Win
Java (TinyVM)	textuell / visuell	nein <sup>3)</sup>	+	Plattform unabhängig

<sup>4</sup> vgl. Schulte 2000



Small Talk	textuell	ja	++	Win
LegOS	textuell	nein	-	Win / Linux / Unix

- 1) Für Not Quite C wurden mittlerweile auch Entwicklungsumgebungen programmiert.
- 2) Die ActiveX-Schnittstelle kann unter Windows in grafische Entwicklungsumgebungen eingebunden werden.
- 3) Die Entwicklung kann in entsprechenden Java-Tools erfolgen.

Tab. 1: Programmierschnittstellen und Entwicklungsumgebungen für den RCX

Unstreitig ist, dass im Informatikunterricht der Sekundarstufe I, auch nach der Auffassung zahlreicher Lehrpläne der Bundesländer, die Programmierung von Computern einführend behandelt werden sollte. Mit den Mindstorms Robotern lassen sich Unterrichtskonzepte verwirklichen, die die Programmierung und Codierung des elektronischen Bausteins im Unterricht als Teil eines komplexeren Modellierungsprozesses verstehen. Wie die Tabelle zeigt, erlauben Softwareschnittstellen zum elektronischen Baustein RCX sowohl textuelle, als auch visuelle Programmierung. Teils werden sie mit eigener Entwicklungsumgebung angeboten, teils können sie in komplexere professionelle Umgebungen eingebunden werden. Es sind imperative und mehr oder weniger objektorientierte Sprachkonzepte für die Schnittstelle vorhanden.

Für den Einstieg mit Anfängern ist sicherlich die von Lego mit den Bausätzen ausgelieferte visuelle Entwicklungsumgebung des RIS gut geeignet, die allerdings keine objektorientierten Zugänge erlaubt. Sie kann Anknüpfungspunkte für einen sich abzeichnenden Trend zum Einsatz von visuellen objektorientierten Modellierungs- und Programmierungsumgebungen auf der Basis von UML (Unified Modeling Language) bieten, deren Verwendung im Informatikunterricht in der Oberstufe äußerst interessant erscheint.

Für die in diesem Artikel dargestellte Unterrichtseinheit mussten wir uns für eine Entwicklungsumgebung entscheiden. Das Unterrichtskonzept ist prinzipiell allerdings auch mit anderen Entwicklungsumgebungen realisierbar. Wir haben uns im Seminar für die Entwicklung einer eigenen Schnittstelle auf der Basis von Visual Basic entschieden, um Schülerinnen und Schülern die Möglichkeit zu geben, die im Zusammenhang mit der Mindstorms Programmierung erworbenen Kenntnisse auch für Problemstellungen in anderen informatischen Kontexten anwenden zu können. Außerdem erlaubt Visual Basic, Sensibilität für einfache Fragen der Gestaltung von Benutzungsschnittstellen zu erzeugen, und ermöglicht in begrenztem Umfang, wenngleich selbst nicht objektorientiert, die Einführung objektorientierter Sichtweisen.

Für VB spricht auch die relative Verbreitung der Sprache und die Tatsache, dass sie für den Informatikunterricht in der Sek II weniger brauchbar erscheint; somit im Informatikunterricht der Sek I kein übermäßiger Beitrag zur Heterogenität der Lerngruppen in der Jahrgangsstufe 11 geleistet wird. Die Softwareschnittstelle RCC wurde in Visual Basic und, auf einer unteren nicht wahrnehmbaren Schicht, auf der Basis von NQC entwickelt. Sie ermöglicht einen sprachlich sinnvoll nachvollziehbaren Programmaufbau, der mit der Syntax von Visual Basic konform geht und nicht auf kryptische Befehle der OCX-Schnittstelle angewiesen ist. Eine Dokumentation des Konzepts ist über den Server der Didaktik der Informatik an der Uni Paderborn erhältlich. Eine ausführliche und gut verständliche Softwaredokumentation ist auch für den Informatikunterricht unerlässlich, wenn man erreichen möchte, dass die Schüler lernen, im Umgang mit einer Entwicklungsumgebung selbständig umzugehen. Dies ist in der Vergangenheit im Informatikunterricht oft vernachlässigt worden.

### **Software 'Roboter Control Center'**

Das Roboter Control Center (RCC) ist eine an der Universität Paderborn entwickelte didaktische Software für den Einsatz im Informatik-Unterricht. Sie ist Bestandteil der geplanten Unterrichtseinheit zum Thema Lego Mindstorms. Das RCC entstand aus der Überlegung heraus, eine einfach konstruierte und modifizierbare Software zu entwickeln, mit der sich die Roboter möglichst intuitiv programmieren lassen.

Die Steuerung der Roboter sollte dabei auf verschiedenen Ebenen möglich sein. Zum einen sollte eine 'interaktive Steuerung' vorhanden sein, die es ermöglicht, den Zustand der Sensoren des Roboters direkt abzufragen und die Motoren direkt anzusteuern. Soll auf dieser Ebene ein Algorithmus ablaufen, so ist der Benutzer der Software zuständig für die Ausführung und die korrekte Abarbeitung

des Algorithmus. Der Schüler, der das Programm in diesem Modus benutzt, erhält eine direkte und enaktive Erfahrung des Algorithmus.

In einem weiteren Schritt ist es möglich, den Roboter direkt von dem Rechner steuern zu lassen, auf dem die RCC-Software läuft. Mit der dem RCC mitgelieferten Programmierschnittstelle kann ein Algorithmus entwickelt werden, der dann in die Software integriert werden kann. Dieser lässt sich daraufhin über die Oberfläche des RCC starten und wird vom Rechner ausgeführt. Jede Zustandsänderung wird dann automatisch über die Infrarotschnittstelle vom PC an den RCX-Baustein bzw. umgekehrt übertragen. Diese Art der Steuerung trat in unseren Überlegungen allerdings später ein wenig in den Hintergrund, da die Übertragung über die Infrarotschnittstelle leider ein wenig langsam ist, wodurch Zustandsänderung zum Teil erst nach ein bis zwei Sekunden bekannt werden.

Die dritte und wohl wichtigste Möglichkeit ist die der autarken Steuerung, auch 'Programmierte Steuerung' genannt. Hierbei kann wiederum ein Algorithmus mit der mitgelieferten Schnittstelle geschrieben werden, der daraufhin auf den RCX-Baustein übertragen werden kann. Das Programm kann danach direkt von dem Baustein ausgeführt werden, eine Verbindung zur Infrarotschnittstelle ist nicht mehr erforderlich.

Die technischen Schwierigkeiten, die bei der direkten Steuerung durch den Rechner auftreten, machen jedoch deutlich, dass es durchaus erforderlich sein kann, Roboter autark arbeiten zu lassen. Wird dies im Unterricht entsprechend problematisiert, so kann dies durchaus ein lohnenswerter Unterrichtsinhalt sein.

Die Programmierung bzw. Steuerung der Roboter ist in den drei Varianten konsistent. Das heißt beispielsweise, dass das direkte Abfragen der Sensorwerte über die Infrarotschnittstelle in der 'Interaktiven Steuerung' genauso verläuft wie das Abfragen der Werte innerhalb eines autarken Programms. Dadurch ist es möglich, durch Dekonstruktion der 'Interaktiven Schnittstelle' auf die Programmierung der Roboter zu schließen. Die Schüler können sich also selbst erarbeiten, wie die Programmierung der Roboter stattfindet.

### **Programmoberfläche**

Die drei zuvor erwähnten Grundfunktionen des Programms wurden in drei verschiedenen Benutzungsdialogen realisiert.

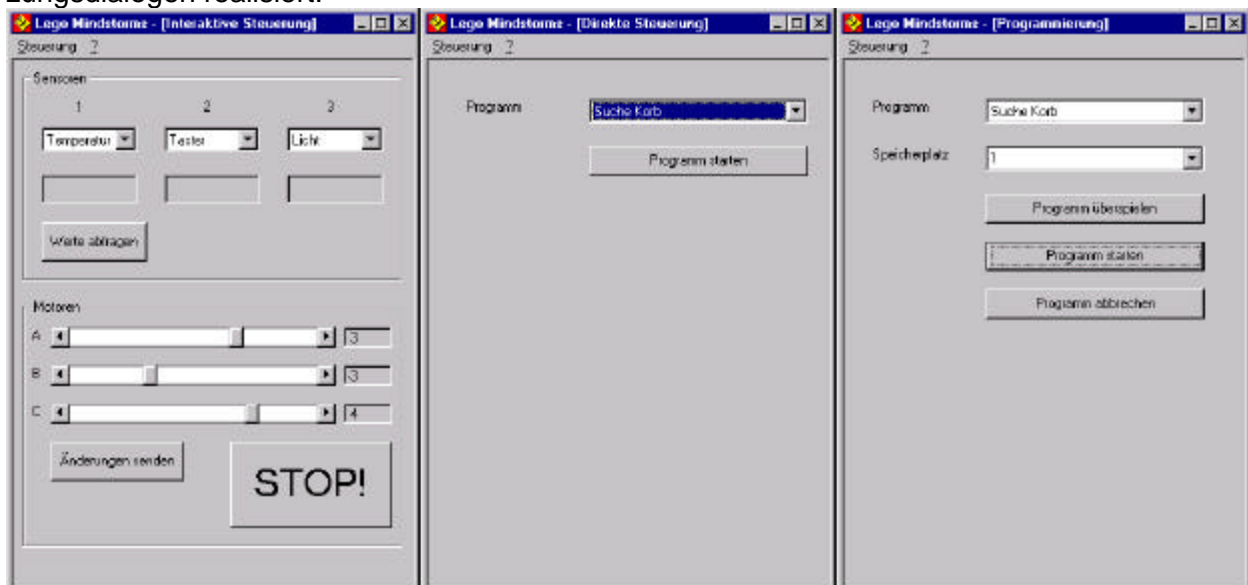


Abb. 5: Benutzungsoberflächen im RCC

Die Dialoge wurden auf das notwendigste reduziert, um den Schülern die Möglichkeit zu geben, die Abläufe, die zwischen der Benutzungsoberfläche und dem zugrundeliegenden Modell geschehen zu verstehen und eventuell zu modifizieren.

### **Interner Aufbau**

Das RCC wurde durchgehend objektorientiert realisiert, wobei allerdings an einigen Stellen aufgrund der eingeschränkten Möglichkeiten, die Visual Basic hierbei bietet, Abstriche gemacht werden muss-

ten. Außerdem wurde auch das Modell von der Oberfläche entkoppelt, so gibt es eigene Klassen für die Kommunikation mit dem Roboter und für die Oberfläche. Aus diesem Grund lassen sich auch Prinzipien wie 'MVC' thematisieren.

Zur Kommunikation mit dem Roboter dient die Klasse 'VirtuRob', die in einer eigenen ActiveX-Schnittstelle namens 'Virtueller\_Roboter' liegt. Die Klasse wurde in eine solche Schnittstelle ausgelagert, um die Einbindung in eigene Programme zu erleichtern. So ist es möglich, dass die Schüler selbst Software in Visual Basic, Delphi, JAVA oder anderen Programmiersprachen entwickeln und dabei auf die Klasse 'VirtuRob' der ActiveX-Schnittstelle zurückgreifen.

'VirtuRob' ist ebenfalls in Visual Basic entwickelt und kann daher selbst im Unterricht dekonstruiert und verändert werden. Die Klasse selbst nutzt zur Kommunikation mit dem Roboter die von Dave Baum entwickelte Programmiersprache NQC, die, wie der Name schon sagt, auf einer C-ähnlichen Syntax basiert. NQC wird ebenfalls in Form eines Open Source Konzepts vertrieben. Das heißt, auch NQC selbst kann, falls dies gewünscht wird, dekonstruiert werden. Allerdings kann dies aufgrund der Komplexität der Software nicht im Anfangsunterricht geschehen. Vielmehr könnte im Sinne eines Spiralcurriculums in einer späteren Phase, möglicherweise auch in der Oberstufe, das Konzept der Lego Mindstorms Roboter noch einmal aufgegriffen und durch Analyse von NQC vertieft werden.

Auch Lego liefert zusammen mit seinem Mindstorms Set eine ActiveX-Schnittstelle mit Namen 'PBrick' (Spirit.ocx) aus. Doch haben wir uns zur Entwicklung einer eigenen Schnittstelle entschieden, weil 'PBrick' für die Nutzung im Unterricht nur bedingt geeignet ist, da sie beispielsweise keine instanzierbare Klasse zur Verfügung stellt. Sie kann nur als Bestandteil einer grafischen Oberfläche benutzt werden, was aber nach dem Paradigma der Trennung von Modell und Oberfläche nicht wünschenswert sein kann. Außerdem liefert sie nur äußerst kompliziert zu nutzende Methoden zur Ansteuerung und Nutzung des Roboters. In 'VirtuRob' wurden diese Funktionen auf die Visual Basic Sprachsyntax angepasst und so umgesetzt, dass sie intuitiv nutzbar sind. Des Weiteren wurden die Namen der Methoden aus didaktischen Gründen eingedeutscht.

Ein Beispiel für eine Schleife:

Lego Syntax:  
`while 9,0,2,2,1`

VirtuRob-Syntax:  
`wahrend "sensorwert(1) = 1"`

Beide Anweisungen haben dieselbe Bedeutung. Während man im VirtuRob-Beispiel sofort erkennt, was der Zweck der Anweisung ist, so ist dies im linken Beispiel alles andere als intuitiv verständlich. Eine solche Syntax trägt damit natürlich bei Anfängern eher dazu bei, Lernbarrieren aufzubauen, was man mit einer leicht verständlichen Syntax verhindern kann. Noch deutlicher wird der Unterschied in der Syntax, wenn man komplexe Bedingungen angeben möchte. Die Lego Schnittstelle lässt es nicht zu, Bedingungen logisch zu verknüpfen, wodurch eine Anweisung wie

```
wahrend "sensorwert(1)=1 AND sensorwert(2)=(55-variablenwert(4))",
```

die 'VirtuRob' erlaubt, mit der Lego Syntax nur durch mehrere aufeinanderfolgende bedingte Anweisungen zu realisieren wäre, deren Ergebnisse in Variablen zwischengespeichert werden müssten. Aus eben diesen Gründen haben wir uns zur Implementation einer eigenen Schnittstelle entschieden. Die Oberfläche des RCC greift selbst auch ausschließlich mit Hilfe der Klasse 'VirtuRob' auf den Roboter zu. Dadurch wird die Programmierung in sich konsistent und ist damit für die Schüler leichter durchschaubar. Die Oberfläche selbst eignet sich auch sehr zur Dekonstruktion, da sich an ihr neben objektorientierten Aspekten auch ereignisgesteuerte Programmierung einführen lässt. So lösen Objekte der Klasse 'VirtuRob' beispielsweise Statusmeldungen in Form von Ereignissen aus, die von der Oberfläche abgefangen werden.

Zur Programmierung eigener Roboter sollten die Schüler selbst Klassen implementieren, die auf 'VirtuRob' zurückgreifen, wobei jede Klasse einem Roboter entsprechen sollte. Die Programme, die der Roboter abarbeiten soll, sollten in Methoden dieser Klassen realisiert sein. Wünschenswert wäre es hier gewesen, diese Klassen von 'VirtuRob' abzuleiten, nur ist dies leider in Visual Basic noch nicht möglich, daher bietet sich eine Nutzung in Form einer Aggregation an. Vererbung soll allerdings laut Microsoft in der nächsten Version von Visual Basic möglich sein. Als Beispiel für eine solche Klasse liegt dem RCC die Klasse 'MyRob' bei.

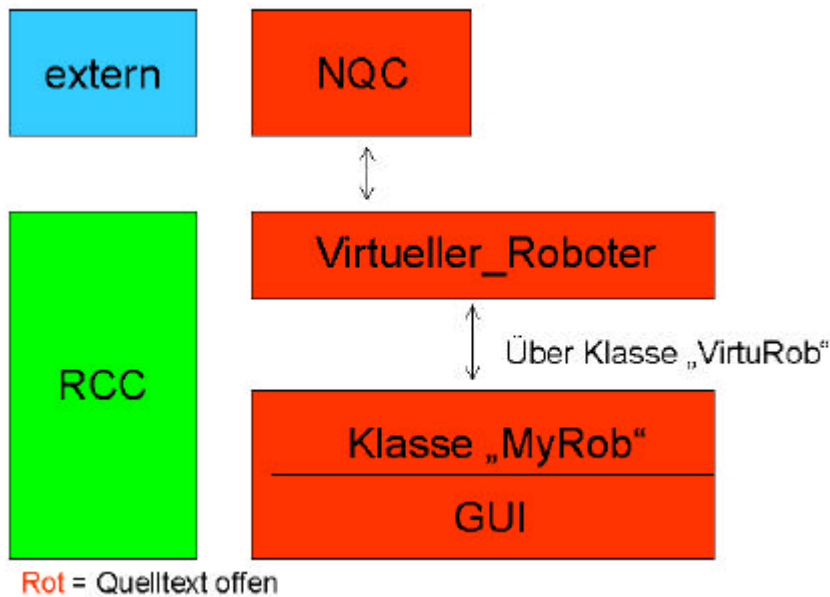


Abb. 6: Einbindung von NQC und Visual Basic in RCC

### Mindstorms in der Unterrichtspraxis

#### Skizze einer Unterrichtseinheit: 'Basketball'

Auf der Basis der vorhergehenden didaktisch-methodischen Überlegungen haben wir im ersten der beiden Seminare ein Unterrichtskonzept entwickelt, das später auch als Orientierungsgrundlage für eine unterrichtspraktische Erprobung diene (s.u.). Wir haben bewusst darauf verzichtet, die Planung präzise bis auf die Ebene von einzelnen Stunden vorzunehmen, da die konkrete Umsetzung des Konzepts von den jeweiligen Rahmenbedingungen vor Ort und insbesondere von den spezifischen Lernvoraussetzungen der einzelnen Gruppen abhängt. Statt dessen wird eine Abfolge von Unterrichtsphasen vorgestellt, die wir auf Grund von didaktisch-methodischen Überlegungen für sinnvoll erachten und die von den Lehrenden als Anregung aufgefasst und entsprechend ihren eigenen Intentionen umgesetzt werden kann.

#### Zielsetzungen

Als globale Zielsetzungen der Unterrichtseinheit können u.a. angesehen werden

- Einführung in elementare Techniken des Modellierens (und in objektorientierte Sichtweisen)
- Zugang zu Fragen der gesellschaftlichen Auswirkungen des Robotereinsatzes
- Ein Informatiksystem in der Variante als eingebettetes System kennenlernen und einen begrenzten Einblick in die interdisziplinäre Verflechtung des Faches gewinnen
- Erlernen des Umgangs mit einfachen (visuellen) Entwicklungsumgebungen
- Entwickeln von elementaren Steuerungsalgorithmen
- Zugang zu Fragen der Gestaltung von Benutzungsoberflächen
- Codierung von Algorithmen in einer einfachen Programmiersprache bzw. einer begrenzten Auswahl einer höheren Programmiersprache
- Mittels der Methoden von Konstruktion und Dekonstruktion von Software, Zugänge zu Problemen der Softwareentwicklung gewinnen

#### Mögliche Phasen einer Unterrichtseinheit mit den Mindstorms Robotern

Phasen	Inhalte / Ziele	Medieneinsatz
1	Einführung in das Thema Ausschnitt einer Filmsequenz zum Robotereinsatz oder Auswertung einer Erkundung Vorführen eines Mindstorms Roboters Erkunden der Mechanik (Motoren und Sensoren) Aufgabenstellung 'Basketball' Spontane Lösungsvorschläge zur Technik des geplanten Roboters Welche Kenntnisse müssen erworben werden?	Filmsequenz Mindstorms Roboter mit Sensoren (RCX)

2	Modellieren wesentlicher Aspekte des Gesamtsystems (Roboter, Umgebung) Festlegen wesentlicher Roboterfunktionen Festlegen der Anforderungen an das Umfeld des Roboters Diskussion der Lösungsvarianten	Papier, Filzstift, Kleber, Klebevorlagen
3	Einführung in die Entwicklungsumgebung (RCC, Visual Basic) Erkunden der Messwertrückmeldungen der Sensoren Erkunden der Motorfunktionen Erkunden der Handsteuerung eines einfachen Robotermodells	RCC Visual Basic Overhead o.ä. RCX
4	Analyse der interaktiven Steuerung im Hinblick auf die spätere Aufgabenlösung Analyse des Download Bereichs Fixieren von elementaren Steuerbefehlen Dekonstruktion der Benutzungsoberfläche	RCC Visual Basic Folien RCX
5	Option: Evtl. Bau eines problembezogenen Roboters mit Schülern Möglichst in Kleingruppen außerhalb des Unterrichts	Lego Bausätze
6	Entwickeln der elementaren Steueralgorithmen in Abhängigkeit von Technik Erlernen von einfachsten Kontrollstrukturen Option: Einüben objektorientierter Sichtweisen	RCC Folien
7	Arbeit mit dem RCC: Programmieren und Testen Erlernen von elementaren Kontrollstrukturen (RCC, Visual Basic) Implementieren der Steuerungsfunktionen Verbessern der Effizienz der Algorithmen	RCC Visual Basic RCX
8	Vorstellen der Ergebnisse Vergleich der unterschiedlichen Lösungen (Algorithmen, Technik, Umfeld) Effizienz der Lösung in Abhängigkeit von Algorithmus, Technik und Umfeld Ausführliche Filmsequenz: Robotereinsatz in der Industrie	Filmsequenz Folien Overhead o.ä.

Tab. 2: Phasen der Unterrichtseinheit 'Basketball'

*Phase 1* sollte der Einführung in das Thema dienen, wobei es uns sinnvoll erscheint, einen Bezug zu einem konkreten Einsatzbereich von Robotern herzustellen. Das kann mit einer kurzen Filmsequenz oder intensiver mit einer Betriebserkundung geschehen. Auf diese Weise werden die eher spielerisch orientierten Zugangsweisen zum Thema Roboter in den konkreten Kontext von gesellschaftlichen Fragestellungen eingebettet. Phase 1 sollte mit der Vorstellung eines Mindstorms Roboters und der Vorführung einer Problemlösung zu einer vorgegebenen Aufgabenstellung - z.B. Roboter folgt Lichtstrahl - abgeschlossen werden. Hierzu gehört auch ein erstes annäherndes Erkunden der Funktionsweise des vorgestellten Roboters als Zusammenwirken von Sensoren, Motoren, Mechanik und Einflüssen des Umfeldes. Abschluss von Phase 1 bildet die Einführung in die von den Schülerinnen und Schülern zu lösende Aufgabenstellung. In unserem Beispiel ist ein kleiner Ball, den der Roboter in einem Behältnis bei sich führt, in einen am Rande des Spielfeldes platzierten 'Korb' zu werfen. Es können erste spontane Lösungsvorschläge diskutiert und festgehalten werden. Außerdem können die Schülerinnen und Schüler feststellen, welche zusätzliche Kenntnisse und Fertigkeiten sie für die Problemlösung erwerben müssen.

*Phase 2* sollte nach Möglichkeit als Gruppenarbeit organisiert werden. Mittels der bereits oben erläuterten elementaren Modellierungsmethode sollte anhand eines Roboter-Prototypen aus Papier das Zusammenwirken von Mechanik, Sensorik und Motoren des Roboters im Grundsatz beschrieben und die technischen Anforderungen an sein Bewegungsumfeld definiert werden. Die Phase kann mit dem Vorstellen und der Diskussion der verschiedenen von den Schülerinnen und Schülern erarbeiteten Lösungsvarianten abschließen.



Abb. 7: Schüler beim Modellieren

*Phase 3* dient dem Erkunden einiger wesentlicher Funktionen der Entwicklungsumgebung, in unserem Fall von RCC und Visual Basic. Es können anhand des in Phase 1 vorgestellten Roboters und mit einer geeigneten Benutzungsoberfläche Messwertmeldungen der Sensoren ausprobiert, die Reaktionen und Funktionen des RCX-Bausteins erforscht und die Steuerung der Motoren getestet werden. Mit Hilfe einer simplen Benutzungsoberfläche kann ein Robotermodell mit geringem Funktionsumfang (z.B. Motor: vor und zurück; Lichtsensor: hell - dunkel) im Direktmodus per Hand gesteuert werden.

In der sich anschließenden *Phase 4* kann eine Einführung in die Methode der Dekonstruktion von Software erfolgen, indem die Befehlskette von Buttons der Benutzungsoberfläche bis hin zum Aufruf von Kommandos auf Methoden- bzw. Prozedurebene verfolgt wird. Die dem Konzept ereignisorientierter Programmierung zugrundeliegende fundamentale Idee kann dabei ebenso thematisiert werden, wie die Auswahl und die Anordnung der Buttons als Element des Softwaredesigns zur Robotersteuerung. Zu diskutieren sind auch Fragen nach den möglichen Intentionen des Designs und ggf. nach sinnvollen Modellierungsalternativen. Im Prozess der Dekonstruktion werden auch erste Elemente der Sprachsyntax von RCC und Visual Basic und elementare Kontrollstrukturen zur Implementierung von Algorithmen erlernt.

Wenn es die unterrichtlichen Rahmenbedingungen zulassen, können anschließend in einer optionalen *Phase 5*, möglichst in Kleingruppen, die Roboter nach den in Phase 2 erarbeiteten Anforderungsdefinitionen von den Schülerinnen und Schülern selbst montiert werden.

Danach können in längeren *Phasen 6 und 7* die elementaren Steueralgorithmen entwickelt, in RCC (Visual Basic) codiert und erprobt und ggf. Steueralgorithmen, Mechanik oder die Technik des Umfeldes verbessert werden. In diesem Zusammenhang sind auch elementare Kontrollstrukturen von VB bzw. der verwendeten Sprache zu erlernen. Objektorientierte Sichtweisen bei der Verwendung des Konzepts eines virtuellen Roboters und von einer Roboterklasse abgeleiteter und instanzierter Roboterobjekte können, wenn gewünscht, optional wie eingangs beschrieben in die Unterrichtseinheit eingebunden werden.

Diese Abfolge von Tätigkeiten wiederholt sich so oft, bis für die Gruppe ein befriedigender Zustand ihres Robotersystems erreicht ist. Auf diese Weise gewinnen die Schülerinnen und Schüler nicht nur Einsicht in die Wirkungsweise eines eingebetteten Informatiksystems und die Notwendigkeit der Abstimmung von Hard- und Software, Mechanik und technischem Umfeld des Systems, sondern lernen den Prozess der Softwareentwicklung als zyklisch verlaufenden Prozess kennen, der durch stetige Verbesserung den Wünschen der Auftraggeber und den Anforderungen der Aufgabenstellung angepasst werden muss. Dies ist ein Prozess, der im Prinzip nicht endgültig abzuschließen ist. Sie nehmen dabei im unterrichtlichen Handeln sowohl die Rollen von Entwicklern, als auch von Auftraggebern ein.

Abschluss der Unterrichtseinheit bildet *Phase 8*. Hier werden die in den Gruppen entwickelten Roboter vorgestellt und eine Demonstration ihrer Fähigkeiten gegeben. Die unterschiedlichen Lösungskonzepte und deren Effizienz können vergleichend im Plenum beurteilt werden. Ein erneuter Bezug zu

realen Robotersystemen (Videosequenz) kann sowohl prinzipielle Übereinstimmungen als auch gravierende Unterschiede in der konkreten technischen Realisierung und in der Effizienz der professionellen Systeme verdeutlichen. Im Unterricht am Beispiel der Mindstorms entwickelte Fähigkeiten wie das Finden eines Ziels, das Aufnehmen, das Transportieren und das Ablegen eines Gegenstandes sind auch Funktionen, die oft von professionellen Systemen ausgeübt werden. Kommt dann noch ins Blickfeld, dass im professionellen Bereich das Umfeld der Roboter nicht nur durch weitere Technik, sondern auch noch in wesentlichem Umfang von Menschen bestimmt wird, Modellieren von Informatiksystemen damit letztlich auch die Veränderung von Arbeitsorganisation beinhaltet, dann sind wesentliche Zielsetzungen der Unterrichtseinheit erreicht.

### Praktische Erfahrungen mit Schülerinnen und Schülern

Im Sommersemester 2000 war eine altersheterogene Gruppe von 16 Schülerinnen und Schülern der Sek I zu zwei Kompaktveranstaltungen von je 4 Zeitstunden Gast der AG Didaktik der Informatik an der Universität Paderborn. Im ersten Teil wurden die oben geschilderten Modellierungskonzepte erprobt und Roboter gebaut, die einen Korb auf dem ebenfalls zu gestaltenden Spielfeld finden und in einen Korb werfen sollten. Im zweiten Teil wurde eine Einführung in die RIS Entwicklungsumgebung gegeben (RIS 1.5), Algorithmen zur Problemlösung entwickelt, auf den RCX übertragen und die Problemlösungsstrategie sowohl hinsichtlich der Algorithmik als auch in Bezug auf Mechanik und Anordnung der Sensoren durch Erproben schrittweise verfeinert. Wir haben uns in dem zeitlich knapp bemessenen Seminar für die visuelle Entwicklungsumgebung des RIS entschieden, um ohne großen Erklärungsaufwand zur Sprachsyntax und zum Umgang mit der Entwicklungsumgebung schnell zu praktischen Ergebnissen zu kommen. Dies ist sicher ein wichtiger Vorteil der visuellen Entwicklungsumgebung RIS. Objektorientierte Sichtweisen konnten hierbei allerdings nicht entwickelt werden.

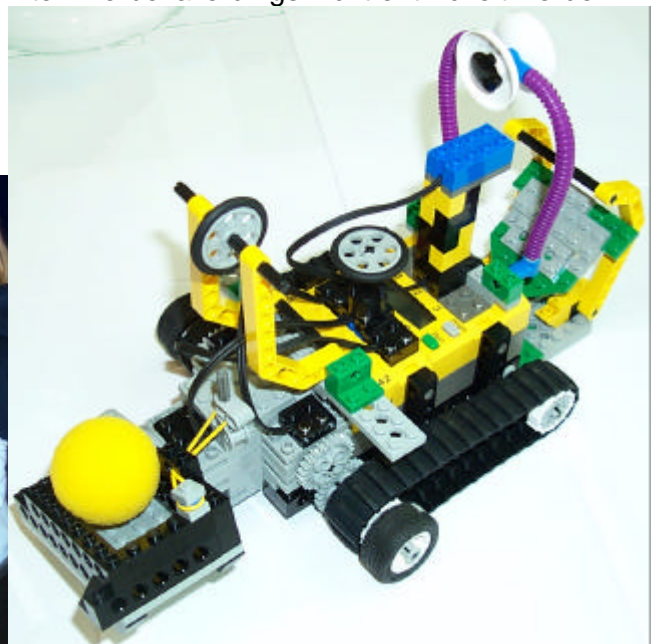


Abb. 8: Schüler beim Bau ihrer Mindstorms Roboter; Abb. 9: Ein von Schülern gebauter Roboter

Die Schülergruppe war mit großer Begeisterung am Werk. Insbesondere der Umstand, dass die Modelle in Kleingruppen selbst gebaut werden konnten<sup>5</sup>, hatte eine enorme motivationale Wirkung. Besonders zu vermerken ist der Umstand, dass eine Kleingruppe von Mädchen sich mit großer Begeisterung beteiligte und hinsichtlich des Designs ihrer 'Roboterin', die auch mit einem Namen versehen wurde, beeindruckende Maßstäbe setzte. Ein Indiz dafür, dass bei geeigneten Rahmenbedingungen auch Schülerinnen motiviert sind, sich im Unterricht mit informatischen und technischen Problemstellungen auseinanderzusetzen. Die altershomogenen Kleingruppen umfassten die Jahrgangsstufen 5-10. Zwischen den Gruppen waren hinsichtlich der Mechanik der Roboter nur geringfügige Unterschiede in Stabilität und Leistungsfähigkeit zu erkennen. Stärkere Differenzen zeigten sich bei der

<sup>5</sup> Wir hatten allerdings je eine(n) Betreuer(in) pro Kleingruppe mit 4 Schüler/innen eingesetzt.

Effizienz der implementierten Algorithmen, wobei hier die älteren Schülergruppen effektivere Lösungen erarbeiteten. Sicher nicht zuletzt auch ein Ergebnis der unterschiedlichen Vorerfahrungen. Wenngleich man die in diesen Veranstaltungen gesammelten Erfahrungen nicht uneingeschränkt auf die schulische Regelpraxis übertragen kann, so haben die beiden Kompaktseminare uns insgesamt doch in der Auffassung bestärkt, dass mit den Mindstorms Robotern ein durchaus lustvoller und spielorientierter Zugang zur Informatik in der Sekundarstufe I möglich ist, der zudem einen Blick auf mehrere der vielen Themenbereiche des Faches und seine interdisziplinären Verflechtungen ermöglicht und einem eindimensionalen syntaxorientierten Unterricht vorbeugt.

### **Literatur**

- Baum, D.: Dave Baum's Definitive Guide To Lego MINDSTORMS. New York 2000 (apress)  
Knudsen, J. B.: The Unofficial Guide To Lego MINDSTORMS Robots. Beijing, Cambridge, Farnham, u.a. 1999 (O'Reilly)  
Magenheim, J.: Informatiksystem und Dekonstruktion als didaktische Kategorien, Tagungsbeitrag zur GI-Tagung 'Informatik, Ausbildung und Beruf 2000'. Würzburg 2000  
Ropohl, G.: Allgemeine Technologie, Eine Systemtheorie der Technik, München. Wien 1999 (Hanser)  
Schulte, C.: Vom Modellieren zum Gestalten, Tagungsbeitrag zur GI-Tagung 'Informatik, Ausbildung und Beruf 2000'. Würzburg 2000  
Schult, T. J.: Präsent mit Potential. Roboterbaukästen mit PC-Anbindung. In: ct-magazin für computer technik. Nr. 25. Hannover 1998  
Feibel, T.: LEGOlise IT. Der Terminator im Lego-Kasten. In: ct-magazin für computer und technik. Nr. 4. Hannover 1998

### **Web-Links zum Thema**

- <http://www.legomindstorms.com/>  
<http://www.lego.de/dacta/robofab/default.htm>  
<http://lcs.www.media.mit.edu/people/fredm/mindstorms/>  
<http://www.enteract.com/~dbaum/lego/>  
<http://www.noga.de/legos/>  
<http://www.crynwr.com/lego-robotics/>  
<http://www.crazyrobots.com/tinyvmsite/>  
<http://www.object-arts.com/bower/bot-kit/>  
<http://ddi.uni-paderborn.de/didaktik/forschung/lego/>  
<http://www.lego.com/dacta/home.asp>  
<http://www.crynwr.com/lego-robotics/>  
<http://lcs.www.media.mit.edu/people/fredm/mindstorms/>

### **Autoren**

Johannes Magenheim  
Thorsten Reinsch  
Michael Hirsch