

Handlungsorientiertes Erlernen von Programmkonstruktionen anhand von Rollenspielen

Stefan Dißmann¹

Universität Dortmund
Lehrstuhl Software-Technologie
44221 Dortmund
stefan.dissmann@udo.edu

Abstract: Im Bereich der Programmierung werden komplexe Abläufe als sequentiell angeordnete Programmtexte formuliert. Viele Lernende sind nicht in der Lage, diese Texte unmittelbar als sich dynamisch ändernde Systeme zu interpretieren und scheitern dann bei dem Versuch, komplexe Algorithmen oder Datenstrukturen anhand ihrer Programmtexte zu verstehen. Dieser Beitrag zeigt mit dem Einsatz von Rollenspielen einen Weg auf, wie Algorithmen und Abläufe auch ohne technische Beschreibung verdeutlicht werden können. Die dabei gewonnenen Erfahrungen können anschließend eingesetzt werden, um die formale technische Beschreibung in einer Programmiersprache zu erarbeiten. Darüber hinaus bietet das erlebte Rollenspiel jedem Mitspieler einen Ansatzpunkt für das Bilden individueller Veranschaulichungen und Analogien.

1 Ausgangsproblematik

Das Ausführen und Überprüfen von selbst erstellten Programmen erfolgt schnell und unkompliziert mit eben dem Rechner, der auch zum Erstellen und Übersetzen Verwendung findet. Lernende können daher die Tätigkeit des Programmierens selbstständig vollziehen und die von ihnen erstellten Ergebnisse unmittelbar selbstständig kontrollieren. Die Programmierausbildung ist somit prädestiniert für einen stark handlungsorientierten Unterricht, in dem die Lernenden die Wirkung der programmiersprachlichen Konstrukte und deren Verwendung in der Programmkonstruktion am selbst erstellten Beispiel praktisch erproben. Leider zeigt sich in der Praxis der Ausbildung, dass eine Vielzahl von Lernenden trotz dieser günstigen Voraussetzungen an dem Erlernen des selbstständigen Konstruierens einfacher Programme scheitern. Auch motivierte

¹ Der Autor ist Mitarbeiter am Fachbereich Informatik der Universität Dortmund und hat in den vergangenen Semestern u.a. Lehrveranstaltungen für Studierende anderer Studiengänge betreut. Diese Studierenden hatten überwiegend weder Informatikunterricht in der Schule besucht noch im privaten Bereich Programmiererfahrungen gesammelt. Ihre Vorbildung und ihr Leistungsvermögen dürften somit mit dem von Schülern vergleichbar sein, die im Schulunterricht erstmals mit Programmieren konfrontiert werden. Der Autor geht davon aus, dass die von ihm entwickelten Lösungsansätze daher auch für den Schulunterricht geeignet sind.

Lernende sind nicht in der Lage, für eine gegebene Aufgabenstellung mit den Konstrukten einer Programmiersprache geeignete Abläufe und Datenstrukturen zu formulieren. Zielgerichtetes Handeln ist für diese Lernenden trotz günstiger Randbedingungen nicht möglich, da die zum geeigneten Handeln notwendige Kompetenz im Umgang mit den programmtechnischen Konzepten fehlt. Das Handeln am Rechner bleibt für diese Lernenden ein planloses Versuchen, dessen Ergebnisse weniger durch konzipierende Entscheidungen als vielmehr durch ein inkompetentes Reagieren auf die Fehlermeldungen des Compilers bestimmt sind. Der handlungsorientierte Unterrichtsansatz scheitert, da die Lernenden in ihrem Handeln nicht die notwendige Qualität erreichen. Obwohl die Lernenden über nahezu optimale Bedingungen zum eigenen Handeln und Experimentieren verfügt, scheitert handlungsorientierter Unterricht schon bei dem Versuch, erste Handlungskompetenz zu vermitteln.

Der Grund für dieses Scheitern liegt nach den Erfahrungen des Autors zumeist in der Art, wie die Tätigkeiten des Programmierwurfs und seine Operationalisierung durchgeführt werden. Komplexe Abläufe müssen erkannt, strukturiert und in eine textuelle, sequentiell geordnete Beschreibung in einer Programmiersprache übertragen werden. Fehlt den Lernenden die Fähigkeit, intuitiv eine eigene Vorstellung von programmiersprachlich formulierten Abläufen zu entwickeln, so gibt es wenig Möglichkeiten, Beispielprogramme zu verstehen und – darauf aufbauend – sinnvoll eigene Programme zu gestalten. Das Betrachten ausgeführter Programme hilft nicht weiter, da sich während der Ausführung die Wirkungsweise der Programmkonstrukte typischerweise nicht zeigt. Gerade dann, wenn nach einer ersten Lernphase die Komplexität von Aufgabenstellungen und damit auch der zugehörigen programmierten Lösungen zunimmt, muss der Lernende in der Lage sein, komplexe Folgen von Zustandsänderungen innerhalb des Programmablaufes zu überblicken und durch gezielte Konzeption zu erzwingen.

Kann sich ein Lernender – ausgehend vom Beispielprogramm – keine eigene Vorstellung von der Wirkungsweise einer Programmkonstruktion erarbeiten, so muss ihm der Lehrer dabei Hilfestellungen geben. Hierzu werden meist Graphiken eingesetzt, die zwar Zustände aber nur eingeschränkt dynamische Änderungen zeigen können. Für ausgewählte Probleme, etwa den Ablauf rekursiver Algorithmen, sind auch rechnergestützte Animationen verfügbar. Allerdings bleibt der Lernende bei diesen Animationen Betrachter und wird nicht in den Lernprozess einbezogen. In diesem Beitrag wird daher eine dritte Form der Veranschaulichung, das Rollenspiel, favorisiert. Rollenspiele, welche die Lernenden unmittelbar einbeziehen, stellen eine stark handlungsorientierte und insbesondere leicht den Verständnisproblemen des Lernenden anpassbare Form des Lernens dar.

2 Animation durch Rollenspiele

In der Zeit von Multimedia und e-Learning ist der lehrende Informatiker leicht versucht, die Realisierung einer Animation der Abläufe ausgeführter Programmkonstruktionen durch eine visualisierende Software vorzunehmen. Software-Animationen von Softwaresystemen sind jedoch lediglich künstliche Veranschaulichungen von Programmstrukturen und -verhalten, die beim Lernenden nicht zwangsläufig zu Assoziationen mit be-

kannten Erfahrungen führen. Für ihren erfolgreichen Einsatz muss eine Software-Animation daher sehr sorgfältig geplant werden, was zu einem recht aufwändigen Erstellungsprozess führt. Dem Lernenden bleibt trotzdem nur die Rolle des Betrachters des Animationsablaufes, da ihm für steuernde Eingriffe das Verständnis der system-internen Abläufe fehlt. Wird die Ausbildung in einer Gruppe durchgeführt, so kann eine solche Software-Animation durch ihren weitgehend vorgegebenen Ablauf die Kommunikation innerhalb des Lernprozesses nur wenig anregen.

Als alternative Form der Veranschaulichung dynamischer Vorgänge bietet sich dagegen das Rollenspiel an, das von Menschen animierte Systemverhalten als Theaterspiel. Menschliche Mitspieler sind leicht steuerbare, flexibel einsetzbare Komponenten eines Systems. Strikte Regeln leiten Menschen zu einem determinierten Handeln an, ein weniger strenges Drehbuch erlaubt aber auch autonomes und nicht deterministisches Handeln. Durch geschicktes Ausnutzen dieser menschlichen Fähigkeiten können Rollenspiele für die Darstellung eines großen Spektrums von Algorithmen und Datenstrukturen eingesetzt werden.

Wird also bei der Veranschaulichung der internen Abläufe eines Programms auf menschliche Akteure zurückgegriffen, so kann mit vergleichsweise wenig Aufwand eine ansprechende Animation geschaffen werden. Gelingt es, das Rollenspiel so einfach zu gestalten, dass keine Proben notwendig sind und kann daher auf beliebige Akteure zurückgegriffen werden, so können die Lernenden unmittelbar mit im Spiel agieren und werden dann sowohl zu Zuhörern und Betrachtern (der anderen Mitspieler) als auch selbst zu einem integralen Bestandteil des animierten Programms. Aus der Präsentation von Algorithmen und Datenstrukturen wird ein handlungsorientiertes Lernprojekt [Gu01, Wö98]. Handlungsorientierte Gruppenarbeit wie ein Rollenspiel mit verschiedenen Mitspielern initiiert zugleich zwangsläufig gruppenspezifische Lernprozesse und kooperatives Lernen.

Das »Durchspielen« des internen Ablaufs eines Produktes ist keine neue Erfindung, sondern zählt seit vielen Jahren zu den Standardaufgaben von Softwareentwicklern und ist in nahezu allen strukturierten oder objektorientierten Vorgehensmodellen implizit oder explizit vorgesehen. Ein Beispiel hierfür ist der zu den klassischen Techniken der Softwareentwicklung zählende »walkthrough« im Rahmen der Qualitätssicherung. Im Umfeld der objektorientierten Entwicklung wird insbesondere beim Einsatz von CRC-Karten während der Anforderungsanalyse die Simulation des durch die Karten spezifizierten Systemmodells als wesentlicher Entwicklungsschritt [Wi95, Am95, Wi90] hervorgehoben.

Rollenspiele sind also eine etablierte Technik, die im Entwicklungsprozess zur Evaluation der entwickelten Lösungen eingesetzt wird. Mitspieler im Rollenspiel sind dabei erfahrene Entwickler, die die technischen Konstruktionsprinzipien beherrschen und durch das gemeinsame Spiel deren Einhaltung am erstellten Produkt überprüfen sollen. Der vorliegende Artikel propagiert demgegenüber das Rollenspiel als Hilfsmittel in einer frühen Phase der Ausbildung, um dem noch unerfahrenen Lernenden die grundsätzliche Wirkungsweise von Programmkonstruktionen nahe zu bringen und ihm den Erwerb von Basiserfahrungen bereits ohne ausreichende Programmierkenntnisse zu ermöglichen. Als

Nebeneffekt bereitet diese Art der Ausbildung die Lernenden natürlicherweise auch auf den traditionellen Einsatzbereich von Rollenspielen in Anforderungsanalyse und Qualitätssicherung vor.

3 Rahmenbedingungen für den Einsatz von Rollenspielen

Das oben entworfene Ausbildungsszenario geht davon aus, dass die betroffenen Lernenden ohne besondere Vorkenntnisse in der Lage sein sollen, als Akteure eines Rollenspiels einen Algorithmus oder eine Datenstruktur gemeinsam so zu animieren, dass sie aus ihrem eigenen Handeln neues Wissen und neue Erkenntnisse ableiten können. Insbesondere sollen sie die grundlegenden Konstruktionsprinzipien erkennen können, ohne bereits auf Wissen in dem animierten Bereich zurückgreifen zu können.

Jeder durch ein Rollenspiel animierte Ablauf muss daher verschiedene, sehr unterschiedliche und sich zum Teil scheinbar konträr gegenüberstehende Rahmenbedingungen erfüllen. Es muss eine Problemlösung »gespielt« werden,

- die eine leicht verständliche Aufgabe erfüllt, um zusätzliche Schwierigkeiten der Lernenden beim Erfassen des Problembereichs zu vermeiden,
- deren Struktur und Arbeitsweise ohne die Verwendung der technischen Begriffe des animierten Bereichs beschrieben werden kann, um die Handlungsanweisungen formulieren zu können,
- deren Struktur und Arbeitsweise aber trotzdem von den Lernenden bis in die Details erkannt werden kann, um die Bedeutung einzelner Aspekte untersuchen zu können,
- deren Struktur und Arbeitsweise gegebenenfalls einfach gezielten Modifikationen unterzogen werden kann, um daraus Rückschlüsse auf die Bedeutung und Wirksamkeit der Gestaltungsprinzipien ableiten zu können und
- deren Struktur und Arbeitsweise leicht spontan geändert werden kann, um Anregungen aus der Diskussion der Lernenden schnell in das Rollenspiel umsetzen zu können.

Wesentliches Hilfsmittel für die Umsetzung dieser Rahmenbedingungen ist der gezielte Einsatz der natürlichen Fähigkeiten und der Intelligenz der Lernenden.

- Die Lebenserfahrung der Lernenden ermöglicht es, eine allgemein bekannte Aufgabenstellung für das Beispielsystem zu wählen.
- Die Fähigkeit zum eigenständigen Schlussfolgern ermöglicht es den Lernenden, auch unscharfe Handlungsanweisungen aufgrund des Kontextes der Aufgabenstellung zu vervollständigen und selbstständig danach zu handeln.
- Die Fähigkeit zur Selbstkontrolle ermöglicht es den Lernenden, sein Handeln zusätzlichen Anweisungen des Lehrers anzupassen und damit spontan zu ändern.

4 Beispiele von Rollenspielen für die Programmierausbildung

Nachdem in den vorangehenden Abschnitten die Idee des Rollenspiels in der Programmierausbildung motiviert und einige Randbedingungen für die Ausgestaltung von Rollenspielen postuliert worden sind, sollen jetzt drei Beispiele den Einsatz in verschiedenen Bereichen der Programmierung veranschaulichen. Als Beispiele für Rollenspiele werden hier vorgestellt:

- Die Einführung in rekursive Algorithmen am Beispiel der Berechnung der Quersumme einer natürlichen Zahl.
- Die Einführung in die Arbeitsweise von Operationen auf rekursiven Datenstrukturen wie linearen Listen und binären Bäumen.
- Die Einführung in die Grundprinzipien objektorientierter Systemkonstruktion.

Die Auswahl dieser Beispiele soll einerseits die Breite des Einsatzbereichs von Rollenspielen in der Informatik und andererseits zugleich die Möglichkeiten demonstrieren, die sich im Spektrum zwischen freiem Spiel und strengem Drehbuch ergeben.

4.1 Rekursion als Rollenspiel - Berechnung der Quersumme

Die Bestimmung der Quersumme einer natürlichen Zahl ist nach Auffassung des Autors ein recht nettes Beispiel für eine rekursiv vorzunehmende Berechnung. Wie bei vielen einfachen rekursiven Algorithmen wäre auch eine iterative Formulierung einfach zu erstellen. Der im Rollenspiel verwendete Algorithmus kombiniert zwei Operationen, die Restbestimmung (*»modulo«*) für die Ermittlung der letzten Ziffer und die ganzzahlige Division durch 10 zur Ermittlung einer aus den restlichen Ziffern bestehenden Zahl. Beide Operationen werfen keine mathematischen Probleme auf, so dass sich die Lernenden beim Rollenspiel völlig auf die Ausführung des rekursiven Arbeitsprinzips konzentrieren können.

Jeder Mitspieler, d.h. jeder Lernende, übernimmt bei der Berechnung der Quersumme die Rolle einer Instanz der rekursiven Berechnungsfunktion. Die Berechnung der Quersumme ist daher ein Rollenspiel mit einem fest vorgegebenen Drehbuch, das den Lernenden keine Freiheiten lässt. Diese Einordnung ist offensichtlich, muss doch bei der Rekursion sichergestellt werden, dass jede Instanz exakt so handelt wie seine Vorgänger und Nachfolger.

Das Drehbuch der Rekursion besteht aus einem einzigen Anweisungsblatt, das die Arbeit einer einzelnen Instanz textuell und graphisch beschreibt. Dieses Blatt wird von beiden Seiten genutzt, wobei die Vorderseite für den *aufrufenden* Spieler bereitsteht und die Rückseite den Algorithmus für den *ausführenden* Spieler enthält. Durch geeignetes Umknicken des oberen bzw. unteren Randes gelingt es, im Spiel das (interne) Handeln auf der Rückseite des Anweisungsblatts vom Aufruf auf der Vorderseite zu trennen. Die Abbildung 1 zeigt die Vorder- und Rückseite des Anweisungsblatts.

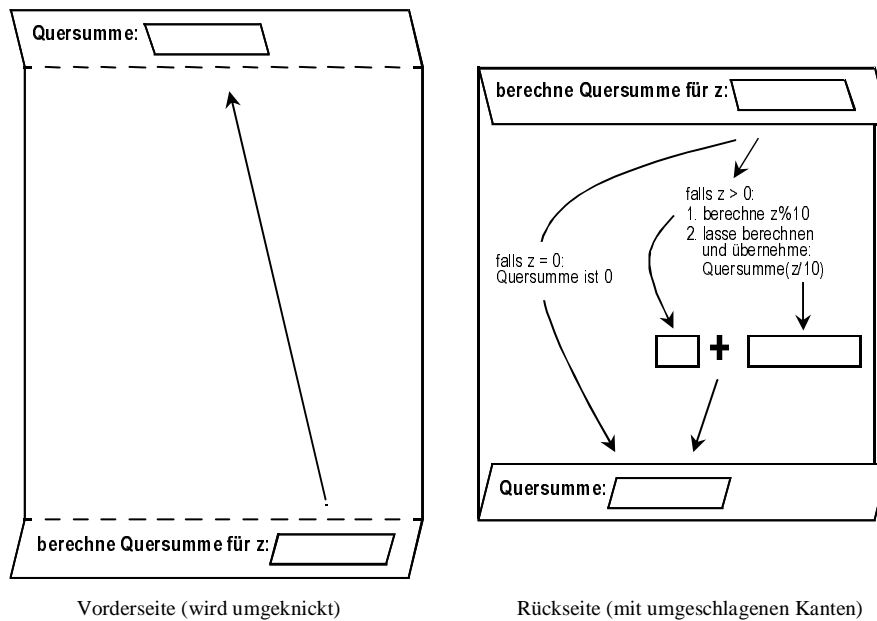


Abbildung 1: Anweisungsblatt zur Berechnung der Quersumme

Das Spiel beginnt, indem ein Mitspieler die Vorderseite eines leeren Arbeitsblatts ausfüllt – also unten eine natürliche Zahl einträgt und die beiden Enden des Blattes umklappt – und dann dieses Blatt an einen anderen Mitspieler weiterreicht, der die Anweisungen auf der Rückseite ausführt. Bei Bedarf nimmt dieser Mitspieler ein weiteres Arbeitsblatt hinzu, füllt dessen Vorderseite aus und aktiviert dann einen weiteren Mitspieler. Führt ein Mitspieler die Abbruchbedingung ($z=0$) aus, so berechnet er unmittelbar das Ergebnis, das er an der Unterkante seines Arbeitsblatts einträgt. Dann reicht er das Arbeitsblatt an seinen Aufrufer zurück; dieser berechnet sein Ergebnis der Addition und die rekursiven Instanzen werden sukzessive abgearbeitet.

Der Autor lässt das Quersummen-Spiel zunächst verdeckt spielen, so dass jeder Mitspieler still sein Arbeitsblatt bearbeitet. Nachdem die Quersumme berechnet wurde, werden alle Arbeitsblätter aufgedeckt und die Arbeitsschritte der Mitspieler von allen Lernenden gemeinsam nachvollzogen. Die Berechnung einer zweiten Quersumme im offenen Spiel dient dann dazu, das rekursive Arbeitsprinzip zu verdeutlichen. Anschließend wird der Algorithmus in eine Programmiersprache übertragen.

4.2 Rekursive Datenstrukturen im Rollenspiel

Das Verständnis von Operationen auf rekursiven Datenstrukturen bereitet einer großen Zahl von Lernenden erhebliche Schwierigkeiten, da ihnen die Dynamik in den Beziehungen zwischen den Elemente zunächst unklar bleibt. Das hier vorgestellte Rollenspiel zur Verdeutlichung von linearen Listen geht von einer nicht objektorientierten Realisierung aus, bei der die strukturbildenden Elemente der Liste und die darauf arbeitenden Operationen voneinander getrennt betrachtet werden können. Wird die lineare Liste im objektorientierten Kontext eingeführt, so sollte sich das Rollenspiel eher an dem in Abschnitt 4.3 vorgestellten Beispiel orientieren. Das Rollenspiel für lineare Listen lässt sich nahezu unverändert auch auf binäre Bäume übertragen. Allerdings muss hierbei beachtet werden, dass für Listen iterative Algorithmen zum Einsatz kommen können, während auf Bäumen natürlicherweise überwiegend rekursive Algorithmen zum Erfolg führen.

Das *Listen*-Rollenspiel ist ein Beispiel für ein freies Spiel mit wenigen und eher unscharfen Hinweisen an die Mitspieler. Ein Lernender übernimmt die Rolle der zu spielenden Operation, beispielsweise des *Einfügens am Ende* einer einfach verketteten linearen Liste. Die übrigen Mitspieler sind die potentiellen Kandidaten für die Elemente einer solchen Liste. Einer dieser Mitspieler wird im Rahmen des *Einfügens* ausgewählt und zum neuen Element gemacht. Er erhält ein Blatt Papier mit den dem Element zugeordneten Inhalten, das er in eine Hand nehmen muss. Die zweite Hand symbolisiert eine verknüpfende Referenz und darf ausschließlich anderen Mitspielern der Rolle Element auf die Schulter gelegt werden. Die wesentliche Regel im Umgang mit diesen Elementen lautet, dass bei jedem Element dem verbindenden Arm nur in die Richtung seiner Hand gefolgt werden darf.

Der Mitspieler, der die Operation *Einfügen* spielt, muss nun einen neuen Mitspieler als Element aktivieren und anhand der durch Mitspieler dargestellten Liste zu deren Ende navigieren und dort den neuen Mitspieler durch eine auf seine Schulter aufgelegte Hand anfügen. Da die Vorgaben in diesem Rollenspiel recht frei formuliert sind, agiert der Lehrer als Berater der Rolle *Einfügen*, ein weiterer Lernender protokolliert die von der Operation vorgenommenen Handlungsschritte. Nach den Erfahrungen des Autors sind die Lernenden in diesem Szenario mit wenigen Hilfestellungen in der Lage, sich die typischen Operationen auf linearen Listen selbst zu erarbeiten. Ein Frontalunterricht, der den Umgang mit linearen Listen einführt, kann weitgehend entfallen. Allerdings muss eine sorgsame und kritische Aufarbeitung des Protokolls erfolgen, um die intuitiv verwendeten Algorithmen präzise herauszuarbeiten.

4.3 Objektorientierte Konstruktion anhand eines Rollenspiels

Objekte in objektorientierten Systemen zeichnen sich dadurch aus, dass sie ergebnisorientierte Handlungen anbieten, deren Ausführung auf intern verborgenen Zustandswerten basieren. Die Animation eines objektorientierten Beispielsystems durch ein Rollenspiel erfolgt im Wesentlichen dadurch, dass der Lernende sich im Sinne einer objektorientierten Denkweise selbst spielt: Der Lernende ist ein intelligentes, hochflexibles und autonomes Objekt, das genau ein Objekt des Beispielsystems simuliert.

Einfache Anweisungen reichen dann aus, um ein solches menschliches Objekt autonom als Systemobjekt arbeiten zu lassen. Das gesamte System wird aus der Menge aller Lernenden gebildet, die Nachrichten austauschen. Sie sind dadurch unmittelbar in die Abläufe integriert und lernen die objektorientierte Denk- und Arbeitsweisen zugleich in ihrer Gesamtwirkung und aus der Sicht eines einzelnen Objekts heraus kennen. Ein Rollenspiel kann im objektorientierten Kontext daher insbesondere dazu eingesetzt werden, um die Kernideen der objektorientierten Konstruktionsprinzipien herauszuarbeiten.

Rollenspiele für die Darstellung eines objektorientierten Systems kommen nicht ohne Anweisungen für jeden einzelnen Mitspieler aus. Da im Gegensatz zu den in den vorangehenden Abschnitten vorgestellten Beispielen jeder Mitspieler eine Rolle mit eigenem Charakter übernehmen muss und diese unterschiedlichen Rollen auch noch zum Erreichen eines gemeinsamen Ziels zusammen arbeiten müssen, müssen insbesondere auch die Aufgabenbereiche klar abgegrenzt werden. Da die Begriffe der Objektorientierung nicht zur Verfügung stehen, wird stattdessen ein System aus menschlichen Spezialisten beschrieben, die nach wenigen vorgegebenen Regeln zusammenarbeiten. Für jeden Rollenspieler, also Spezialisten des Systems, gelten die folgenden vier Grundregeln, die auf natürliche Art das Einhalten der Prinzipien Kapselung, Lokalität und Geheimhaltung erzwingen.

- Er hält sich an eine Menge von für ihn vorgegebenen Handlungsanweisungen.
- Er fragt andere Spezialisten, sobald er an Grenzen seiner Handlungsmöglichkeiten stößt.
- Er hilft anderen Spezialisten, sofern ihn diese um Hilfe bitten.
- Er sichert seinen Status als Spezialist dadurch, dass er die ihm zugeordneten Daten und Handlungsanweisungen streng vertraulich behandelt.

Der Nachrichtenfluss innerhalb des Systems erfolgt durch den Austausch schriftlicher Kurznachrichten von einem Spezialisten zum anderen, die jeweils für die Frage und die zugehörige Antwort verwendet werden. Dadurch ist die Abfolge der Nachrichten deutlich sichtbar, sie kann leicht protokolliert und in einer sich anschließenden Diskussion auch nachvollzogen werden.

Um die Konzentration der Spieler auf die Abläufe des Systems zu ermöglichen, bietet es sich an, ein objektorientiertes System nachzuspielen, dessen Leistung allen beteiligten Lernenden unmittelbar und intuitiv verständlich ist. Der Autor setzt hierzu ein Beispiel aus dem auch in Lehrbüchern [Wi90, Ru91] beliebten Bankbereich ein, einen Ausschnitt aus dem Kontext der Benutzung eines Bankautomaten. Dieser Bankautomat stellt zwei Funktionen bereit, das Abfragen des Kontostands und das Abheben von Bargeld. Für das Ausführen dieser beiden Funktionen wird ein Ausschnitt aus dem zugehörigen Banksystem benötigt, der die Überprüfung der Eingaben, das Ermitteln eines Kontos und dessen Kontostands und die Berechnung von Zinsen umfasst. Der Bankautomat muss gegebenenfalls Banknoten in Höhe des angeforderten Betrags ausgeben können. Im Rollenspiel sind entsprechend Rollen für die Spezialisten *Kundenkontakt*, *Kontenverwaltung*, *Sollzinsberechnung*, *Banknotenausgabe* und für mehrere Spezialisten *Konto*

vorgesehen. Hinzu kommen mehrere Kunden, die den Automaten nutzen. Durch die Zahl der Kunden, die unmittelbar auch die Zahl der notwendigen Konten bestimmt, kann eine flexible Anzahl von Rollen für die Mitspieler geschaffen werden, so dass ihre Zahl der teilnehmenden Lernenden angepasst werden kann.

Die Abbildungen 2 und 3 zeigen exemplarisch die Anweisungen und Daten für die Spezialisten *Kontenverwaltung* und *Sollzinsberechnung*. Die Vorgaben für die anderen Spezialisten können ähnlich kurz gehalten werden.

Spezialist für Kontenverwaltung			
Aufgaben:			
Verwalten der Zuordnung von Kontonummer, Kartenummer, PIN und Kontobezeichnung.			
Diese Zuordnung darf nicht verändert werden.			
Die Zuordnung ist sicherheitskritisch und darf nicht herausgegeben werden.			
Kontonummer	Kartenummer	PIN	Kontobezeichnung
2341	1432	4567	6
5109	9015	3128	4
1344	4413	6891	3
...

Abbildung 2: Anweisungen für Kontenverwaltung

Spezialist für Sollzinsberechnung	
Aufgaben:	
Durchführen der Zinsberechnung!	
Die Formel ist das exklusive Wissen dieses Spezialisten und darf nicht weitergegeben werden!	
Kreditzinsen berechnen sich nach folgender Formel:	
Anzahl der Tage * Zinssatz aus Tabelle * Betrag / 360	
Grenzbetrag	Zinssatz
Fehlbetrag kleiner als Euro 10000	0.10
Fehlbetrag größer gleich Euro 10000	zusätzlich 0.02 für den Fehlbetrag über DM 10000

Abbildung 3: Anweisungen für Sollzinsberechnung

Das Spiel beginnt mit einem Arbeitsauftrag eines Kunden an das System, indem dieser dem Spezialisten für den *Kundenkontakt* die erforderlichen Angaben liefert. Da dieser Spezialist mit den ihm zur Verfügung stehenden Daten lediglich die Vollständigkeit der Angaben prüfen kann, muss er sich zwangsläufig hilfeschend mit einer Nachricht an einen anderen Spezialisten wenden und der Nachrichtenaustausch und damit die Ausführung des Systems beginnen. Der Lehrer muss gelegentlich eingreifen, um dem gerade aktiven Spezialisten Nachrichtentexte, Nachrichteneempfänger oder auch Antworten auf Nachrichten vorzuschlagen. Da die Leistungen der Spezialisten nicht formal definiert sind, kommen immer wieder Anfragen vor, die von dem angesprochenen Spezialisten nicht beantwortet werden können oder sich auf die Herausgabe seiner vertraulichen Daten beziehen. Da auch die Antworten auf Nachrichten nicht definiert sind, kann der angesprochene Spezialist hierauf entsprechend flexibel reagieren. Der entstehende Nachrichtenverkehr zwischen den Spezialisten des Systems wird für die teilnehmenden Lernenden graphisch visualisiert. Werden die beteiligten Spezialisten als Knoten dargestellt und wird für jede ausgetauschte Nachricht eine verbindende Kante eingetragen, so ergibt sich schrittweise ein Kollaborationsdiagramm [RJB99]. Diese Darstellungsform ist in der erlebten Spielsituation unmittelbar verständlich und braucht nicht näher erläutert werden.

Das objektorientierte Rollenspiel zeigt bei seiner Durchführung sehr schnell, dass für das Zusammenwirken der Spezialisten keine detaillierten Kenntnisse fremder Handlungsanweisungen notwendig sind. Die Prinzipien der Lokalität, der Kapselung und der Geheimhaltung der Interna von Spezialisten werden durch die Regeln erzwungen und können anschließend leicht durch den Lehrer expliziert werden.

Für das vertiefte Verständnis der Wirkungsweise der objektorientierten Gestaltungsprinzipien ist es wichtig, dass die Lernenden nach einem ersten spielerischen Umgang mit dem von ihnen selbst realisierten und ausgestalteten System dessen Einzelaspekte zunehmend erkennen und so die Vorzüge der objektorientierten Entwicklung herausarbeiten und explizieren. Das Rollenspiel beginnt daher immer mit einfachen Aufgabenstellungen an das System, denen etwas komplexeren Aufgaben folgen. Das allgemein bekannte Szenario des Bankautomaten ermöglicht es den Lernenden, selbst Sonder- und Ausnahmesituationen zu entdecken und das Verhalten des Systems in diesen Situationen durchzuspielen. Insgesamt wird den Lernenden durch das Rollenspiel deutlich, dass ein System auf der Basis von Kommunikation über Nachrichten tragfähig ist. Etwas umfangreichere Erläuterungen zum Einsatz von Rollenspielen bei der Schulung objektorientierter Konzepte finden sich in [Di01] und [Di03].

5 Zusammenfassung

Rollenspiele bieten eine Möglichkeit, im Rahmen der Ausbildung Programme durch die Lernenden selbst simulieren und analysieren zu lassen. Vielfach lässt sich die Funktionsweise von Algorithmen leicht durch ein Rollenspiel erarbeiten oder nachbilden und so auch Lernenden ohne ausreichende Programmierkenntnisse demonstrieren. Dabei können insbesondere die grundlegenden Prinzipien deutlich gemacht werden.

Die Vorbereitung eines Rollenspiels erfordert im Vergleich zu technischen Animationen sehr wenig Aufwand. Allerdings stellt sich die Frage, ob die zu erwartende Verbesserung des Lernerfolgs den Einsatz von Rollenspielen überhaupt rechtfertigt. Mit der Motivation des Spiels, dem Erläutern der Regeln und der nachträglichen Aufarbeitung der Abläufe des gespielten Systems verbraucht das Rollenspiel recht viel Unterrichtszeit und kann selten mehr als einen ersten Einstieg in die Thematik bilden. Die Erfahrungen des Autors mit diesem Einstieg zeigen jedoch, dass sich der Aufwand im Verlauf der weiteren Ausbildung rentiert, und zwar insbesondere in den folgenden drei Bereichen:

- Ein experimentelles Erarbeiten des Lernziels für die nächste Unterrichtseinheit motiviert die Lernenden sehr viel stärker als dies vorgegebene Zielsetzungen durch den Lehrer vermögen.
- Wird der Lernende bereits am Anfang eines Ausbildungsabschnitts mit den Begriffen des betrachteten Themenbereichs vertraut gemacht, so werden damit auch die Ziele definiert, auf die während der weiteren Ausbildung hingearbeitet werden kann.
- Das Rollenspiel verschafft Lernenden und Lehrer gemeinsame Erfahrungen zu einem Algorithmus oder funktionstüchtigen System. Diese Erfahrungen können bei jeder Diskussion zur Konstruktion weiterer Beispiele herangezogen werden. Jeder Lehrer wird bei Verständnisschwierigkeiten und Fragen gerne an ein verstandenes und operationalisiertes Beispiel anknüpfen. Nach Einschätzung des Autors kann allein hierdurch der Aufwand für das Rollenspiel leicht zurückgewonnen werden.

Der Autor hat Rollenspiele bisher nur in einem Teilbereich der Informatikausbildung eingesetzt, der Softwareerstellung mit Programmierung und dem vorangehenden Algorithmen- und Systementwurf. Es ist aber beispielsweise auch vorstellbar, Hardwarekomponenten wie die Bestandteile eines Prozessors als Rollen nachzuspielen und so die Arbeitsweise der Hardware zu demonstrieren.

Literaturverzeichnis

- [Gu01] Gudjons, H.: Handlungsorientiert lernen. Klinkhardt, 2001.
- [Wö98] Wöll, G.: Handeln: Lernen durch Erfahrung. Schneider, 1998.
- [Wi95] Wilkinson, N.M.; Using CRC Cards, SIGS Books, 1995.
- [Am95] Ambler, S.W.: The Object Primer. SIGS Books, 1995.
- [Wi90] Wirfs-Brock, R. et. al.: Designing Object-Oriented Software. Prentice Hall, 1990.
- [Ru91] Rumbaugh, J. et. al.: Object-Oriented Modeling and Design. Prentice Hall, 1991.
- [RJB99] Rumbaugh, J.; Jacobson, I.; Booch G.: Unified Modelling Language Reference Manual. Addison-Wesley 1999.

- [Di01] Dissmann, S.: Erfahrungen mit dem Einsatz von Rollenspielen zur Einführung in die objektorientierte Denkweise. Net.ObjectDays 2001, Erfurt, S. 355-360.
- [Di03] Dissmann, S.: Einführung objektorientierter Gestaltungsprinzipien anhand von Rollenspielen. SEUH 8 Software-Engineering im Unterricht der Hochschulen, Berlin, dpunkt.verlag 2003, S. 30-40.