

Entwicklung eingebetteter Software in einem Softwarepraktikum mit Lego Mindstorms¹

André Heuer · Kim Lauenroth · Vanessa Stricker · Klaus Pohl

Software Systems Engineering

Institut für Informatik und Wirtschaftsinformatik (ICB)

Universität Duisburg-Essen

{andre.heuer|kim.lauenroth|vanessa.stricker|klaus.pohl}@sse.uni-due.de

Zusammenfassung

In der Bachelor-Veranstaltung Softwareentwicklung und Programmierung (SEP) führen die Studierenden in einem Softwareprojekt die wesentlichen Aktivitäten der Softwareentwicklung vom Requirements Engineering über die Implementierung bis zum Softwaretesten durch. Um den Studierenden einen Einblick in die Besonderheiten und Herausforderungen der Softwareentwicklung für eingebettete Systeme zu geben, wurden mit Hilfe von Studiengebühren Lego-Mindstorms-Baukästen für diese Veranstaltung erworben und eingeführt.

In diesem Beitrag werden die Herausforderungen, deren Lösungsmöglichkeiten und Erfahrungen bei der Durchführung des SEPs mit Lego Mindstorms beschrieben und nötige Anpassungen der Veranstaltung präsentiert. Die Akzeptanz der Einführung der Lego Mindstorms wird anhand einer Umfrage belegt. Zudem vermittelt der Beitrag hilfreiche Informationen zur Durchführung ähnlicher Lehrveranstaltungen mit Lego Mindstorms.

1 Dieser Beitrag wurde teilweise gefördert durch das BMBF-Projekt REMsES, Fördernummer 01 IS F06 D.

1 Einleitung

Die Veranstaltung Softwareentwicklung und Programmierung (SEP) ist nach den aktuellen Prüfungsordnungen ein Bestandteil der praktischen Ausbildung in den Bachelor-Studiengängen Angewandte Informatik – Systems Engineering, Wirtschaftsinformatik sowie im Lehramtsstudiengang Informatik an der Universität Duisburg-Essen. Das SEP wird üblicherweise von Studierenden im 2. oder 3. Semester besucht und verfolgt drei wesentliche Zielsetzungen²:

- Die Vertiefung von Programmierkenntnissen
- Die Vermittlung von Soft-Skills, insb. die teamorientierte Arbeitsweise
- Die Vermittlung elementarer, methodischer Vorgehensweisen zur Softwareentwicklung

1.1 Motivation für die Einführung von Lego Mindstorms

Ein wesentliches Ziel des Studiengangs Angewandte Informatik – Systems Engineering ist die Vermittlung einer systemorientierten Sichtweise auf die Aufgabenstellungen bei der Entwicklung von Software und softwaregestützten Diensten. Eine Systemorientierung in der Softwareentwicklung trägt unter anderem der Tatsache Rechnung, dass die Entwicklung von Software für eingebettete Systeme in den vergangenen Jahren erheblich an Bedeutung gewonnen hat (vgl. [Liggesmeyer und Rombach 2005]). Zum Beispiel ist eingebettete Software ein wesentlicher Bestandteil von modernen Fahrzeugen (vgl. z.B. [Broy 2006]), Systemen in der Avionik, in der Automatisierungstechnik und in der Medizintechnik. Um die Systemorientierung in der Veranstaltung SEP zu verstärken und den Studierenden erste Einblicke in die Entwicklung eingebetteter Software zu bieten, werden seit dem Sommersemester 2008 Lego-Mindstorms-Baukästen in der Veranstaltung eingesetzt. Die Analogien zwischen typischen Bestandteilen eingebetteter Systeme und Lego-Mindstorms-Elementen (siehe Abschnitt 2.2 für eine Beschreibung der Elemente) sind in Tabelle 1 dargestellt.

| Typische Bestandteile eingebetteter Systeme | Lego-Mindstorms-Elemente |
|---|--|
| Elektronische Steuerung | NXT-Baustein |
| Sensoren | Berührungs-, Licht-, Ultraschall-, Geräuschssensor |
| Aktuatoren | Servomotoren |
| Mechanik | Lego-Technik-Bausteine |

Tab. 1 Analogien zwischen eingebetteten Systemen und Lego Mindstorms

2 Für weitere Details zur Veranstaltung SEP verweisen wir auf [Lauenroth et al. 2007].

1.2 Ziel und Aufbau des Beitrags

Der vorliegende Beitrag zeigt am Beispiel SEP die wesentlichen Herausforderungen bei der Einführung von Lego Mindstorms (LM) auf, stellt Lösungen für diese Herausforderungen vor und fasst die Erfahrungen bei der Verwendung von LM in einem Softwarepraktikum zusammen. Der Beitrag soll damit Verantwortliche von Softwarepraktika bei der Entscheidung über die Einführung von LM-Baukästen in der Lehre unterstützen.

Kapitel 2 gibt zu diesem Zweck einen kurzen Überblick über die Kosten der Anschaffung, die Bestandteile eines LM-Baukastens und die im SEP eingesetzte Programmierumgebung. Kapitel 3 zeigt, wie im SEP den folgenden Herausforderungen bei der Durchführung eines Softwarepraktikums mit LM begegnet wurde:

- (H1) Einführung der Studierenden in die Softwareentwicklung für LM
- (H2) Definieren einer geeigneten Aufgabenstellung für die Studierenden, welche die Besonderheiten der Softwareentwicklung für LM berücksichtigt
- (H3) Anpassung des Ablaufs der Veranstaltung bedingt durch den Einsatz von LM

In Kapitel 4 wird ein Resümee aus dem Einsatz von Mindstorms in der Lehrveranstaltung gezogen. Dazu werden die Ergebnisse der im Anschluss an die Veranstaltung durchgeführten Evaluierung vorgestellt und die aufgrund der Verwendung von LM beobachteten Lerneffekte der Studierenden und der Veranstaltungsbereiber beschrieben. Der Beitrag wird in Kapitel 5 mit einer Zusammenfassung und einem Ausblick abgeschlossen.

2 Lego-Mindstorms-Baukästen und Programmierumgebung

In diesem Abschnitt wird ein Überblick über Anschaffungskosten von LM-Baukästen gegeben (Abschnitt 2.1) und der Inhalt eines LM-Baukastens (Abschnitt 2.2) sowie die verwendete Programmierumgebung (Abschnitt 2.3) kurz vorgestellt.

2.1 Kosten für die Anschaffung

In der Veranstaltung SEP werden pro Semester zwischen 10 und 15 Gruppen bzw. Projektteams betreut. Für den Einsatz von LM in der Veranstaltung waren folgende Anschaffungen nötig:

- 18 LM-Baukästen
- Bluetoothdongles für jeden Baukasten, um die Kommunikation zwischen PCs und LM-Robotern zu ermöglichen.
- Sortierkästen für jeden Baukasten zur Aufbewahrung.

Insgesamt wurden für Baukästen, Bluetoothdongles und Sortierkästen ca. 4.000 Euro investiert.

2.2 Inhalt des Baukasten

Der LM-Baukasten beinhaltet verschiedene Hardwareelemente, eine Software zur Programmierung der Roboter sowie Materialien, die die Inbetriebnahme unterstützen sollen. Diese Elemente werden im Folgenden beschrieben:

■ **NXT-Baustein:**

Das Kernstück des LM-Baukastens bildet der programmierbare NXT-Baustein. Dieser Baustein enthält einen 32-Bit-Mikroprozessor, 64-KB-Arbeitspeicher, 256-KB-Flash-Speicher, Lautsprecher sowie Anschlüsse für bis zu vier Sensoren und drei Servomotoren. Weiterhin verfügt er über je eine USB 2.0- und eine Bluetooth 2.0-Schnittstelle, die die Kommunikation mit dem Baustein erlauben, ein LCD zur Darstellung von Ausgaben und vier Bedientasten.

■ **Sensoren:**

Der LM-Baukasten enthält vier verschiedene Sensoren:

- **Berührungssensor:**

Der Berührungssensor erlaubt dem NXT zu erkennen, ob in der Umgebung des Roboters ein Hindernis erreicht wurde.

- **Ultraschallsensor:**

Der Ultraschallsensor ermöglicht es, Distanzen zu Objekten in Zentimeter oder Zoll mittels Ultraschall mit einer Reichweite von bis zu 255 cm und einer Präzision von +/- 3 cm zu messen, um somit auch auf Bewegungen in der Umgebung reagieren zu können.

- **Lichtsensoren:**

Der Lichtsensor ist der zweite Sensor, der dem Roboter das Erkennen seiner Umgebung erlaubt, indem Farben anhand der Lichtintensität erkannt werden können.

- **Geräuschsensor:**

Der Geräuschsensor misst die Lautstärke in der Umgebung des Roboters in Dezibel (bis zu 90 db) und kann zum Beispiel auf lautes Klatschen reagieren.

■ **Servomotoren:**

Drei Servomotoren sind mit eingebauten Rotationssensoren sowie alle benötigten Kabel zur Verbindung der Motoren und Sensoren mit dem NXT-Baustein im Baukasten vorhanden. Die Servomotoren verwenden eingebaute Drehsensoren, die es erlauben, die Achsendrehung auf 1° genau zu beobachten und somit z.B. durch präzise Messung der Roboterbewegungen jederzeit in die ursprüngliche Startposition zurückzukehren [Bagnall 2007].

■ **Legobausteine:**

Der LM-Baukasten enthält über 500 Lego-Technik-Bausteine, u.a. einfache Bausteine, Stangen, Verbindungselemente, Zahnräder und Reifen, wodurch Roboter mit unterschiedlichen Funktionen und Mechaniken gebaut werden können. Darüber hinaus sind zwei Bälle und Zangenelemente enthalten.

■ **Dokumentationsmaterialien:**

Der Baukasten enthält verschiedene Dokumentationsmaterialien, die die Inbetriebnahme des NXT erleichtern. Neben einem Quickstart Guide wird außerdem eine Bedienungsanleitung bereitgestellt, die eine Einführung in die NXT-Technik enthält und das Zusammenbauen von vorgefertigten Modellen mit Hilfe der mitgelieferten Teile erläutert. Ein solches Model, genannt Tribot (siehe Abb. 1), wurde im SEP verwendet, da es durch die Verwendung von drei Rädern, aller Sensoren sowie einer Greifzange den Anforderungen aus der Aufgabenstellung (siehe Abschnitt 3.2) gerecht wird.



Abb. 1 Foto eines Tribots

■ **Programmierungsumgebung:**

Zur Programmierung des NXT-Bausteins wird die Software NXT-G (für PC oder Mac) zur Verfügung gestellt. NXT-G basiert auf der grafischen Programmiersprache labVIEW (vgl. [Georgi und Metin 2007]) und erlaubt die Programmierung des NXT-Bausteins mit Hilfe von grafischen Elementen.

2.3 Programmierungsumgebung leJOS

Die Programmiersprache Java (siehe z.B. [Horstmann und Cornell 2007]) wird im 1. Semester für alle Informatikstudiengänge gelehrt und ist daher den Studierenden im SEP bekannt. Um an die im ersten Semester vermittelten Java-Kenntnisse der Studierenden anzuknüpfen, wurde im SEP die Programmierungsumgebung leJOS (vgl. [Bagnall 2007]) zur Entwicklung von Software für den NXT-Baustein eingesetzt. leJOS portiert einen Teil der Java Virtual Machine und stellt ein reduziertes Java für die Programmierung des NXT-Bausteins zur Verfügung. Es ist mit Windows, Linux und Macintosh kompatibel, so dass für die Studierenden keine Einschränkungen bezüglich ihres gewohnten Entwicklungssystems entstanden. Das Ausführen und Hochladen von bereits geschriebenem leJOS Programmcode ist über die Konsole möglich. Darüber hinaus wird ein Eclipse Plug-in für leJOS zur Verfügung gestellt, das die Programmierung der LM mit leJOS in Eclipse (siehe z. B. [Holzner 2005]) ermöglicht.

3 Einbindung von Lego Mindstorms in das Veranstaltungskonzept des SEP

In diesem Abschnitt werden die in Abschnitt 1.2 genannten Herausforderungen bei der Durchführung eines Softwarepraktikums mit LM aufgegriffen. Abschnitt 3.1 beschreibt, wie die Studierenden in die Programmierung von LM eingeführt wurden (H1). In Abschnitt 3.2 wird die Aufgabenstellung für den Einsatz von LM vorgestellt (H2). In Abschnitt 3.3 werden die wesentlichen Unterschiede in der Durchführung der Veranstaltung mit LM gegenüber früheren Veranstaltungen skizziert (H3).

3.1 Einführung der Studierenden in die Programmierung von Lego Mindstorms

Um den Studierenden einen möglichst einfachen Einstieg in die Programmierung der LM zu geben, wurden die erweiterten Lehrinhalte didaktisch in mehrere Schulungsteile aufbereitet. Die begleitenden Schulungsmaterialien, in Form von Powerpointfolien, wurden teilweise durch Beispielprogramme ergänzt. Die gesamte Einführung in die LM wurde in die folgenden drei Schulungsteile unterteilt:

- **Allgemeine Einführung in den LM-Baukasten:**
Zu Beginn des ersten Schulungsteils haben alle Studierende eine Einführung in die Bestandteile des LM-Baukastens erhalten. Diese Einführung hat die Funktionalitäten des NXT-Bausteins, der Sensoren und der Servomotoren erläutert. Das begleitende Schulungsmaterial umfasst acht Powerpointfolien.
- **Programmierung der LM mit leJOS:**
Dieser Teil der Schulung gab den Studierenden eine Einführung in die leJOS-API. Anhand von einfachen Programmbeispielen wurden die Ansteuerung der Servomotoren und das Auslesen von Sensordaten erläutert. Das begleitende Schulungsmaterial umfasst 31 Powerpointfolien und ein Beispielprogramm, das sämtliche Sensoren und Servomotoren des LM-Baukastens ansteuert.
- **Wegverfolgungsalgorithmus für LM:**
Im letzten Teil der Schulung wurde den Studierenden ein einfacher Algorithmus zur Verfolgung von Wegen mit Hilfe des Lichtsensors vermittelt. Dieser Schulungsteil war speziell auf die Aufgabenstellung zugeschnitten, um den Studierenden eine mögliche Lösung für das Wegverfolgungsproblem anzubieten. Das Schulungsmaterial umfasst zehn Powerpointfolien und ein Beispielprogramm, das den Wegverfolgungsalgorithmus umsetzt.

Das vorgestellte Schulungsmaterial wurde hauptsächlich durch die beiden studentischen Hilfskräfte der Veranstaltung in der vorlesungsfreien Zeit erstellt. Insgesamt wurden dafür ca. 80 Arbeitsstunden aufgewandt.

3.2 Aufgabenstellung »TanSEPort«

Die bisherigen Aufgabentypen, die im SEP zum Einsatz kamen, umfassten die Anwendungs- und Spieleentwicklung (vgl. [Lauenroth et al. 2007]). Für beide Arten wurden die Aufgaben so gestellt, dass sie eine hinreichende Komplexität boten, um den gewünschten Lerneffekt bei den Studierenden zu erreichen, ohne sie dabei zu überfordern. Weiterhin wurden Szenarien und Features in den Softwareentwicklungsprozess eingeführt, um die Qualität der einzelnen Entwicklungsaktivitäten zu erhöhen (vgl. [Lauenroth et al. 2007]). Die neue Aufgabenstellung, die den Einsatz von LM vorsieht, sollte eine ähnliche Komplexität aufweisen und der gleichen Methodik folgen. Die gestellte Aufgabe wird im Folgenden erläutert.

Im Sommersemester 2008 bestand die Aufgabe des SEPs in der Durchführung des Projektes »TranSEPort«. Die Aufgabenstellung sah es vor, ein System zu entwickeln, das den Transport von Waren in einer bestimmten vorgegebenen Umgebung erlaubt. Diese Umgebung enthält Städte und Straßen, wobei die Straßen auch in Kreuzungen münden können. Bestandteil der Aufgabenstellung war es, bis zu drei Roboter gleichzeitig durch einen Server zu koordinieren, um Aufträge zum Transport der Waren zwischen den Städten abzarbeiten. Dabei sollen die Roboter möglichst selbstständig ihren Weg zur Ausführung des ihnen zugeteilten Auftrags finden.

Entsprechend wurde durch die Aufgabenstellung vorgegeben, dass eine Client-Server-Struktur von den Studierenden zu entwickeln war. Der Server musste folgende Aufgaben übernehmen:

- Verwaltung des Warenbestands in den einzelnen Städten
- Empfang neuer Transportaufträge
- Koordinierte Transportabwicklung inkl. Routenplanung und Kollisionsvermeidung

Der Client musste folgende Funktionen umsetzen:

- Eigenständige Funktion zur Straßenerkennung
- Erkennung von Kreuzungen und Städten
- Aufnahme und Ablage von Waren an Städten unter Verwendung der Greifzange (Waren wurden durch die mitgelieferten Bälle repräsentiert)
- Kommunikation zwischen Server und Roboter inkl. Auftragserteilung des Servers an den Roboter und Positions- und Sensordatenüberwachung über Bluetooth
- Auftragsbefreiung

3.3 Durchführung der Veranstaltung mit Lego Mindstorms

Der Ablauf des SEPs orientiert sich an den Phasen des in [Lauenroth et. al. 2007] beschriebenen Softwareentwicklungsprozesses (Requirements Engineering, Design, Prototypentwicklung, Implementierung und Test). Im Folgenden werden zusätzliche Lerneffekte aufgelistet, die bei den Studierenden hervorgerufen wurden, indem einige Informationen und Ressourcen nicht sofort zur Verfügung gestellt wurden:

■ **Ressourcenknappheit:**

Die Ressourcenknappheit in Softwareprojekten ließ sich beim Einsatz der LM gut vermitteln, indem beispielsweise die LM-Baukästen erst während der Designphase herausgegeben wurden. Das heißt, die Studierenden mussten sich bereits mit den Anforderungen an das zu entwickelnde System zu einer Zeit auseinandersetzen, zu der sie noch keinen Zugriff auf die einzusetzende Hardware hatten. Dies hat auch dazu geführt, dass die Studierenden erlernt haben, wie sie die Spezifikation eines Hardwaresystems nutzen können, um ausreichend Informationen für die Ableitung von Anforderungen gewinnen zu können. Weiterhin waren die Studierenden somit gezwungen, die Erarbeitung des Designs mit der Einarbeitung in die LM zu koordinieren.

■ **Hilfestellung zu Programmierinhalten:**

Die Bereitstellung der Hilfestellung erfolgte erst während der Erstellung des Prototypens. Dadurch wurde zum einen darauf abgezielt, dass die Studierenden sich während der Erstellung des Designs nicht von der Implementierung leiten lassen. Zum anderen haben die Studierenden erfahren, dass eine systematische und geplante Vorgehensweise wichtig ist, um das doppelte Durchführen von Aktivitäten zu vermeiden und existierende Ergebnisse effizient wiederverwenden zu können.

■ **Zeitdruck:**

Um Zeitdruck bei den Studierenden entstehen zu lassen, wurde die Testphase durch mehrere Faktoren beeinflusst. Zum einen sah die Aufgabenstellung vor, dass das System bis zu drei LM gleichzeitig steuern sollte, so dass den Studierenden ein sehr restriktiver Zugang zu weiteren LM-Robotern gewährt werden musste. Dieser Zugang musste zusätzlich mit den anderen Gruppen, die an dem SEP teilnahmen, koordiniert werden. Dies galt ebenfalls für die Testumgebung. Die Testumgebung wurde erst während der Testphase selbst zur Verfügung gestellt. In der Woche vor Beginn der Tests wurde allerdings eine Karte der Testumgebung in Form einer Grafikdatei zur Verfügung gestellt (nicht maßstabsgetreu), die Auskunft über die Lage der Städte und Straßen gab. Durch diesen restriktiven Zugang zu der Testumgebung, waren die Studierenden gezwungen, ihre Testfälle detailliert zu entwerfen und effizient durchzuführen.

■ Späte Änderung der Testumgebung:

Um den Studierenden zu vermitteln, dass späte Änderungen an der Spezifikation der Testumgebung keine Seltenheit sind, wurden gezielt die Farben, die zur Kennzeichnung der Städte und Kreuzungen verwendet wurden, während der laufenden Tests geändert. Somit mussten zum einen die entsprechenden Komponenten der Implementierung kurzfristig angepasst werden und die entsprechenden Tests wiederholt durchgeführt werden. Zum anderen haben die Studierenden durch die ausgeführten Tests bereits vor der Änderung die Notwendigkeit der Anpassung erkannt, da das System mit den vorgegebenen Markierungen nicht realisierbar war.

4 Evaluierung der Verwendung von Lego Mindstorms im SEP

In diesem Abschnitt wird die Evaluierung der Verwendung von LM-Baukästen im SEP beschrieben. Zunächst wird in Abschnitt 4.1 auf die Evaluierung der Veranstaltung durch die Studierenden mittels Fragebögen eingegangen. Im Anschluss daran werden im darauffolgenden Abschnitt die Lerneffekte seitens der Studierenden beschrieben. In Abschnitt 4.3 werden die Erfahrungen der Betreuer in Hinblick auf die beobachteten Lerneffekte beschrieben.

4.1 Veranstaltungsevaluierung durch die Studierenden

Die Evaluierung der Veranstaltung durch die Studierenden ist in drei Teile aufgeteilt. Zunächst wird der Aufbau der Evaluierung erläutert, anschließend die Durchführung der Evaluierung. Abgeschlossen wird mit der Auswertung und der Interpretation der Ergebnisse der Evaluierung aus Sicht der Studierenden.

Aufbau der Evaluierung

Um die Einführung der LM-Baukästen zu evaluieren, wurde von den Studierenden die Lehrveranstaltung anhand von Fragebögen am Ende der Veranstaltung bewertet. Um den Erfolg der Veranstaltung zu bewerten, waren die folgenden Fragestellungen von maßgebender Bedeutung:

1. Wurden die definierten Ziele aus Kapitel 1 weiterhin erreicht:
 - a) Vertiefung von Programmierkenntnissen
 - b) Vermittlung von Soft-Skills, insb. die teamorientierte Arbeitsweise
 - c) Vermittlung elementarer, methodischer Vorgehensweisen zur Softwareentwicklung
2. Ist die Verwendung von Studiengebühren zum Erreichen der Ziele angebracht?

Um die beiden genannten Fragestellungen zu beantworten, wurden die folgenden sieben Fragen mit entsprechenden Kontrollfragen in den abschließenden Mei-

nungstest integriert (in den Klammern ist die Fragestellung angegeben, welche der Fragestellungen aus dem vorherigen Absatz evaluiert werden sollte). Sie sollten auf einer Skala (sehr zutreffend, zutreffend, mittel, unzutreffend, sehr unzutreffend) beantwortet werden:

- (1) Die vorgestellten theoretischen Inhalte wurden durch die Entwicklung mit Mindstorms-Roboter verdeutlicht (Fragestellung 1.c).
- (2) Die Entwicklung der Lejos-Software zeigte mir, dass eine Software ohne die Spezifikation von Anforderungen schwierig zu realisieren ist (Fragestellung 1.c).
- (3) Das Arbeiten mit Mindstorms-Robotern hat mir Spaß gemacht.
- (4) Das Arbeiten mit Mindstorms-Robotern motivierte mich, meine Programmierkenntnisse zu vertiefen (Fragestellung 1.a).
- (5) Die Verwendung von Studiengebühren zur Anschaffung der Mindstorms-Roboter halte ich für sinnvoll (Fragestellung 2).
- (6) Nach diesem Projekt hat sich mein Interesse an der Softwareentwicklung gesteigert (Fragestellung 1.a).
- (7) Das Arbeiten in einem Softwareentwicklungsteam ist entscheidend für den Erfolg des Projektes (Fragestellung 1.b).

Die Fragestellung (3) dient der Prüfung der Motivation der Studierenden, die durch den Einsatz der LM-Baukästen erreicht wurde.

Durchführung der Evaluierung

Zur Befragung der Studierenden wurden 130 Fragebögen ausgeteilt. Die Studierenden hatten ca. 20 Minuten Zeit, den Fragebogen auszufüllen. Es wurden 109 ausgefüllte Fragebögen zurückgegeben, was einer Rücklaufquote von ca. 77% entspricht.

Auswertung und Interpretation

Die Ergebnisse der Befragung sind in Abbildung 2 als Balkendiagramme zu finden. Zur folgenden Interpretation der Ergebnisse werden nicht negative Bewertungen als Zustimmung der Studierenden gewertet, d.h., die Kategorien »mittel«, »zutreffend« und »sehr zutreffend« werden als Zustimmung zu den vorgestellten Items gewertet:

- (1) 90 Studierende (88%) äußerten ihre Zustimmung. Wir schließen daraus, dass die von uns vermittelten theoretischen Inhalte trotz der Einführung der LM klar und deutlich vermittelt wurden. Dieses Bild gilt sowohl für die Studierenden des Studiengangs Angewandte Informatik sowie auch der anderen Studiengänge.
- (2) 92 Studierende (92%) äußerten ihre Zustimmung. Wir schließen daraus, dass den Studierenden anhand des entwickelten Spieltriebs für den LM die

Wichtigkeit zur Spezifikation von Anforderungen in Hinblick auf ein methodisches Vorgehen deutlich wurde.

- (3) Fast alle Studierenden (97 %) äußerten ihre Zustimmung. Wir schließen daraus, dass die Motivation der Studierenden durch den Einsatz der LM-Baukästen hoch war.
- (4) Fast alle Studierende (97 %) äußerten ihre Zustimmung. Wir schließen daraus, dass das Veranstaltungsziel zur Vertiefung der Programmierkenntnisse erreicht wurde.
- (5) 99 Studierende (94 %) äußerten ihre Zustimmung. Wir schließen daraus, dass die Studierenden der Investition ihrer Studiengebühren in die LM-Baukästen positiv gegenüberstanden.
- (6) 86 Studierende (82 %) äußerten ihre Zustimmung. Wir schließen daraus, dass das Interesse der Studierenden an Softwareentwicklung weiter gestiegen ist.
- (7) Fast alle Studierenden (98 %) äußerten ihre Zustimmung. Wir schließen daraus, dass die Studierenden die Notwendigkeit der Arbeit in einem Team und die Gruppendynamik innerhalb des Teams erfahren haben.

Insgesamt lässt sich durch die Evaluierungsergebnisse festhalten, dass aus studentischer Sicht die Einführung der LM-Baukästen einen positiven Einfluss auf die Veranstaltung hatte. Auch die Motivation der Studierenden ist im Vergleich zum Sommersemester 2007 gestiegen, wo 90 % (118 von 131 Studierenden) Zustimmung zur Frage »Mir hat das SEP Spaß gemacht« gaben. Insgesamt wurden die Ziele der Veranstaltung erreicht, und den Studierenden konnte ein Einblick in die Entwicklung von Software für eingebettete Systeme gegeben werden.

4.2 Lerneffekte durch den Einsatz der Lego Mindstorms

Durch die Einführung der LM-Baukästen ergaben sich weitere neue Herausforderungen und zusätzliche Lerneffekte für die Studierenden, die durch die Spiele- und Anwendungsentwicklung nicht erzielt werden konnten. Im Folgenden werden diese Lerneffekte anhand der einzelnen Phasen der Veranstaltung vorgestellt:

■ Requirements Engineering:

Durch den Einsatz von Hardware im SEP haben die Studierenden während des Requirements Engineering die Erfahrung gemacht, dass nicht allein die Aufgabenstellung als Anforderungsquelle zu berücksichtigen war, sondern auch die Eigenschaften und Fähigkeiten der zur Verfügung stehenden Hardware (z.B. Sensoren und Servomotoren) und die Umgebung des Systems. Hierzu mussten die Studierenden u.a. in den Szenarien für das geplante System auf die Umgebung und auf die verwendeten Hardwarebestandteile eingehen. Zum Beispiel muss das Szenario »Aufnehmen von Ware« beschreiben,

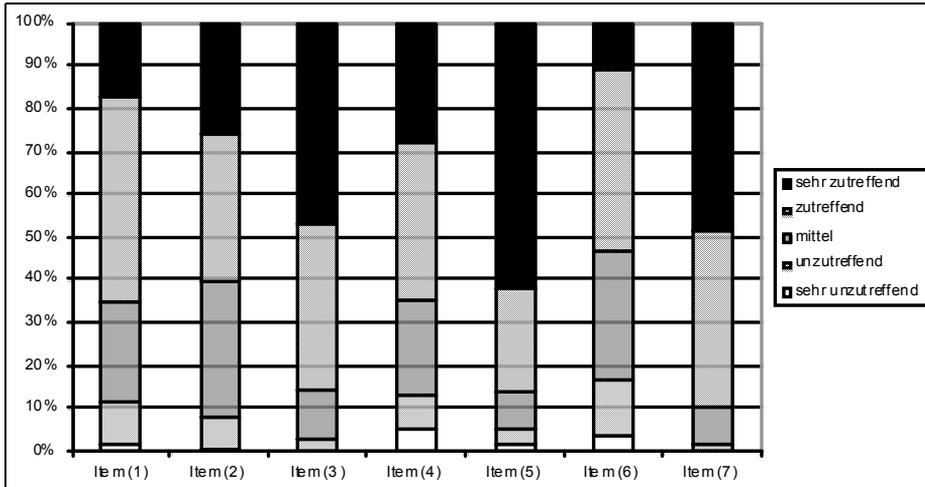


Abb. 2 Verteilung der Antworten der Studierenden zu den sieben Items

wie das System mit Hilfe der Sensoren und Motoren zum Ablageplatz der Ware findet und wie die Ware aufgenommen wird.

■ Design:

Ziel des Designs mit den LM war es, den Studierenden den Unterschied zwischen Hardware- und Software-Design zu vermitteln. Im Gegensatz zu früheren Aufgaben waren nicht mehr ausschließlich zu kontrollierende Softwarekomponenten sowie deren Schnittstellen zu berücksichtigen, sondern auch zu kontrollierende Hardwarekomponenten. Weiterhin konnten die Studierenden auch erkennen, dass die sorgfältige und frühzeitige Planung des Deployment der einzelnen Komponenten auf der verfügbaren Hardware ein gutes Design ausmacht. Während des Designs wurden unter anderem Message Sequence Charts zur Konkretisierung der Szenarien aus den Anforderungen verwendet, wobei die Studierenden z.B. die Sensoren der Roboter in den MSCs berücksichtigen sollten.

■ Prototyp:

Der zu entwickelnde Prototyp zeigte den Studierenden, dass dieser nicht nur die grafische Gestalt des Systems darstellen kann, sondern zum Beispiel auch für eine frühzeitige Erprobung einer rudimentären Kommunikation zwischen Server und Robotern verwendet werden konnte. Hierzu wurde für den Prototypen gefordert, eine manuelle Steuerung für den Roboter zu realisieren. Damit wurde bereits durch den Prototyp sichergestellt, dass sich die Studierenden möglichst frühzeitig mit der Kommunikation zwischen Server und Robotern auseinandersetzen, da dies aus Sicht der Betreuer den schwierigsten Teil der Aufgabenstellung darstellte.

■ **Implementierung:**

Durch die Programmierung mit leJOS waren die Studierenden gezwungen, sich in eine neue Entwicklungsumgebung und die wichtigsten Elemente der zugrunde liegenden API in möglichst kurzer Zeit einzuarbeiten. Um den Studierenden den Einstieg in die Implementierung zu erleichtern, wurden in dieser Phase verschiedene Quellcodebeispiele zur Verfügung gestellt und erläutert, z.B. eine Beispielimplementierung für den Algorithmus zur Straßenverfolgung.

■ **Test:**

Beim Test konnten die Studierenden erfahren, welche Schwierigkeiten eine fremde und nicht frei verfügbare Testumgebung mit sich bringt, da es nicht möglich war, das entwickelte System ausschließlich in einer von ihnen erzeugten Testumgebung ausreichend zu testen. Stattdessen musste auch die Laufzeitumgebung in den Fokus des Tests gerückt werden. Hierzu wurde rechtzeitig zum Testbeginn eine geeignete Testumgebung bereitgestellt.

Ein Video, das die entwickelten Systeme während der Projektabnahme in Aktion zeigt, ist unter <http://sep.icb.uni-due.de> zu finden.

4.3 Erfahrungen

Aus Sicht der Veranstaltungsbetreuung sind die folgenden Lerneffekte im Zusammenhang mit der Einführung der LM eingetreten:

■ **Fehlerhafte leJOS-Bibliotheken:**

Während der Implementierung der Software hat sich herausgestellt, dass die verwendete Version von leJOS und die entsprechenden Java-Bibliotheken teilweise fehlerhaft waren und durch ältere Versionen ersetzt werden mussten. Dieser Umstand stellte einen guten Lerneffekt für die Studierenden dar, da sie auf die Änderung der Entwicklungsumgebung reagieren mussten.

■ **Schwierigkeiten mit der Eclipse-Umgebung:**

Zur Entwicklung von Software mit leJOS werden Plug-ins für Eclipse zur Verfügung gestellt, die Aufgaben wie das Kompilieren und Kopieren der Dateien auf den NXT übernehmen. Die Verwendung von Eclipse wurde im SEP empfohlen. Für einige Studierende im 2. Semester stellte Eclipse durch seinen Funktionsumfang jedoch eine zu große Herausforderung dar und wurde daher nicht von allen Gruppen genutzt.

■ **Unerwartete Umgebungseffekte:**

Während der Nutzung der Testumgebung durch die Studierenden hat sich herausgestellt, dass die Lichtsensoren empfindlich auf wechselnde Lichtverhältnisse innerhalb der Testumgebung reagierten. Für die Studierenden hatte

dies zur Konsequenz, dass z.B. Kalibrieringsroutinen für die Lichtsensoren in das System integriert werden mussten.

■ Seiteneffekte in der Hardware:

Die Roboter zeigten unerwartete Seiteneffekte, die erst während des Betriebs auftraten. So wurden z.B. bei schwachen Batterien die Servomotoren mit der Drehungsmessung sehr unzuverlässig.

5 Zusammenfassung und Ausblick

In diesem Beitrag haben wir die Einführung von Lego-Mindstorms-Baukästen in ein Softwarepraktikum beschrieben. Das Ziel dieser Einführung bestand unter anderem darin, das Aufgabenspektrum der Veranstaltung um die Programmierung von Software für eingebettete Systeme zu erweitern. Für die Anschaffung der Baukästen sind Kosten in Höhe von ca. 4.000 Euro angefallen. Für die Erstellung der Schulung und der Schulungsmaterialien wurden ca. 80 Arbeitsstunden aufgewendet. Wir bewerten diese Kosten für die Einführung der Baukästen im Vergleich zu den Personalkosten im Lehrbetrieb einer Hochschule als vergleichsweise gering.

Die Akzeptanz dieser Maßnahme durch die Studierenden wird durch die positiven Evaluierungsergebnisse deutlich. Hervorhebenswert ist, dass über 94 % der Studierenden die Verwendung von Studiengebühren für die Anschaffung der Baukästen als sinnvoll erachtet haben. Aus Sicht der Betreuer konnte durch die Anschaffung der LM-Baukästen Lerneffekte bei den Studierenden erzeugt werden, die typisch für eingebettete Systeme sein können (z.B. unerwartete Einflüsse aus der Systemumgebung).

Für die kommenden Veranstaltungen planen wir, den Studierenden die Art der zu bearbeitenden Aufgabenstellung zur Wahl zu stellen, d.h. Programmierung einer Anwendung, Programmierung eines Spiels und Programmierung mit LM-Baukästen. Wir wollen im Rahmen der Anmeldung zur Veranstaltung die Interessen der Studierenden für einen Aufgabentyp abfragen und dann entsprechend der geäußerten Interessen viele Gruppen mit verschiedenen Aufgabentypen anbieten. Den Fokus unserer weiteren Forschungsarbeit werden wir insbesondere darauf richten, ob die Art der Aufgabenstellung einen Einfluss auf den Lerneffekt ausübt.

Danksagung

Wir möchten einen besonderen Dank an unsere studentischen Hilfskräfte Florian Buchloh und Thomas Keller für ihre wertvollen Beiträge zur Gestaltung des SEPs sowie für ihren unermüdlichen Einsatz bei der Betreuung von SEP-Gruppen und bei der Erfassung der Evaluierungsbögen aussprechen.

Literatur

- [Bagnall 2007] B. Bagnall: Maximum LEGO NXT: Building Robots with Java Brains, Variant Press, Winipeg, 2007.
- [Broy 2006] M. Broy: Challenges in automotive software engineering. In Proceedings of the 28th International Conference on Software Engineering; ICSE '06. ACM, S. 33-42, 2006.
- [Georgi und Metin 2007] W. Georgi, E. Metin: Einführung in LabVIEW, Hanser Fachbuchverlag, 2007.
- [Holzner 2005] S. Holzner: Eclipse Cookbook, O'Reilly Media, 2005.
- [Horstmann und Cornell 2007] C. S. Horstmann, G. Cornell: Core Java 2 Vol.1: Fundamentals, Prentice Hall International, 8. Auflage, 2007.
- [Lauenroth et al. 2007] K. Lauenroth, E. Sikora, K. Pohl: Positive Effekte von Szenarien und Features in einem Softwarepraktikum. In: Software Engineering im Unterricht der Hochschulen, SEUH 10, dpunkt.verlag, 2007.
- [Liggesmeyer und Rombach 2005] P. Liggesmeyer, D. Rombach (Hrsg.): Software Engineering eingebetteter Systeme – Grundlagen – Methodik – Anwendungen. Elsevier, Heidelberg, 2005.