

# Ein Middleware-Kurs mit mobilen Robotern

Michael Mock<sup>1)</sup>, Walter Nowak<sup>2)</sup>

<sup>1)</sup>Fraunhofer AIS  
Schloss Birlinghoven  
53754 St. Augustin  
michael.mock@ais.fraunhofer.de

<sup>2)</sup>B-IT Bonn-Aachen International Center  
for Information Technology  
53754 St. Augustin  
walter.nowak@ais.fraunhofer.de

**Abstract:** Middleware-Systeme leisten einen entscheidenden Beitrag, um dynamische Gruppen von mobilen Systemen, Robotern und Geräten zu verwalten. Dieses Papier stellt einen Kurs mit JavaSpaces und Jini vor, einer frei verfügbaren Middleware von SUN für mobile Geräte. Der Kurs besteht aus einführenden Vorlesungen und einem studentischen Teamprojekt, in dem Middleware-Technologien zum Design und zur Umsetzung des Rent-a-Robot Szenarios eingesetzt werden. In diesem Szenario interagieren mobile Roboter dynamisch mit wechselnden Benutzern. Der curriculare Aufbau des Kurses und Erfahrungen mit dem Projekt werden präsentiert.

## 1 Einleitung

Middleware unterstützt die Verwaltung hochgradig dynamischer Beziehungen, wie sie in der mobilen Robotik und bei der Verwendung intelligenter Geräte oft auftreten. Mobile Systeme müssen in verschiedene Kontexte dynamisch eingebunden werden oder diese verlassen. Middleware stellt ein wichtiges Mittel zur Vereinfachung und Beherrschung solcher Systeme dar. Der in diesem Papier beschriebene Kurs ermöglicht es Studierenden im Robotikumfeld, die Konzepte und Ideen von Middleware zu verstehen, sowie diese zur Erstellung einer dynamischen, verteilten Anwendung unmittelbar einzusetzen.

Robotik stellt hierbei nicht nur ein natürliches Umfeld dar, sondern kann ebenso einen positiven Einfluss auf die Motivation haben. Viele Studenten und Studentinnen sind deutlich motivierter und fühlen sich mehr selbstbestätigt, wenn sie mit realen, sich bewegenden Objekten umgehen können, anstatt mit rein abstrakten Software-Komponenten. Dies trifft insbesondere dann zu, wenn sie gar etwas Eigenes erschaffen können, wie in unserem Fall eine komplexe, verteilte, kooperative und dynamische Anwendung.

Dennoch führt der Einsatz von Middleware und Robotern nicht automatisch zu den oben beschriebenen positiven Effekten. Das Problem liegt darin, Middleware-Konzepte in einer solchen Form einzuführen, dass sie auch unmittelbar für eine komplexe Anwendung eingesetzt werden können. In diesem Beitrag zeigen wir, wie das sorgfältige Design und die Organisation eines Kurses mit nur 2+2 Wochenstunden in einem Semester zu den oben genannten Zielen führen können. Design beinhaltet hierbei nicht nur Vorlesungen und gut vorbereitete Übungen, sondern insbesondere das Kursprojekt Rent-a-Robot und die Integration all dieser Teile. Der Kurs wurde von den Autoren im Master Programm “Autonomous Systems” gehalten, welches vom B-IT Bonn-Aachen International Center for Information Technology an der Fachhochschule Bonn-Rhein-Sieg in Kooperation mit dem Fraunhofer Institut AIS angeboten wird. Folien zur Vorlesung und die Übungen sind im Internet auf der Homepage der Autoren [www.ais.fraunhofer.de/~mock](http://www.ais.fraunhofer.de/~mock) verfügbar.

Unseres Wissens ist dieser Kurs einzigartig in der Art und Weise, wie Middleware-Konzepte eingesetzt werden in Kombination mit einer relativ komplexen, verteilten Roboterapplikation, welche von den Studierenden selbst in einem Semester entwickelt und umgesetzt wird. In Bezug auf die Nutzung von Lego Mindstorms führt der Kurs den Ansatz weiter, der auf Schulniveau zum Beispiel im Roberta Projekt [MPT04] bereits erfolgreich eingesetzt wurde.

## 2 Ein Middleware-Kurs mit mobilen Robotern

In diesem Kapitel werden die Struktur des Kurses, einschließlich des Aufbaus der Vorlesungen und Übungen, sowie das Projektszenario beschrieben.

### 2.1 Voraussetzungen und Lernziele

Die Voraussetzungen und Lernziele des Kurses sind in Tabelle 1 aufgelistet.

Voraussetzungen	Lernziele
<ul style="list-style-type: none"> <li>- Grundlegende Software-konzepte</li> <li>- Programmier-erfahrung</li> </ul>	<ul style="list-style-type: none"> <li>- Lernen von Problemen und Ansätzen im Zusammenhang mit vernetzten mobilen Systemen</li> <li>- Praktische Erfahrungen mit Java und Jini als Middleware für vernetzte Systeme machen</li> <li>- Erfahrungen mit verhaltensbasierter Programmierung mobiler Roboter</li> <li>- Einen vollständigen Software-Entwurfsprozess in einem Team unter Benutzung von XP (eXtreme Programming) lernen und umsetzen</li> <li>- Erstellen von Rent-a-Robot, einem Leasing-System für mobile Roboter</li> </ul>

Tabelle 1: Voraussetzungen und Lernziele des Kurses

Verschiedene konkurrierende Middleware-Umsetzungen existieren auf dem Markt, darunter CORBA, Microsoft UPnP und .NET, Jini und JavaSpaces von Sun. Zu einem gewissen Grad sind die Konzepte hinter diesen Varianten vergleichbar. Das auf Java basierende Jini und JavaSpaces [FHA99] von Sun ist hierbei eine frei verfügbare und einfach zu benutzende Lösung, welche sich somit gut zu Lehrzwecken eignet.

## 2.1 Die Rent-a-Robot Anwendung

Die Idee des Projektes ist, ein motivierendes Szenario zu nehmen, welches leicht zu verstehen aber nicht trivial ist, und das innerhalb eines Ein-Semester-Kurses umgesetzt werden kann. Für das Rent-a-Robot Projekt nehmen wir an, dass verschiedene Roboter mit unterschiedlichen Fähigkeiten bestimmte Aufgaben ausführen können. In unserem Beispielszenario nähert sich ein Benutzer den Robotern, erhält zunächst einen Überblick über die verfügbaren Roboter, und kann diesen dann Aufgaben zuweisen. Dieses Szenario erfordert ein verteiltes, dynamisches Programmkonzept, da Roboter und Benutzer jederzeit neu hinzukommen oder verschwinden können. Aufgaben können zu jeder Zeit zugewiesen werden, so dass sich der Zustand der Roboter dynamisch ändert. In unserem Kurs haben wir uns für Lego Mindstorms Roboter entschieden. Diese sind einfach zu benutzen und zu programmieren. Der Einsatz von Java und LeJOS [LeJOS] passt zudem zu der restlichen Programmierumgebung (JavaSpaces etc.). Auch andere Roboterplattformen wurden in Erwägung gezogen, unter ihnen Sony Aibos, Volksbots und Khepera Roboter, kamen in diesem Kurs aber bisher nicht zum Einsatz.

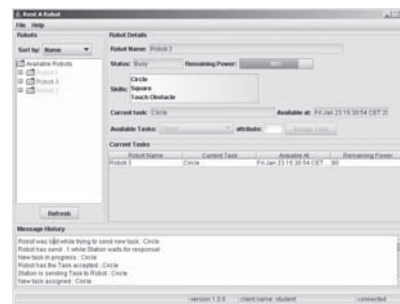


Abbildung 1: Das linke Bild zeigt das Rent-a-Robot Szenario während der Entwicklung. Ein schwarzer Kreis auf dem Boden markiert den Arbeitsbereich der Lego Mindstorms. Das rechte Bild ist ein Screenshot der Client-GUI, in dem 3 verfügbare Roboter entdeckt wurden.

Da Lego Mindstorms kein Wireless LAN unterstützen, kamen Laptop-Rechner als Zwischenstationen zum Einsatz. Als Konsequenz hieraus wurde der Middleware Server-Teil auf diesen Laptops implementiert, und nicht direkt auf den Robotern. Ein weiterer Grund hierfür sind die Beschränkungen durch LeJOS und die Roboter Hardware. Eine Erweiterung, die JavaSpace Clients direkt auf kleinen Geräten ermöglicht, wurde in einer Masterarbeit entwickelt, welche aus diesem Projekt hervorging [Co05, MC05].

## 2.2 Struktur und Gliederung des Kurses

Der Kurs ist in zwei Phasen aufgeteilt. Die erste Hälfte besteht aus Vorlesungen und Übungen, während in der zweiten das Rent-a-Robot Projekt umgesetzt wird. Der Kurs ist von vornherein mit dem Projekt als konkrete Zielvorgabe organisiert, so dass den Studierenden zu jedem Zeitpunkt klar ist, zu welchem Zweck und wie der gelernte Stoff im Projekt später eingesetzt werden kann. Middleware-Aspekte werden Schritt für Schritt eingeführt, so dass sie unmittelbar in den Übungen derselben Woche eingesetzt werden können und ihre Bedeutung für das spätere Projekt stets deutlich wird.

Am Ende der Vorlesungsphase hat das Auditorium in inkrementeller und angeleiteter Weise die komplette Middleware-Infrastruktur und -Architektur für das Projekt danach bereits verstanden und erstellt. Für die Übungen haben wir ein Rahmenprogramm-Skelett vorgegeben, welches in allen Übungen verwendet und schrittweise erweitert wird, so dass es als Basis für das Projekt dienen kann.

#	Vorlesung	Übung
1	Einführung - Motivation "Rent-a-Robot" Szenario - Hintergrund: Ubiquitous Computing - Middleware und Jini Übersicht - Agile Teammethoden, z.B. eXtreme Programming	Gegeben: Programm zum lookup und connect zum JavaSpace, entries lesen und schreiben Aufgaben: - Übersicht, Programm starten - entry erweitern mit Roboterfähigkeiten - das neue Feld auf Bildschirm ausgeben Neue Konzepte: Programm mit Jini, Entry
2	JavaSpaces (1) - Generelle Konzepte von JavaSpace - JavaSpace Interface - Nutzung von JavaSpaces im "Rent-a-Robot" Beispielcode	Aufgaben: - neuer entry für Roboterstatus - Statuswechsel simulieren mittels Tastatur - JavaSpace aktualisieren (take + write) Neue Konzepte: grundlegende Operationen mit entries
3	JavaSpaces (2) + Exkurs: verhaltensbasierte Roboterkontrollarchitekturen - Einführung in verhaltensbasierte Roboterkontrolle - Weitergehende Konzepte von JavaSpaces: Leasing / Distributed Event Notification / Distributed Transactions	Gegeben: Beispielprogramm mit notify() Aufgaben: - beliebige Nachricht an JavaSpace senden - auf Nachrichten warten mittels notify() - Nachrichten auf Bildschirm ausgeben Neue Konzepte: events und listeners
4	JavaSpaces (2) + Transaktionen und Persistenz - Wichtige Middleware Konzepte Transaktionen / Persistenz - Weitergehende Konzepte von JavaSpaces: Verteilte Transaktionen / Persistenz	
5	Java (optional) - Grundkonzepte von Java - Ein Crashkurs in Java	Aufgaben: - über Lösungen nachdenken bzgl. request/reply - request entry hinzufügen (mit unique ID) - warten und reagieren auf request entries - reply entry hinzufügen (passende ID) Neue Konzepte: Emulation von request/reply
6	Jini - Basiskonzepte von Jini - Discovery, Lookup und Join - Klassen für Discovery, Lookup und Join - Jini Discovery und Lookup im Rent-a-Robot Beispiel - Vergleich: UPnP Universal Plug and Play	
7	Java auf Lego Mindstorms - Einführung to Lego Mindstorms - LeJOS o Übersicht o Packages o Kommunikation o Einschränkungen	Gegeben: mehrere LeJOS Beispielprogramme Aufgaben: - Download von Programmen - kompilieren eigener Programme - Programm für einfache Aufgabe erstellen: o fahre Kreis mit bestimmtem Radius o fahre Rechteck - Umsetzung davon mittels Verhaltensframework Neue Konzepte: Mindstorms, LeJOS

Tabelle 2: Struktur und Inhalt des Kurses

### 3. Erfahrungen und Fazit

Der Kurs wurde bisher dreimal gehalten, zuerst mit einem klassischen modulbasierten Ansatz für das Projekt, später mit agilen Softwaretechniken XP (eXtreme Programming). Teilgenommen haben im ersten Durchgang fünf und danach immer 10 Studenten und Studentinnen. Im klassischen modulbasierten Ansatz wurde zunächst die Architektur des Gesamtsystems entwickelt und dann einzelne Module an einzelne Personen verteilt. Der Ansatz funktionierte bemerkenswert gut. Meistens war es möglich, Module passend zu Studieninteressen zu wählen, so dass die Motivation noch höher war. Zudem konnte auch außerhalb der regulären Vorlesungszeiten an den Teilen weitergearbeitet werden.

Im Falle von XP wurden die meisten der “best practices” (user stories, quick design, pair programming, unit und acceptance tests, continuous integration, effort estimation und monitoring, ...) im Projekt versucht umzusetzen. In unserem Fall konnte XP jedoch nicht alle positiven Effekt erzeugen, die in „regulären“ Vollzeit-Programmierprojekten in Erscheinung treten sollen. Die Programmiererfahrungen in der Gruppe waren sehr heterogen, was zu negativen Teameffekten führte. Zudem war das wöchentliche Arbeitsschema eher hinderlich. Trotzdem hatte den meisten Studenten und Studentinnen diese Methode gefallen und sie würden sie auch für das nächste Projekt wieder empfehlen.

Zusammengefasst haben wir gezeigt, wie Middleware, d.h. eine spezielle Variante Jini und JavaSpaces, erfolgreich in einem Masterkurs eingesetzt wurde. Der Einsatz von Robotern erhöhte die Motivation maßgeblich und half somit, sich mit den Problemen von dynamischen, verteilten Anwendungen und den Einsatzmöglichkeiten von Middleware vertraut zu machen. Eine angemessene Struktur der Vorlesungen und Übungen, welche die für das Projekt nötigen Konzepte einführt und vorbereitet, scheint notwendig zu sein, um eine geeignete Lernerfahrung bieten zu können, das Verstehen und Anwenden des Stoffes in sehr begrenzter Zeit zu garantieren und somit die Beschäftigung mit den Konzepten auch über die Vorlesung hinaus zu fördern.

**Danksagung:** Die praktischen Arbeiten dieses Kurses wurden inspiriert von einem studentischen Programmierkurs, veranstaltet von Prof. Jörg Kaiser an der Universität Ulm, in dem Kaffeemaschinen mittels Jini und JavaSpaces ferngesteuert wurden. Vielen Dank an Prof. Kaiser, dass er uns die erstellten Programme als Grundlage überlassen hat.

### Literaturverzeichnis

- [Co05] Couturier, S.: A JavaSpace Client for Small Devices. Masterarbeit, eingereicht an der FH Bonn-Rhein-Sieg.
- [FHA99] Freeman, E.; Hupfer, S.; Arnold, K.: JavaSpaces Principles, Patterns, and Practice, Pearson Education, ISBN 0201309556.
- [LeJOS] LeJOS: <http://lejos.sourceforge.net/>, 22.6.2005.
- [MC05] Mock, M.; Couturier, S.: Middleware Integration of Small Devices, zur Veröffentlichung angenommen auf der ETFA2005, 10th IEEE International Conference on Emerging Technologies and Factory Automation, Facolta' di Ingegneria, Catania, Italy, 2005.
- [MPT04] Müllerburg, M.; Petersen, U.; Theidig, G.: Mit Robotern spielend lernen: Das Projekt Roberta, ROBOTIK 2004, VDI-Berichte 1841, S. 393-400, 2004.