

## Aufgabe 2

Gruppe: TR-1

Mitglieder: Sebastian Kohl (SK), Mirko Schmidt (MS), Peter Matjeschk (PM), Carsten Lüdecke (CL), Michael Thiele (MT), Sven Fischer (SF), Ibrahim Osman (IO)

### 1) Begriffe

– KOP:

Abkürzung für die Kooperationsplattform, die im Rahmen des Praktikums entwickelt werden soll.

– Benutzer:

ALLE Studenten oder Professoren, sowie Administratoren, die sich im System anmelden oder damit arbeiten können.

Aufgenommen, da oftmals mit Benutzer nur Studenten, Tutoren oder Dozenten assoziiert werden, aber nicht alle möglichen Benutzer der Software.

– Struts:

Struts beschleunigen die Entwicklung von Java-Web-Applikationen durch standardisierte APIs. Diese übernehmen wichtige Kontrolle über die HTTP-Anfragen und die Präsentationsschicht.

Tutorial ist zu finden unter:

<http://www.java-webdesign.com/navigationaction.do?l=tutorials/struts-einfuehrung>

Struts werden im Projekt benutzt oder sollten benutzt werden, aber gearbeitet hat damit noch keiner der Mitarbeiter und daher wurde der Begriff in den Glossar aufgenommen.

– API:

API ist die Abkürzung für Application Programming Interface (englisch für Schnittstelle zur Anwendungsprogrammierung). Ein API ist die Schnittstelle, die ein Betriebssystem oder auch ein anderes Softwaresystem anderen Programmen zur Verfügung stellt.

Ein API stellt Routinen, Protokolle und Dienstprogramme für das Erstellen von Software dar. Ein gutes API erleichtert das Programmieren durch die Bereitstellung von Software-Blöcken, die der Programmierer im Idealfall nur noch mit eigenen Codefragmenten verbinden muss.

– Servlets:

Servlets sind Server-Applikationen, die ähnlich wie PHP und CGI auf dem Server ausgeführt werden und dynamisch Seiten erstellen, die im Browser angezeigt werden. Servlets sind dabei in Java realisiert und bieten deutlich mehr Komfort in der Programmierung und eine verbesserte und strikere Trennung von verschiedenen Programmiersprachen. Bei PHP ist der Code oft gekoppelt mit HTML und schwimmt zu einer einzigen Masse. Servlets lösen dieses Problem durch strikte Trennung per MVC-Konzept. Servlets benötigen einen Runtime-Container, wie z.B. Jakarta-Tomcat. Dieser ist zu finden unter:

<http://apache.mirrorplus.org/jakarta/tomcat-5/v5.5.9/bin/jakarta-tomcat-5.5.9.exe>

Offizielle Seite zum Tomcat:

<http://jakarta.apache.org/tomcat>

Tutorials zu finden unter:

<http://www.google.de/search?q=servlets+tutorial> (die ersten Hits sind stets die besten ;-)

Servlets sind für das Projekt der Kooperationsplattform das A & O, auch wenn keiner der Gruppen-Mitglieder damit gearbeitet hat oder wusste, dass diese überhaupt existieren. Daher wurden auch sie in den Glossar übernommen.

– Prüfungsnummern:

## Aufgabe 2

Gruppe: TR-1

Mitglieder: Sebastian Kohl (SK), Mirko Schmidt (MS), Peter Matjeschk (PM), Carsten Lüdecke (CL), Michael Thiele (MT), Sven Fischer (SF), Ibrahim Osman (IO)

Manche Lehrstühle verwenden Prüfungsnummern anstatt Matrikelnummern. Diese können entweder einer Person statisch zugeordnet sein, wie es die Matrikelnummern auch sind oder bei jeder Prüfung neu und individuell vergeben werden.

Prüfungsnummern gibt es nicht an allen Fakultäten, sollen aber eventuell in der Kooperationsplattform berücksichtigt werden, wenn es um Erstellung einer Klausur geht und sollten deshalb geklärt werden.

– Webservice:

Ein Webservice ist ein Dienst, der über Internetprotokolle arbeitet und über einen eindeutigen URI abrufbar ist. Webservices dienen unter anderem dem Datenaustausch zwischen Client und Server meist mit Hilfe von XML.

Entwicklungstools/Developerpack und Tutorials zu finden unter:

<http://java.sun.com/webservices/downloads/webservicespack.html>

Für den Webservice gilt ähnliches wie für Servlets und Struts, da es sehr wahrscheinlich ist, dass ein solcher zur Interkommunikation zwischen der KOP und den UebManager-Instanzen, bzw. deren Daten.

– URI:

Uniform Resource Identifier dienen dazu, eine physikalische oder abstrakten Ressource eindeutig über eine Zeichenfolge zu identifizieren.

Ein URI wird im Zusammenhang mit einem Webservice verwendet und eventuell nicht allen bekannt und wurde deshalb dem Glossar hinzugefügt.

– XML:

Extensible Markup Language ist eine Art standardisierte Scriptsprache zur Speicherung von Daten und Dokumenten in einer Baumstruktur, die sehr einfach von Maschinen und Menschen (ein-)zu lesen und zu verstehen ist.

XML wird sehr ausführlich benutzt werden müssen und ist genau wie Servlets etc. für viele etwas absolut neues.

## 2) Konzepte

Konzepte von „ATutor“ in Hinsicht auf Features, die diese Software leistet seien hier einmal analysiert:

Zu finden ist der ATutor unter <http://www.atutor.ca/atutor/index.php>. Es handelt sich dabei um ein *Open Source-Projekt*, wie es für KOP ebenfalls geplant ist. Es lässt sich leicht und schnell installieren, da es nur in einen Ordner zu schieben ist. Ein deutliches Pro, was auch in KOP wahrscheinlich gut umsetzbar ist dank Tomcat und WAR-Archiven. ATutor bietet mehrere Sprachen und Oberflächendesigns, was sicherlich praktisch ist, aber für KOP nicht Ziel sein soll und daher vernachlässigt werden kann. Studenten/Dozenten können im ATutor mehrere Kurse verwalten, bietet Arbeitsgruppen und ist dem UebManager bzw. KOP dahingehend überlegen. Ebenfalls ist eine Kontext-sensitive Hilfe vorhanden, was in KOP eventuell ebenfalls einbauen könnte, wobei fraglich ist inwiefern dies tatsächlich notwendig ist. ATutor bietet Kommunikationshilfen, wie etwa private Mails an andere Benutzer (sicherlich praktisch, bringt aber viele Probleme mit sich), Chatrooms, ein Forum und private Nachrichten an andere Benutzer. Ebenfalls vorhanden sind Link-Datenbanken (wie KOP sie in etwa bieten soll), Editoren für Inhalte, wie der UebManager sie schon zur Verfügung stellt, ein News-System (sollte man für KOP wirklich überlegen, da dies sowohl als Webseite, als auch als eMail-Nachricht wie etwa im Stundenplaner

## Aufgabe 2

Gruppe: TR-1

Mitglieder: Sebastian Kohl (SK), Mirko Schmidt (MS), Peter Matjeschk (PM), Carsten Lüdecke (CL), Michael Thiele (MT), Sven Fischer (SF), Ibrahim Osman (IO)

der Uni Leipzig, ein durchaus mächtiges Werkzeug ist) und Statistiken, sowie Umfragen. Alles in allem ist ATutor ein durchaus praktisches System, was aber wohl mehr für Schulen geeignet ist und bietet einige Features, die entweder überflüssig oder aber schon integriert sind. Dennoch bleiben das News-System und die Kommunikationsmöglichkeit zwischen einzelnen Benutzern, die interessante Features für KOP wären.

Konzepte des Programms „Schülermanager“ in Bezug auf KOP und den UebManager:

Eine weitere Ähnliche Software ist der „Schülermanager“, wobei der Name schon deutlich macht, dass er für den Übungs- und Vorlesungsbetrieb an einer Universität ungeeignet ist. Dennoch bietet er eine ähnliche Strukturierung, da sich Schüler, Lehrer und Rektor/Admin mit eigenen Rechten anmelden können. Auch diese Plattform bietet die Möglichkeit das Design anzupassen oder eine andere Sprache zu wählen. Als interessant aufzuführen ist die Möglichkeit kranke Lehrer, Ersatz und Ausfall online ersichtlich machen zu können, was ebenfalls im Hinterkopf behalten werden sollte im Hinblick auf Features für KOP. Allerdings beinhaltet der Schülermanager zwar praktische, aber für eine Universität unnötige Features wie Planung fester Stundenpläne durch den Rektor und Ansichten dieser.

Eine weitere Software, die die Recherche zu Tage gefördert hat ist „LEO“ (Leipzig Education Online) und ist im Einsatz zu finden unter [www.leo.uni-leipzig.de](http://www.leo.uni-leipzig.de)

Die Plattform ist sehr ähnlich dem UebManager, da sie Aus- und Einschreibung zu Veranstaltungen und Prüfungen bietet und Downloads von Folien und Lernmaterial bereithält. Allerdings bietet auch sie eine Kommunikation durch Nachrichten, Chat oder Forum zwischen den Studenten und Dozenten.

LEO bietet eine umfangreiche Studentenverwaltung, Stundenplan-, Terminkalender- und Adressbuchverwaltung. Ein witziges Feature für den Stunden ist sicherlich die Ansicht des Mensa-Speiseplans nach dem Logout aus dem System. Insgesamt dient LEO auch der Lernunterstützung und studienbezogene Informationsbeschaffung für BWL.

Grundlegende Konzepte, die die KOP in jedem Fall leisten soll:

- Navigation zwischen verschiedenen Instanzen des UebManagers und bietet eine Oberfläche für alle an
- Auflistung aller vorhanden Übungen
  - Einschreibung oder wieder Austragung aus dieser, gekoppelt mit eventueller Dead-Line
  - Wechsel des Übungstermins, falls ein Wechsel in eine andere Gruppe möglich ist (ohne Austragung und Neu-Einschreibung)
- Auflistung der Prüfungen / Klausuren
  - Dozent kann selber Klausuren und Prüfungen hinzufügen und die Rahmenbedingungen (Zulassungen, Dead-Lines, Info-Links und Mail-Adresse)
  - Einschreibung und eventuelle Abmeldung aus dieser, bis zu einer festen Dead-Line.
  - Überprüfung der Zulassung über Punkte aus dem UebManager und/oder Zulassungslisten in XML
    - \* Niemals Matrikelnummern und Namen gekoppelt aus dem System ausgeben!
    - \* Absicherung der Zulassungslisten und Verhinderung von unbefugtem Zugriff
    - \* Absicherung, dass Studenten keine Prüfung zweimal schreiben können, obwohl sie bereits bestanden haben.
- Listen-Einsicht für Professoren
  - Professoren können für Ihre Vorlesungen und Klausuren Listen ausdrucken
    - HTML-Listen mit Namen/Prüfungsnummern, Matrikelnummern, Studiengang (evt. Email) als Checkliste
    - HTML-Listen als Aushänge und öffentliche Referenzen nur mit Namen/Prüfungsnummern ODER Matrikelnummern/Prüfungsnummern
    - obige Listen als PDF-Dateien

## Aufgabe 2

Gruppe: TR-1

Mitglieder: Sebastian Kohl (SK), Mirko Schmidt (MS), Peter Matjeschk (PM), Carsten Lüdecke (CL), Michael Thiele (MT), Sven Fischer (SF), Ibrahim Osman (IO)

Optionale Konzepte, die in KOP eingebettet werden könnten:

- Newsletterfunktion per eMail zu gebuchten oder neuen Vorlesung und Klausuren
- News-Funktion als extra Webseite, die wie eine Art Log funktioniert
- Versand privater Nachrichten an einen Benutzer für Studenten oder an ganze Benutzergruppen für Dozenten, bzw. an Eingeschriebene Benutzer bei Klausuren oder Veranstaltungen.

### 3) Beschreibung der Rahmenapplikation

Die Projektarbeit ist zu integrieren in die Web-Applikation UebManager die der Organisation und Verwaltung des Übungsbetriebes von Lehrveranstaltungen dient.

Quelle:

Der UebManager ist im Rahmen des Softwarepraktikums 2003 entstanden und ist als OpenSource-Projekt unter folgender URL verfügbar: <http://sourceforge.net/projects/uebman>

Einsatzgebiete und Leistungsmerkmale:

Der UebManager ist ein webbasiertes Informationssystem zur Unterstützung des Übungsbetriebes einer Vorlesung am Lehrstuhl einer Universität. Er stellt dabei umfassende Funktionen bereit, die der Verwaltung einzelner Übungsgruppen, dem Bereitstellen von Übungsaufgaben sowie dem Korrigieren eingesandter Lösungen seitens der Studenten dienen.

Dabei findet die Bedienung ausschließlich über einen Web-Browser statt, der mit dem Übungsbetriebsverwaltungs-Server über HTTP kommuniziert, was eine 24h-Nutzung ermöglicht.

Der Server setzt wiederum auf dem Konzept der Java-Servlet-Technologie auf. Da das Projekt auf Java basiert, das aufgrund seiner Architektur zur sicheren sowie vollständig objektorientierten Programmierung zwingt, wird die Sicherheit beim Umgang mit sensiblen Daten gewährleistet. Desweiteren ist eine leichte Erweiterung des Servers möglich, da das Konzept auf modularem Design, ganz im Sinne von OOP basiert.

Außerdem unterstützt dieser Server die flexible Verwendung von HTML-Templates, in denen eingebettete Variablen durch dynamisch generierten Content ersetzt werden. Dadurch wird eine Trennung von HTML-Frontend und Java-Backend erreicht.

Beschreibung von Anwendungsfällen:

Die Anwendung gliedert sich in Funktionen, die ohne Authentifizierung zugänglich sind und in Bereiche, zu denen sich der Benutzer authentifiziert haben muss.

Zu erstgenannten Funktionen gehören u.a.:

- Student registriert eigenen Account
- registrierte Benutzer können sich vergessenes Passwort zuschicken lassen
- allgemeine Infos/Materialien zur Vorlesung
- öffentlicher Forumsbereich

## Aufgabe 2

Gruppe: TR-1

Mitglieder: Sebastian Kohl (SK), Mirko Schmidt (MS), Peter Matjeschk (PM), Carsten Lüdecke (CL), Michael Thiele (MT), Sven Fischer (SF), Ibrahim Osman (IO)

Funktionen, die nur registrierten Benutzern nach Auth. zugänglich sind:

Administrator:

- Konfiguration
- Datensicherung (evtl.)
- Archivierung aller persistenten Daten des Servers

Dozent:

- Erstellung Übungsgruppen (Anzahl, Raum, Übungsgruppenleiter)
- Materialien zur Vorlesung veröffentlichen
- Übungsaufgaben veröffentlichen und Abgabetermin festlegen
- Moderation und des Forums
- vollständiger Zugriff auf Punkteübersichten/Klausureinschreibungen

Tutor:

- Download gelöster Übungsaufgaben
- Upload korrigierter Übungsaufgaben inkl. Bewertung
- Zugriff auf Forum

Student:

- Einschreibung zu bestimmter Übungsgruppe
- Download von Übungsserien
- Upload der Lösung zur Serie
- eigenen Punktestand abfragen
- Klausureinschreibung
- Zugriff auf Forum

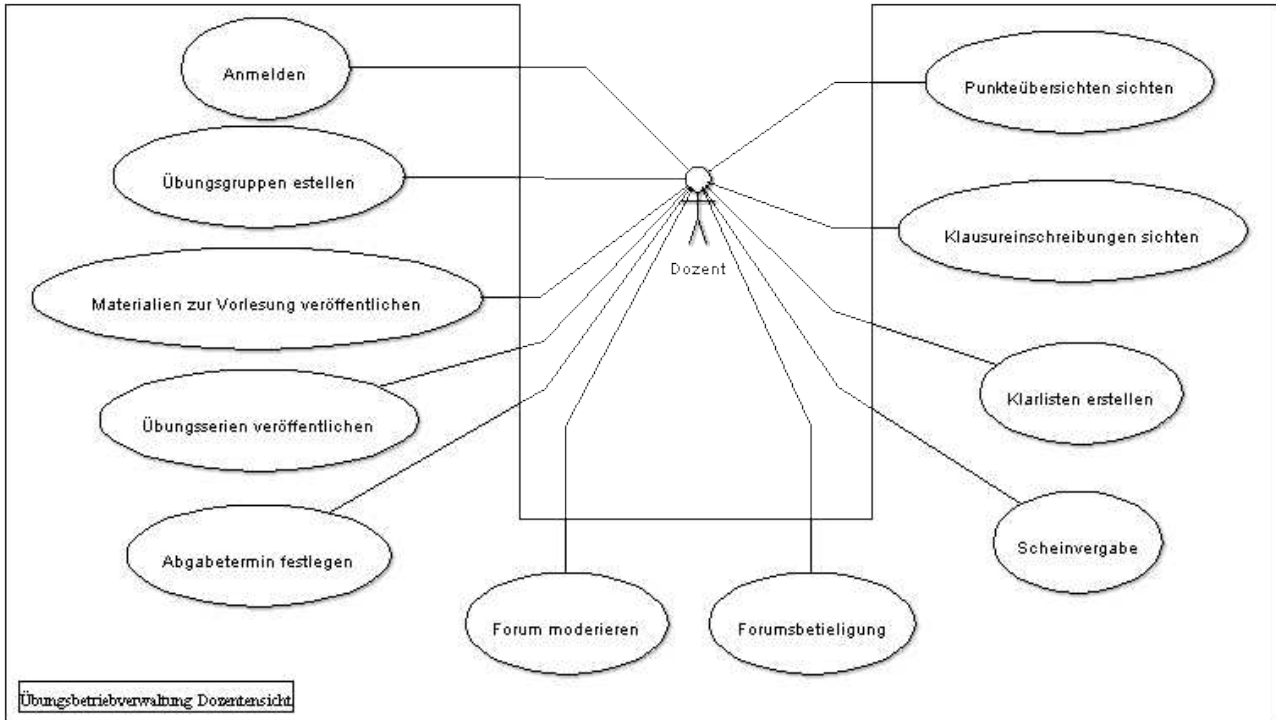
## Aufgabe 2

Gruppe: TR-1

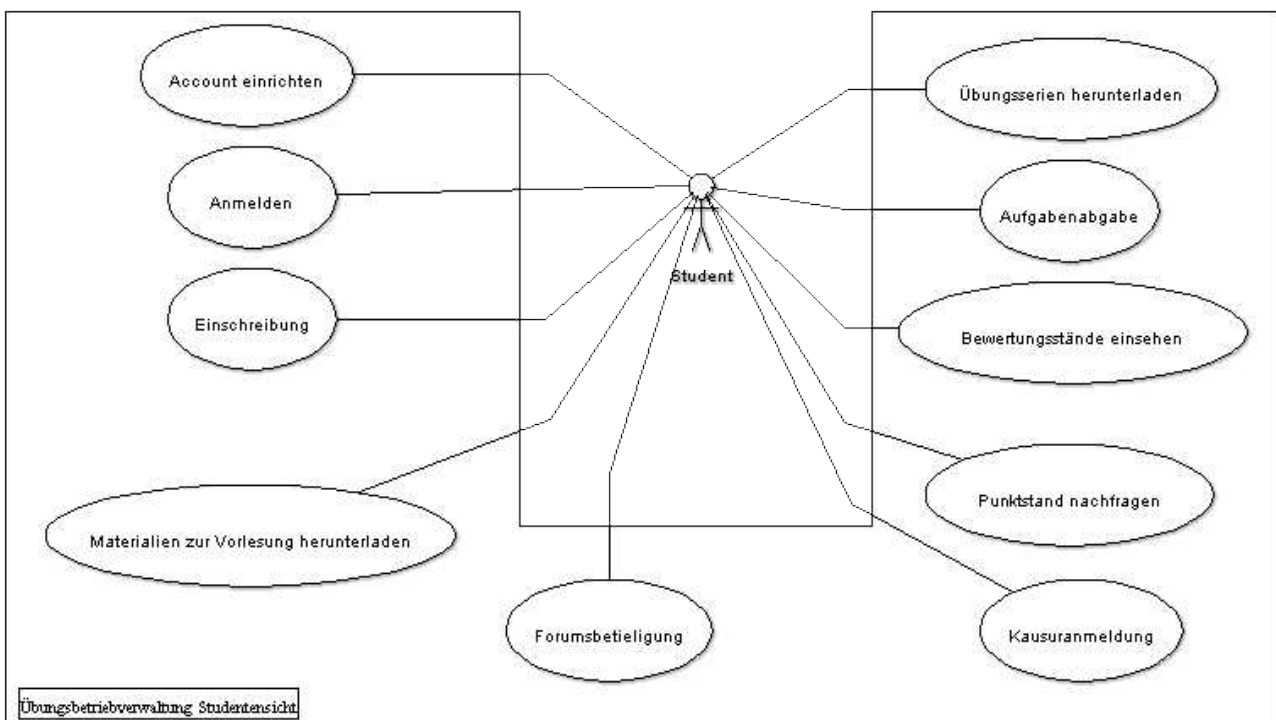
Mitglieder: Sebastian Kohl (SK), Mirko Schmidt (MS), Peter Matjeschk (PM), Carsten Lüdecke (CL), Michael Thiele (MT), Sven Fischer (SF), Ibrahim Osman (IO)

Im folgendem werden diese Funktionen noch mal in use case - Diagrammen dargestellt

Dozent:



Student:

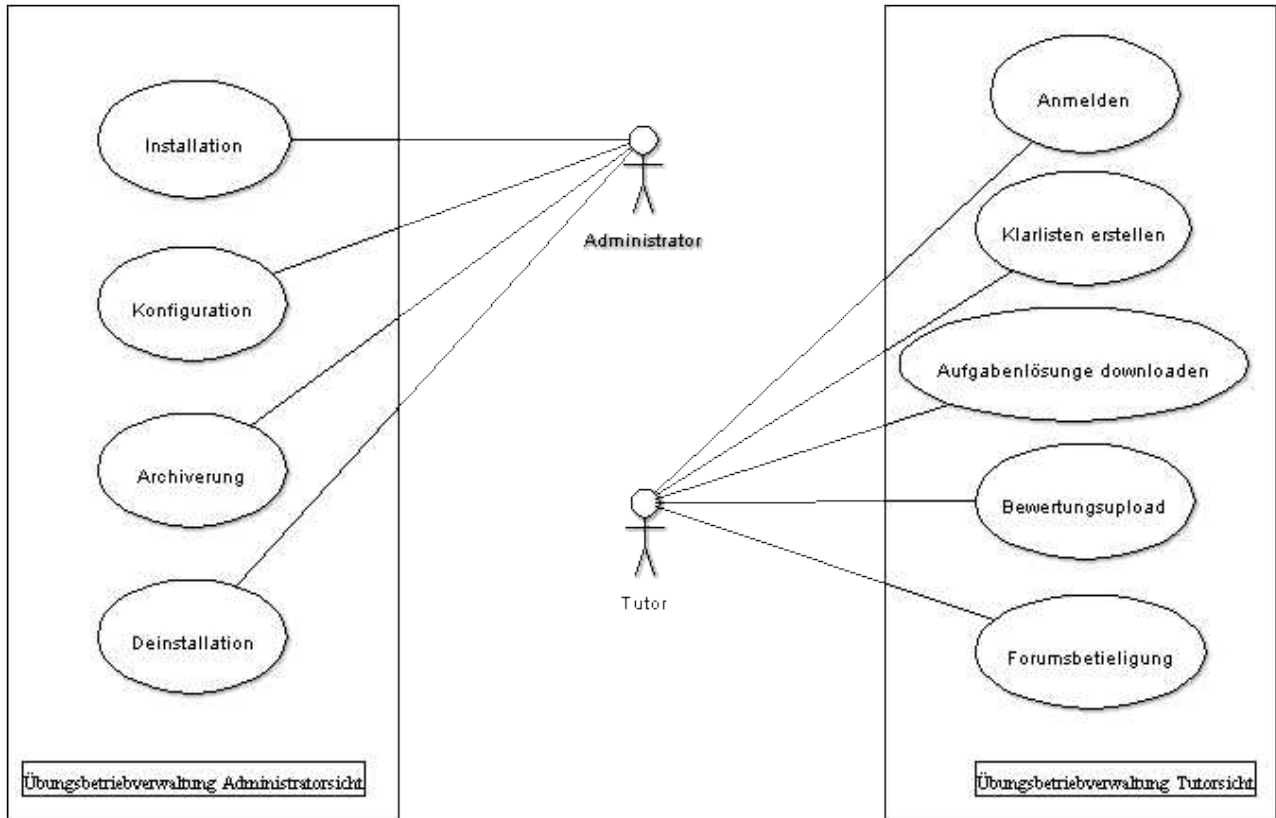


## Aufgabe 2

Gruppe: TR-1

Mitglieder: Sebastian Kohl (SK), Mirko Schmidt (MS), Peter Matjeschk (PM), Carsten Lüdecke (CL), Michael Thiele (MT), Sven Fischer (SF), Ibrahim Osman (IO)

Administrator und Tutor:



## Aufgabe 2

Gruppe: TR-1

Mitglieder: Sebastian Kohl (SK), Mirko Schmidt (MS), Peter Matjeschk (PM), Carsten Lüdecke (CL), Michael Thiele (MT), Sven Fischer (SF), Ibrahim Osman (IO)

Folgender Zustandsautomat beschreibt grob die Möglichkeiten, die einem Studenten geboten werden:

