

Modulhandbuch

Bachelor-Master-Studienprogramm

Informatik

*Fakultät für Elektrotechnik, Informatik und Mathematik der
Universität Paderborn*

*Deutschsprachiger Bachelor-Studiengang
Informatik (6 Sem.)*

*Deutschsprachiger Master-Studiengang
Informatik (4 Sem.)*

Paderborn, den 28. Mai 2009

Inhaltsverzeichnis

| | |
|---|-----------|
| Vorbemerkungen | 4 |
| Schema für Veranstaltungs- und Modulbeschreibungen | 4 |
| Prüfungsformen | 6 |
| I. Module im 1. Studienabschnitt des Bachelorstudiengangs | 7 |
| I.1 Gebiet Softwaretechnik und Informationssysteme | 7 |
| I.1.1 Programmiertechnik | 7 |
| I.1.2 Softwaretechnik | 11 |
| I.1.3 Datenbanken-Grundlagen | 14 |
| I.2 Gebiet Modelle und Algorithmen..... | 17 |
| I.2.1 Modellierung..... | 17 |
| I.2.2 Datenstrukturen und Algorithmen..... | 20 |
| I.2.3 Einführung in Berechenbarkeit, Komplexität und formale Sprachen | 23 |
| I.3 Gebiet Eingebettete Systeme und Systemsoftware..... | 26 |
| I.3.1 Grundlagen der technischen Informatik und Rechnerarchitektur..... | 26 |
| I.3.2 Konzepte und Methoden der Systemsoftware..... | 29 |
| I.4 Mathematik..... | 32 |
| I.4.1 Grundlagen der Mathematik | 32 |
| I.4.2 Stochastik..... | 34 |
| II. Module im 2. Studienabschnitt des Bachelorstudiengangs | 36 |
| II.1 Gebiet Softwaretechnik und Informationssysteme | 36 |
| II.1.1 Softwaretechnik und Informationssysteme | 36 |
| II.2 Gebiet Modelle und Algorithmen..... | 41 |
| II.2.1 Modelle und Algorithmen..... | 41 |
| II.3 Gebiet Eingebettete Systeme und Systemsoftware..... | 44 |
| II.3.1 Eingebettete Systeme und Systemsoftware..... | 44 |
| II.4 Gebiet Mensch-Maschine-Wechselwirkung..... | 48 |
| II.4.1 Mensch-Maschine-Wechselwirkung..... | 48 |
| II.5 Gebietsübergreifend | 52 |
| II.5.1 Informatik im Kontext | 52 |
| III. Module im Masterstudiengang | 54 |
| III.1 Gebiet Softwaretechnik und Informationssysteme | 54 |
| III.1.1 Modellbasierte Softwareentwicklung..... | 54 |
| III.1.2 Sprachen und Programmiermethoden | 57 |
| III.1.3 Datenbanken und Informationssysteme | 61 |
| III.1.4 Wissensbasierte Systeme | 64 |
| III.1.5 Analytische Methoden des Software Engineering | 66 |
| III.1.6 Konstruktive Methoden des Software Engineering..... | 69 |
| III.2 Gebiet Modelle und Algorithmen..... | 72 |
| III.2.1 Algorithmen I..... | 72 |
| III.2.2 Algorithmen II..... | 74 |
| III.2.3 Berechenbarkeit und Komplexität..... | 75 |
| III.2.4 Algorithmen in Rechnernetzen..... | 77 |
| III.2.5 Codes und Kryptographie | 79 |
| III.2.6 Optimierung | 81 |
| III.3 Gebiet Eingebettete Systeme und Systemsoftware | 83 |
| III.3.1 Verteilte Rechnersysteme..... | 83 |
| III.3.2 Systemsoftware | 87 |
| III.3.3 Rechnernetze..... | 90 |
| III.3.4 Eingebettete Systeme | 93 |
| III.3.5 HW/SW-Codesign | 97 |
| III.3.6 Eingebettete- und Echtzeitsysteme..... | 101 |
| III.4 Gebiet Mensch-Maschine-Wechselwirkung | 105 |
| III.4.1 Grafische Datenverarbeitung | 105 |
| III.4.2 Informatik und Gesellschaft..... | 109 |

| | | |
|---------|---|-----|
| III.4.3 | Konzepte digitaler Medien..... | 113 |
| III.4.4 | Computer gestützte kooperative Zusammenarbeit und Lernen..... | 115 |
| III.4.5 | Entwicklung von Benutzungsschnittstellen..... | 119 |
| III.4.6 | Mensch-Maschine-Wechselwirkung..... | 123 |
| III.4.7 | Mensch-Computer-Interaktion..... | 127 |

Vorbemerkungen

Die Modulbeschreibungen in diesem Katalog sollen

- Ziele, Inhalte und Zusammenhänge des Studienganges auf der Ebene von Modulen und Lehrveranstaltungen umfassend beschreiben,
- Studierenden nützliche, verbindliche Informationen für die Planung ihres Studiums geben,
- Lehrenden und anderen interessierten Personen einen tiefgehenden Einblick in die Ausgestaltung der Module des Studienganges geben.

Die Modulbeschreibungen sind nach einem vorgegebenen Schema weitgehend einheitlich strukturiert. Es wurde in einem intensiven Diskussionsprozess entwickelt, an dem sich Gremien und Kollegen des Faches, der Fakultät und des Paderborner Lehrerausbildungszentrum (PLAZ) beteiligt haben. Darin sind Vorgaben, Anregungen und Ideen aus vielfältigen Materialien zur Modularisierung eingeflossen. Wir haben besonderen Wert darauf gelegt, die Rolle des Moduls im Studiengang und die angestrebten Lernziele möglichst aussagekräftig zu beschreiben – neben den Angaben zu Inhalt und Organisation. Damit soll Lernenden und Lehrenden nicht nur gezeigt werden was vermittelt wird sondern auch warum das geschieht. Das Schema der Beschreibungen ist im Folgenden angegeben. Der erste Teil (bis einschließlich Abschnitt „Modus“) ist als über mehrere Jahre fest anzusehen, der Rest der Beschreibung kann für jede Instanz des Moduls ggf. angepasst werden.

Im Übrigen werden im Modulhandbuch auch die Prüfungsmodalitäten für die einzelnen Module, aber auch die generell vom Prüfungsausschuss genehmigten Prüfungsformen festgehalten.

Schema für Veranstaltungs- und Modulbeschreibungen

Veranstaltung: Name der Veranstaltung

Rolle der Veranstaltung im Studiengang

- Verortung im Studium (Zuordnung zu den Zielen des Studiengangs und Teilbereichen des Studienplans)
- Beschreibung wesentlicher Inhalte und Methoden der Veranstaltung und deren Bedeutung für ein Teilgebiet bzw. das Fach Informatik
- Zusammenhang mit anderen Veranstaltungen / Modulen
- Verwendbarkeit der Veranstaltung in anderen Studiengängen

Inhaltliche Gliederung der Veranstaltung

Inhaltliche Struktur und zeitliche Abfolge der Veranstaltung

Inhaltliche Verwendbarkeit

Anhand von geeigneten Beispielen sollen exemplarisch typische Anwendungsfelder für Inhalte und Methoden der Veranstaltung beschrieben werden.

Voraussetzungen und Vorkenntnisse

Angaben über notwendige formale Voraussetzungen und erforderliche Vorkenntnisse für die erfolgreiche Teilnahme an der Veranstaltung.

Lernziele der Veranstaltung

Benennung von Qualifikationszielen durch Verknüpfung von Inhalten (zentrale Wissensbereiche) und Fähigkeiten (zentrale Kompetenzbereiche): z.B. „*Studierende sollen in der Lage sein ...*“.

Vermittlung von Faktenwissen – Inhaltskompetenz

Relevante Wissensbereiche der Veranstaltung und ausgewählte Anwendungszusammenhänge.

Vermittlung von methodischem Wissen – Methodenkompetenz

In der Veranstaltung vermittelte fachwissenschaftliche Methoden, die die Studierenden an typischen Beispielen anwenden sollen.

Vermittlung von Transferkompetenz

Beispiele für die Anwendung der in der Veranstaltung erlernten Methoden in neuen Kontexten - ausgehend von Inhalten der Veranstaltung.

Vermittlung von normativ-bewertender Kompetenz

Kriterien und Beispiele für die Bewertung der in der Veranstaltung erlernten Inhalte und Methoden im Hinblick auf informatikbezogene Problemstellungen (z.B. Eignung und Grenzen der Methoden, die Qualität von Lösungen / Lösungsansätzen, die gesellschaftlichen und sozialen Implikationen von Lösungen / Lösungsansätze bzw. von Produkten).

Schlüsselqualifikationen

Erwarteter Beitrag der Veranstaltung zur Vermittlung von Schlüsselqualifikationen wie

- Kooperations- und Teamfähigkeit
- Präsentations- und Moderationskompetenz
- Fähigkeit zur Nutzung moderner IuK-Technologien
- Strategien des Wissenserwerbs
- interkulturelle Kompetenzen
- fachbezogenen Fremdsprachenkompetenzen

Modulzugehörigkeit

- Pflicht-, Wahlpflicht- oder Wahlveranstaltung
- Zugehörigkeit zu Modulen

Modus

- Leistungspunkte pro Modul (Workload)
- Leistungspunkte der Veranstaltung
- Umfang und Form des Angebots, z.B. 6 SWS (4V, 2Ü)
- Häufigkeit des Angebotes, z.B. jedes WS
- Dauer, z.B. 1 Semester

Methodische Umsetzung

Angaben zu Sozialformen und didaktisch-methodischen Arbeitsweisen in der Veranstaltung (z.B. Übungen in kleinen Gruppen, Projektlernen mit hohem Aktivitätsanteil der Studierenden, durchgehende Fallorientierung bei der Vermittlung von Inhalten, kleinere Anwendungsbeispiele als Ausgangspunkte zur Einführung in ein Teilthema, spätere Konkretisierung von theoretischen Konzepten an praktischen Beispielen, Selbststudienphasen mit LO's, guided tours in virtuellen Lernumgebungen, Dekonstruktion von Informatiksystemen mit Transfer, blended learning)

Organisationsformen / Medieneinsatz / Literaturangaben

- Angaben zu Organisationsformen nach denen die Veranstaltung durchgeführt wird (z. B. Vorlesung, Übung, Seminar, Praktikum, Projekt, Selbststudium, virtuelles Seminar)
- Hinweise auf erwartete Aktivitäten der Studierenden
- Eingesetzte Materialien z.B. Übungsblätter, Musterlösungen, Animationen....
- Eingesetzte Medien z.B. Hinweise auf IDE's, Softwaretools....
- Literaturhinweise zur Veranstaltung
- ggf. Hinweis auf ein (webbasiertes) Veranstaltungsskript

Prüfungsmodalitäten

- Angaben über Formen studienbegleitender Prüfungen in der Veranstaltung (z. B. schriftliche, mündliche Prüfungen, Vortrag, Hausarbeit, Projektarbeit, Praktikumstestat) die Aussagen über das Erreichen der Standards / Lernziele ermöglichen
- Angaben zur Kompensation einer veranstaltungsbezogenen Teilprüfung innerhalb einer Modulprüfung
- Angaben zur Notenermittlung

Modulverantwortliche(r)

Name des Betreuers / der Betreuerin des Moduls

Prüfungsformen

Die folgenden Prüfungselemente wurden vom Prüfungsausschuss Informatik in seiner Sitzung vom 22.03.07 pauschal für alle Lehrveranstaltungen genehmigt. Die folgenden Prüfungsformen (auch in Kombination) bedürfen demnach keiner gesonderten Genehmigung:

- Prozentual aufgeteilte Ermittlung der Note** mit vorher festgelegten Anteilen aus den Elementen Klausur, Mündliche Prüfung, Aktive Teilnahme an Übungen, Bearbeitung von schriftlichen Hausaufgaben (Teilprüfungen).
- Bestehen der Gesamtprüfung** nur nach Bestehen aller oder einiger vorher festgelegter Teilprüfungen aus den Elementen Klausur, Mündliche Prüfung, Aktive Teilnahme an Übungsgruppen, Bearbeiten von schriftlichen Hausaufgaben.
- Verbesserung der Note** (Bonussystem, dadurch kein Bestehen möglich) durch Aktive Teilnahme an Übungsgruppen oder Bearbeiten von schriftlichen Hausaufgaben.
- Verschlechterung der Note** (Malussystem, dadurch kein Nichtbestehen möglich) durch Nichtteilnahme an den Übungen oder Nichtbearbeiten von schriftlichen Hausaufgaben.

I. Module im 1. Studienabschnitt des Bachelorstudiengangs

I.1 Gebiet Softwaretechnik und Informationssysteme

I.1.1 Programmiertechnik

Rolle im Studiengang Informatik

Das Entwickeln von Software ist ein zentraler Tätigkeitsbereich in der Informatik. Software-Entwickler müssen Aufgaben analysieren und modellieren, Software-Strukturen entwerfen und diese in einer Programmiersprache implementieren können. Dieser Modul vermittelt einführende und grundlegende Kenntnisse und Fähigkeiten in der Programmierung. Zusammen mit den Modulen Modellierung und Softwaretechnik werden damit die Grundlagen für das Arbeitsgebiet Software-Entwicklung gelegt und praktisch eingeübt.

Dieser Modul soll die Teilnehmer befähigen,

- eine für die Software-Entwicklung relevante Programmiersprache anzuwenden (zurzeit Java),
- Grundbegriffe der objektorientierten Programmiermethodik einzusetzen,
- Grundkonzepte von Programmier- und Anwendungssprachen im allgemeinen zu verstehen,
- typische Eigenschaften nicht-imperativer Sprachen zu verstehen.

Insgesamt sollen sie damit in der Lage sein, neue Programmiersprachen und deren Anwendungen selbständig erlernen zu können. Im Informatikstudium bildet dieser Modul zusammen mit den Pflichtmodulen Modellierung und Softwaretechnik den Kern der Grundausbildung in Gebiet Softwaretechnik. Die Wahlmodule zu Sprachen und Programmiermethoden im Bachelor- und im Master-Studiengang vertiefen die Themen und Ziele dieses Moduls im Hinblick auf Sprachen, deren Übersetzung und Anwendung.

Inhaltliche Gliederung

Der Modul ist in drei Teile gegliedert: Grundlagen der Programmierung 1 (GP1, 1 Semester) und Grundlagen der Programmierung 2 (GP2, 1/2 Semester) leisten die grundlegende Ausbildung in einer Programmiersprache, Grundlagen der Programmiersprachen (GPS, 1/2 Semester) vermittelt die Konzepte von Programmiersprachen im allgemeinen.

Grundlagen der Programmierung 1 (GP1)

1. Grundbegriffe zu Programmen und ihrer Ausführung
2. Klassen, Objekte, Datentypen
3. Programm-und Datenstrukturen
4. Objektorientierte Abstraktion
5. Objektorientierte Bibliotheken

Grundlagen der Programmierung 2 (GP2)

1. Graphische Benutzungsschnittstellen
2. Ereignisbehandlung und Applets
3. Parallele Prozesse, Synchronisation, Monitore

Grundlagen der Programmiersprachen (GPS)

1. Syntaktische Strukturen
2. Gültigkeit von Definitionen
3. Lebensdauer von Variablen
4. Datentypen
5. Aufruf, Parameterübergabe
6. Funktionale Programmierung
7. Logische Programmierung

Inhaltliche Verwendbarkeit

Die in diesem Modul erworbenen Kenntnisse und Fähigkeiten können überall im Studium und Beruf eingesetzt werden, wo es gilt, Programme zu entwickeln. Dazu ist es nötig, nach den Übungen dieses Moduls noch weitere praktische Erfahrungen zu sammeln. Zusammen mit den Modulen Modellierung und Softwaretechnik vermittelt dieser Modul die Befähigung, im Studium und im Beruf Software zu entwickeln. Mit den Kenntnissen aus Grundlagen der Programmiersprachen sollen diese Fähigkeiten auch unabhängig von der jeweiligen Programmiersprache einsetzbar sein. Außerdem werden in Lehrveranstaltungen zu Sprachen und Programmiermethoden speziell die Themen aus diesem Modul vertieft und weiterentwickelt.

Voraussetzungen und Vorkenntnisse

Der Veranstaltungsteil Grundlagen der Programmierung setzt grundlegende Fähigkeit in der Rechnerbenutzung voraus. Programmierkenntnisse werden nicht erwartet, können aber den Einstieg erleichtern. Der Veranstaltungsteil Grundlagen der Programmiersprachen setzt voraus, dass eine Programmiersprache grundlegend erlernt wurde, wie sie z. B. im ersten Teil des Moduls vermittelt wird. Außerdem wird die Kenntnis des Kalküls kontextfreie Grammatiken, z. B. aus dem Modul Modellierung vorausgesetzt.

Lernziele

Die Studierenden sollen ...

Vermittlung von Faktenwissen

- die Konstrukte der Programmiersprache Java erlernen (GP),
- Grundkonzepte von Programmier- und Anwendungssprachen verstehen (GPS),
- typische Eigenschaften nicht-imperativer Sprachen verstehen (GPS)

Vermittlung von methodischem Wissen

- die gelernten Sprachkonstrukte sinnvoll und mit Verständnis anwenden (GP),
- objektorientierte Grundkonzepte verstehen und anwenden (GP),
- Software aus objektorientierten Bibliotheken wiederverwenden (GP),
- einfache Grammatiken, Typspezifikationen, funktionale und logische Programme entwickeln können (GPS)

Vermittlung von Transferkompetenz

- praktische Erfahrungen in der Programmentwicklung auf neue Aufgaben übertragen (GP, GPS)

- neue Programmier- und Anwendungssprachen selbständig erlernen (GP, GPS)

Vermittlung von normativ-bewertenden Kompetenzen

- die Eignung von Sprachen für spezielle Zwecke beurteilen (GPS)

Schlüsselqualifikationen

- Kooperations- und Teamfähigkeit in den Präsenzübungen
- Strategien des Wissenserwerbs:
- Kombination aus Vorlesung, Vor- und Nachbereitung am
- Vorlesungsmaterial, Präsenzübungen mit betreuter
- Gruppenarbeit, Hausaufgaben und Zentralübung.

Modulzugehörigkeit

Pflichtmodul im Gebiet Softwaretechnik

Modus

Leistungspunkte: 8+4+4 ECTS (GP1, GP2, GPS)

SWS: 4+2, 2+1, 2+1

Häufigkeit: jährlich; GP1 im WS; GP2, GPS nacheinander im SS

Dauer: 2 Semester

Methodische Umsetzung

In GP werden

- die Sprachkonstrukte an typischen Beispielen eingeführt, erläutert und in den Übungen praktisch erprobt,
- objektorientierte Methoden, überwiegend an der Benutzung von Bibliotheken gezeigt,
- in einigen Übungsstunden praktische Programmieraufgaben unter Anleitung an Rechnern bearbeitet.

In GPS werden

- Sprachkonstrukte, Spracheigenschaften und Programmierparadigmen im Vergleich und in Gegenüberstellung zu den in GP gelernten herausgearbeitet,
- funktionale und logische Sprachkonstrukte und Programmierkonzepte auch praktisch an Beispielen in SML und Prolog erarbeitet.

Organisationsform, Medieneinsatz, Literatureingaben

- Vorlesungen mit Folienpräsentation
- Präsenzübungen in Kleingruppen
- einige Übungsstunden unter Anleitung an Rechnern
- Erwartete Aktivitäten der Studierenden: Mitarbeit bei Präsenzübungen, Hausaufgaben, Vor- und Nacharbeit der Vorlesungen
- Übungsblätter, Musterlösungen werden in Zentralübungen vorgestellt
- Textbuch für GP: J. Bishop: Java lernen, Pearson Studium, 2. Aufl., 2001

- Web-basiertes Vorlesungsmaterial

Prüfungsmodalitäten

Klausur zu GP1

Praktischer Test zu GP1

Klausur zur GP2 und GPS

Modulverantwortlicher

Szwillus

I.1.2 Softwaretechnik

Rolle im Studiengang

Die Softwaretechnik befasst sich mit Konzepten, Sprachen, Methoden und Werkzeugen zur Erstellung und Wartung großer Softwaresysteme. Hierbei liegt ein wesentliches Augenmerk auf der Qualität der bearbeiteten Softwaresysteme. Hierzu gehört insbesondere die Sicherstellung funktionaler und nicht-funktionaler Anforderungen an das Softwaresystem, wobei abhängig vom Einsatzbereich ein unterschiedliches Gewicht auf den einzelnen Systemanforderungen liegt. Als Beispiele seien hier etwa Sicherheitsanforderungen in eingebetteten Systemen oder Benutzbarkeitsanforderungen in interaktiven Systemen genannt.

Die Veranstaltungen in diesem Modul führen zum einen in die objektorientierte Spezifikation von Softwaresystemen mittels der inzwischen als de-facto Standard geltenden Sprache UML ein. In einem anschließenden Praktikum wird die Entwicklung eines nicht trivialen Softwareprojekts im Team durchgeführt, um die bisher erworbenen Kenntnisse im Modul Programmiertechnik sowie in diesem Modul praktisch umzusetzen.

Inhaltliche Gliederung der Veranstaltung

Der Modul besteht aus zwei Pflichtveranstaltungen:

- Softwareentwurf (SE)
- Softwaretechnikpraktikum (SWTPRA)

Die Veranstaltungen sind wie folgt gegliedert

Softwareentwurf (SE):

In der Vorlesung werden Modellierungssprachen zur Beschreibung des statischen und dynamischen Aspekts von Softwaresystemen im Allgemeinen und von Benutzungsschnittstellen im Besonderen eingeführt. Hierzu gehört insbesondere die objektorientierte Modellierungssprache UML (Unified Modeling Language), die wiederum auf Diagrammsprachen wie Klassendiagrammen, Sequenzdiagrammen, Kollaborationsdiagrammen, Zustandsdiagrammen und Aktivitätsdiagrammen beruht. Die Vorlesung wird abgerundet mit methodischen Hinweisen zum Einsatz dieser Sprachen im Software-Entwicklungsprozess.

Softwaretechnikpraktikum (SWTPRA):

Das Softwaretechnikpraktikum ist ein 6-stündiges Praktikum inklusive einer Vorlesung über Projektmanagement. Eine komplexe Softwareentwicklungsaufgabe wird im Team von ca. 10 Studierenden unter Verwendung von UML und Java bearbeitet.

Schwerpunkte des Praktikums liegen in der Erfahrung einer teamorientierten Softwareentwicklung unter Benutzung marktüblicher Werkzeuge und Methoden (Rational Rose, Configuration and Version Management (CVS)). Zu Beginn des Praktikums arbeiten sich die Studierenden anhand eines bereits in Teilen vorliegenden Quelltexts, der im Praktikum zu erweitern ist, in die Aufgabe ein und müssen diesen re-dokumentieren. Die Erstellung von Meilensteinplänen, ein teilweise durch die Studierenden zu übernehmendes Projektmanagement sowie die Anfertigung von Kostenschätzungen und die Protokollierung des Aufwandes durch Stundenzettel sind wesentliche Bestandteile, um die Praxisnähe des Projekts sicherzustellen.

Inhaltliche Verwendbarkeit

Die in diesem Modul erworbenen Kenntnisse und Fähigkeiten bilden die wesentliche Grundlage für eine methodisch anspruchsvolle Durchführung und Leitung großer Softwareprojekte in der Industrie.

Voraussetzungen und Vorkenntnisse

Für die Veranstaltung Softwareentwurf sind grundlegende Kenntnisse in einer zum Softwareentwurf geeigneten Sprache (z.B. Java) Voraussetzung. Voraussetzung für die Veranstaltung Softwaretechnikpraktikum sind ein erfolgreich abgeschlossenes Modul Programmierertechnik (I.1.1) sowie die Veranstaltung Softwareentwurf dieses Moduls (I.1.2).

Lernziele

Die Studierenden sollen...

Vermittlung von Faktenwissen

- Techniken und Werkzeuge zur (objektorientierten) Modellierung, Dokumentation und Organisation großer Softwareprojekte erlernen

Vermittlung von methodischem Wissen

- Sprachen und Werkzeuge im Softwareentwicklungsprozess einsetzen können sowie den organisatorischen Ablauf eines Softwareprojekts von der Anforderungsdefinition bis zur Abgabe kennen lernen

Vermittlung von Transferkompetenz

Sprachen und Werkzeuge für ihren Einsatz in einem Softwareentwicklungsprozess lernen

Vermittlung von normativ-bewertender Kompetenz

- den praktischen Nutzen von planerisch durchdachten Projekten erkennen lernen
- die Probleme team-orientierter Softwareentwicklung kennen lernen sowie erste Ansätze zu ihrer Bewältigung

Schlüsselqualifikationen

- Kooperations- und Teamfähigkeit in den Präsenzübungen und Kleingruppen
- Strategien des Wissenserwerbs: Kombination aus Vorlesung, Vor- und Nachbereitung am Vorlesungsmaterial, Präsenzübungen mit betreuter Gruppenarbeit, Hausaufgaben.
- Präsentation technischer Sachverhalte (SWTPRA)
- Technisches Schreiben bei der Erstellung der Projektdokumentation (SWTPRA)

Modulzugehörigkeit

Pflichtmodul im Grundstudium

Modus

Leistungspunkte: 4 (SE) + 10 (SWTPRA) ECTS

SWS: 2+1 (SE), 1+4+1 (SWTPRA)

Häufigkeit: jede Veranstaltung einmal pro Jahr

Methodische Umsetzung

- Methoden und Techniken werden an typischen Beispielen eingeführt, erläutert und in den Übungen praktisch erprobt (SE)
- Durchführung eines Projekts, wie oben beschrieben, mit regelmäßigen Zwischenpräsentationen und Gruppensitzungen, die protokolliert werden (SWTPRA)

Organisationsform, Medieneinsatz, Literaturangaben

- Vorlesungen mit Folienpräsentation (SE)
- Präsenzübungen in Kleingruppen (SE)
Kleingruppenveranstaltung (SWTPRA)

Prüfungsmodalitäten

Klausur (SE), mündliche Präsentationen zu bestimmten Meilensteinen und schriftliche Abgabe des Quelltexts, des Entwurfs, der Dokumentation, Testprotokolle, Protokolle der Gruppensitzungen sowie einer lauffähigen Installation auf einer Webseite (SWTPRA). Zum Bestehen des Moduls muss jede Einzelleistung einzeln bestanden werden. Die Note des Gesamtmoduls ergibt sich dann aus der Note der Klausur für die Veranstaltung Softwareentwurf (SE).

Modulverantwortlicher

Schäfer

I.1.3 Datenbanken-Grundlagen

Rolle des Moduls im Studiengang BSc Informatik

Datenbanken spielen eine zentrale Rolle in Unternehmen, weil ein Großteil des „Wissen“ der Unternehmen als Daten in Datenbanken gespeichert wird. Für das Unternehmen ist es entscheidend, dass diese Daten korrekt, insbesondere konsistent sind und dass sie leicht erfragt bzw. erschlossen werden können. Deshalb kommt der Organisation großer Datenbestände in Datenbanken eine zentrale Rolle zu. Dabei geht es unter anderem darum, Datenbestände so zu organisieren, dass sie redundanzfrei aber trotzdem vollständig sind. Weiterhin sind die in Datenbanken abgelegten Datenbestände die wesentliche Datenquelle für eine Vielzahl von Anwendungsprogrammen, sie werden aber auch durch Anwendungsprogramme aktualisiert. Dieses Modul erschließt die Grundlagen für Datenbanken, die in nahezu allen Unternehmen in der Praxis eingesetzt werden.

Inhaltliche Gliederung der Veranstaltung

Der Modul besteht aus einer Lehrveranstaltung „Grundlagen von Datenbanksystemen“. Diese Lehrveranstaltung ist wie folgt gegliedert:

1. Entity-Relationship-Modell und konzeptueller Datenbankentwurf
2. Relationales Datenmodell
3. Relationale Algebra, Tupelkalkül, Domainkalkül und relationale Vollständigkeit
4. Datendefinitionssprache von SQL
5. Datenmanipulation in SQL
6. Die Anfragesprache von SQL
7. Sichten, Zugriffsrechte und View-Update-Problematik
8. Transaktionen in SQL
9. Eingebettetes SQL
10. Funktionale Abhängigkeiten, Schlüssel und andere Integritätsbedingungen
11. Datenbankschemaentwurf und Normalformen

Inhaltliche Verwendbarkeit

Die erworbenen Kenntnisse und Fähigkeiten werden in nahezu allen Unternehmen in der Praxis angewandt. Darüber hinaus werden sie in weitergehenden Veranstaltungen vertieft, insbesondere in der Veranstaltung Datenbanken und Informationssysteme 1 sowie in Spezialvorlesungen und Seminaren.

Voraussetzungen und Vorkenntnisse

Kenntnisse in der Programmierung werden in dem Umfang vorausgesetzt, wie sie in den Veranstaltungen Grundlagen der Programmierung 1 und 2 gelehrt werden.

Lernziele der Veranstaltung

Studierende lernen

Vermittlung von Faktenwissen

- Theorie und Konzepte relationaler Datenbanken kennen
- Grundkonzepte und relationaler Anfragesprachen kennen
- Grundlagen des Datenbankentwurfs kennen

Vermittlung von methodischem Wissen

in Kleingruppen-Präsenz-Übungen:

- Komplexe Anfragen an relationale Datenbanken korrekt zu formulieren
- ein Datenbankschema möglichst redundanzfrei zu entwerfen

in praktischen Übungen am Rechner:

- eigene SQL-Anfragen an existierende relationale Datenbanken stellen
- Programme zu schreiben, die Datenbestände aus Datenbanken lesen oder verändern
- eigene Datenbanken zu definieren und aufzubauen

Vermittlung von Transferkompetenz

- die erworbenen Kompetenzen und Fertigkeiten auf andere Datenquellen oder andere Datenbanksysteme zu übertragen
- Umgang mit Zugriffsrechten

Vermittlung von normativ-bewertenden Kompetenzen

- die Eignung und Grenzen des relationalen Datenmodells bewerten und einzuschätzen
- den Programmieraufwand für Datenbankabfragen und Datenbankprogrammierung einzuschätzen
- die Folgen einer Datenbankschema-Änderung zu erkennen und abzuschätzen
- die Risiken eines schlecht entworfenen Datenbankschemas zu bewerten.

Schlüsselqualifikationen

Studierende lernen in praktischen Übungen den Umgang mit SQL kennen, der wesentlichsten in der Industrie benutzten Datenbank-Anfragesprache. Sie erwerben durch eigene Rechnerübungen mit dieser Technologie zudem die notwendige Praxis, um sich eine Vielzahl darauf aufbauender Datenbank-Technologien zu erschließen zu können.

Modulzugehörigkeit

Pflichtmodul im Grundstudium.

Modus

- Leistungspunkte pro Modul (Workload) : 4
- Leistungspunkte der Veranstaltung : 4
- SWS: 2V + 1Ü
- Häufigkeit des Angebotes: jedes SS
- Dauer: 1 Semester

Methodische Umsetzung

- Die Grundlagen und Konzepte werden im Rahmen einer Vorlesung eingeführt.
- Die theoretischen Konzepte werden danach in Präsenzübungen in Kleingruppen vertieft.
Diese Methode wird insbesondere bei Kernkonzepten von Datenbanken (Datenmodell, Algebra und Kalkül, Integritätsbedingungen und Datenbankschemaentwurf) genutzt.
- Die praktischen Fertigkeiten werden erlernt anhand von Übungen am Rechner, bei denen ausgehend von Beispielen aus der Vorlesung eigene Datenbankabfragen zu stellen und eigene Datenbanken zu entwickeln sind. Diese Methode wird insbesondere genutzt beim Erlernen von SQL.

Organisationsformen / Medieneinsatz / Literaturangaben

- Vorlesung mit Lehrbüchern oder Skript bzw. Folienpräsentation und kleinen lauffähigen Beispielprogrammen am Rechner

- Übungen:
einerseits als Präsenzübungen in Kleingruppen mit Übungsblättern und Hausaufgaben,
andererseits praktische Übungen am Rechner.
- erwartete Aktivitäten der Studierenden: Mitarbeit bei Präsenzübungen, Hausaufgaben,
Entwicklung und Test eigener Datenbank-Anwendungssoftware am Rechner
- Standardlehrbücher über Datenbanken, Lehrmaterialien im Web

Prüfungsmodalitäten

Klausur

Modulverantwortliche(r)

Böttcher

I.2 Gebiet Modelle und Algorithmen

I.2.1 Modellierung

Rolle der Veranstaltung im Studiengang

Das Modellieren ist eine für das Fach Informatik typische Arbeitsmethode, die in allen Gebieten des Faches angewandt wird. Aufgaben, Probleme oder Strukturen werden untersucht und als Ganzes oder in Teilaspekten beschrieben, bevor sie durch den Entwurf von Software, Algorithmen, Daten und/oder Hardware gelöst bzw. implementiert werden. Mit der Modellierung eines Problems zeigt man, ob und wie es verstanden wurde. Damit ist sie Voraussetzung und Maßstab für die Lösung und sie liefert meist auch den Schlüssel für einen systematischen Entwurf. Als Ausdrucksmittel für die Modellierung steht ein breites Spektrum von Kalkülen und Notationen zur Verfügung. Sie sind spezifisch für unterschiedliche Arten von Problemen und Aufgaben. Deshalb werden in den verschiedenen Gebieten der Informatik unterschiedliche Modellierungsmethoden eingesetzt. In den entwurfsorientierten Gebieten (Softwaretechnik, Hardware-Entwurf) ist die Bedeutung der Modellierung und die Vielfalt der Methoden natürlich besonders stark ausgeprägt.

Inhaltliche Gliederung der Veranstaltung

1. Einführung

Begriffe Modell, Modellierung

2. Modellierung mit grundlegenden Kalkülen

Wertebereiche, Terme, Algebren

3. Logik

Aussagenlogik, Programmverifikation, Prädikatenlogik

4. Modellierung mit Graphen

Weg, Verbindung, Zuordnung, Abhängigkeiten, Abfolgen, Fluss

5. Modellierung von Strukturen

kontext-freie Grammatiken, Entity-Relationship-Modell

6. Modellierung von Abläufen

endliche Automaten, Petri-Netze

Inhaltliche Verwendbarkeit

Die erworbenen Kenntnisse und Fähigkeiten werden in vielen Vorlesungen angewandt und vertieft, z.B. Grammatiken in GdP, ER-Modell, in TSE, Logik in Wissensbasierten Systemen und in Berechenbarkeit, Petri-Netze in GTI, Graphen in DuA. Kenntnisse der grundlegenden Kalküle, Wertebereiche, Terme und der Logik werden bei jeder Art von formaler Beschreibung benötigt. Auch für die Berufstätigkeit der Informatiker ist das Modellieren eine typische Arbeitsmethode (siehe oben).

Voraussetzungen und Vorkenntnisse

Bereitschaft und Fähigkeit, formale Kalküle zu erlernen.

Lernziele der Veranstaltung

Studierende sollen

Vermittlung von Faktenwissen

- Grundkonzepte der vermittelten Kalküle erlernen,
- einen Überblick über grundlegende Modellierungsmethoden und -kalküle bekommen

Vermittlung von methodischem Wissen

- den konzeptionellen Kern der Kalküle beherrschen,
- die für die Methoden typischen Techniken erlernen,
- Kalküle an typischen Beispielen anwenden

Vermittlung von Transferkompetenz

in Übungen und Hausaufgaben neue Aufgaben mit den erlernten Kalkülen modellieren.

Vermittlung von normativ-bewertenden Kompetenzen

- an einer größeren Aufgabe die Eignung der Kalküle für die Modellierung von Teilaspekten
- untersuchen
- den praktischen Wert von präzisen Beschreibungen erkennen.

Schlüsselqualifikationen

- Kooperations- und Teamfähigkeit in den Präsenzübungen
- Strategien des Wissenserwerbs: Kombination aus Vorlesung, Vor- und nachbereitung am Vorlesungsmaterial, Präsenzübungen mit betreuter Gruppenarbeit, Hausaufgaben und Zentralübung

Modulzugehörigkeit

Pflichtmodul im Gebiet Modelle und Algorithmen.

Modus

- Leistungspunkte pro Modul (Workload) : 10
- SWS: 4V, 4Ü
- Häufigkeit des Angebotes: jedes WS
- Dauer: 1 Semester

Methodische Umsetzung

Zu jedem Modellierungskalkül wird

- mit einigen typischen kleinen Beispielproblemen motivierend hingeführt, der konzeptionelle Kern des Kalküls vorgestellt,
- Anwendungstechniken und Einsatzgebiete an Beispielen gezeigt und in den Übungen erprobt,
- auf weiterführende Aspekte des Kalküls, seine Rolle in Informatikgebieten und Vorlesungen sowie auf algorithmische Lösungsverfahren hier nur verwiesen.
- eine mittelgroße Modellierungsaufgabe (z.B. Getränkeautomat) bearbeitet. Am Ende der Vorlesung werden die Anwendungen vergleichend diskutiert.

Organisationsformen / Medieneinsatz / Literaturangaben

- Vorlesung mit Folienpräsentation

- Präsenzübungen in Kleingruppen
- erwartete Aktivitäten der Studierenden: Mitarbeit bei Präsenzübungen, Hausaufgaben

- Übungsblätter, Musterlösungen werden in Zentralübungen vorgestellt
- Web-basiertes Vorlesungsmaterial: WS 2001/2002:
U. Kastens: <http://www.uni-paderborn.de/cs/ag-kastens/model>

Prüfungsmodalitäten

Klausur

Modulverantwortliche(r)

Kastens

I.2.2 Datenstrukturen und Algorithmen

Rolle der Veranstaltung im BSc-Studiengang

Algorithmen bilden die Grundlage jeder Hardware und Software: Ein Schaltkreis setzt einen Algorithmus in Hardware um, ein Programm macht einen Algorithmus "für den Rechner verstehbar". Algorithmen spielen daher eine zentrale Rolle in der Informatik. Wesentliches Ziel des Algorithmenentwurfs ist die (Ressourcen-) Effizienz, d.h. die Entwicklung von Algorithmen, die ein gegebenes Problem möglichst schnell oder mit möglichst geringem Speicherbedarf lösen.

Untrennbar verbunden mit effizienten Algorithmen sind effiziente Datenstrukturen, also Methoden, große Datenmengen im Rechner so zu organisieren, dass Anfragen wie Suchen Einfügen, Löschen aber auch komplexere Anfragen effizient beantwortet werden können.

Die in dieser Veranstaltung vorgestellten Entwurfs- und Analysemethoden für effiziente Algorithmen und Datenstrukturen, sowie die grundlegenden Beispiele wie Sortierverfahren, dynamische Suchstrukturen und Graphenalgorithmen gehören zu den Grundlagen für Algorithmenentwicklung und Programmierung in weiten Bereichen der Informatik.

Inhaltliche Gliederung der Veranstaltung

1. Einführung
Rechenmodelle, Effizienzmaße, Beispiele
2. Sortierverfahren
Quicksort, Heapsort, Mergesort
3. Datenstrukturen
Verkettete Listen, Bäume, Graphen
Dynamische Suchstrukturen
Suchbäumen, Balancierung von Suchbäumen, Hashing
4. Entwurfs- und Analyseverfahren
Rekursion und das Mastertheorem, Teile-und-Herrsche, Dynamische Programmierung, Backtracking, Branch & Bound, Greedy Algorithmen
5. Graphenalgorithmen
Kürzeste Wege, Minimale Spannbäume, Flussprobleme

Inhaltliche Verwendbarkeit

Die erworbenen Kenntnisse und Fähigkeiten werden in vielen Gebieten angewandt und vertieft, z.B. in Betriebssystemen und Informationssystemen, Hard- und Softwareentwurf, Computergraphik, Operations Research und natürlich in den weiterführenden Vorlesungen über Algorithmen, Netzwerke, Optimierung und Parallelität. Auch für die Berufstätigkeit der Informatiker ist der Algorithmenentwurf eine typische Arbeitsmethode.

Voraussetzungen und Vorkenntnisse

Bereitschaft und Fähigkeit, den kreativen Prozess des Algorithmenentwurfs und die Effizienzanalyse u.a. mit mathematischen Methoden zu erlernen.

Lernziele der Veranstaltung

Vermittlung von Faktenwissen

- Entwurfsmethoden für effiziente Datenstrukturen und Algorithmen.
- Effiziente Datenstrukturen und Algorithmen für ausgewählte grundlegende Probleme
- Methoden zum Korrektheitsbeweis und zur Effizienzanalyse von Algorithmen und Datenstrukturen

Vermittlung von methodischem Wissen

- Selbstständiges, kreatives Entwickeln von Algorithmen und Datenstrukturen ("Wie gestalte ich den kreativen Prozess vom algorithmischen Problem zum effizienten Algorithmus?")
- Einsetzen mathematischer Methoden zum Korrektheitsbeweis und zur Effizienzanalyse
- Verständnis für Wechselwirkung zwischen Algorithmus und Datenstruktur
- Einschätzen der Qualität von Algorithmen und algorithmischen Ansätzen unter Effizienzaspekten
- Selbstständiges Aneignen von neuen Algorithmen, Datenstrukturen und algorithmischen Ideen und Analysen

Vermittlung von Transferkompetenz

In Übungen und Hausaufgaben werden Entwurf und Analyse von Algorithmen an ausgewählten Beispielen eingeübt.

Vermittlung von normativ-bewertenden Kompetenzen

- Einschätzen der Qualität von Algorithmen und algorithmischen Ansätzen unter Effizienzaspekten
- Einschätzen von Problemen in Hinblick auf ihre algorithmische Komplexität

Methodische Umsetzung

Für Probleme wie z.B. Sortieren oder dynamische Suchstrukturen werden sehr unterschiedliche algorithmische Methoden vorgestellt und verglichen. Dabei werden Anforderungen an benötigte Datenstrukturen herausgearbeitet, und auch hier unterschiedliche Verfahren entwickelt und analysiert (z.B. für Suchstrukturen, Prioritätswarteschlangen oder Union-Find Strukturen). Anhand solcher Verfahren werden die mathematischen Methoden zur Korrektheits- und Effizienzanalyse vermittelt.

Modulzugehörigkeit

Pflichtmodul im Gebiet Modelle und Algorithmen.

Modus

- Leistungspunkte pro Modul (Workload) : 8
- SWS (4V, 2Ü)
- Häufigkeit des Angebotes: jedes SS
- Dauer (1 Semester)

Formen der Vermittlung / Medieneinsatz / Literaturangaben

- Vorlesung mit Beamer und Tafelanschrieb
- Übungen in Kleingruppen
- erwartete Aktivitäten der Studierenden: Mitarbeit bei Präsenzübungen, Hausaufgaben

- Übungsblätter, Musterlösungen werden in Zentralübungen vorgestellt
- webbasiertes Vorlesungsmaterial: SS 2001/2002: Friedhelm Meyer auf der Heide:
<http://www.uni-paderborn.de/fachbereich/AG/agmadh/vorl/DaStrAlg01/dua.html>

Schlüsselqualifikationen

- Kooperations- und Teamfähigkeit in den Präsenzübungen
- Strategien des Wissenserwerbs: Kombination aus Vorlesung, Vor- und nachbereitung am Vorlesungsmaterial, Präsenzübungen mit betreuter Gruppenarbeit, Hausaufgaben und Zentralübung
- Kreatives Problemlösen am Beispiel der Entwicklung effizienter Algorithmen.

Prüfungsmodalitäten

Klausur

Modulverantwortliche(r)

Meyer auf der Heide

I.2.3 Einführung in Berechenbarkeit, Komplexität und formale Sprachen

Rolle der Veranstaltung im Studiengang

Die Modellierung und Analyse von Problemen sowie die Beurteilung gefundener Lösungen sind grundlegende Bestandteile der Informatik. Wesentliches Hilfsmittel zur Modellierung von Problemen sind formale Sprachen und Grammatiken. Durch formale Sprachen und Grammatiken beschriebene Probleme sind einer mathematischen Analyse zugänglich. Die Analyse von Problemen basiert auf der Unterscheidung unterschiedlicher Typen von Problemen sowie der Möglichkeit die Schwierigkeitsgrade von Problemen vergleichen zu können. Die größte und wichtigste Unterscheidung von Problemtypen ist die Unterscheidung in Probleme die auf einem Computer prinzipiell lösbar sind und solchen Problemen, die prinzipiell auf keinem Computer gelöst werden können. Die Klasse der prinzipiell lösbaren Probleme wird dann noch weiter gemäß unterschiedlicher Komplexitätsmaße wie Zeit- und Speicherbedarf unterteilt werden. Grundlage dieser Klassifikation müssen immer Rechenmodelle sein, die zugleich mathematisch präzise wie realistisch sein müssen. Die in dieser Veranstaltung beschriebenen Modellierungs- und Klassifikationskonzepte finden Anwendung in beiden Schwerpunkten der Paderborner Informatik. Die Modellierungskonzepte findet weite Anwendung insbesondere in der Software-Entwicklung. Die Klassifikationskonzepte bilden die Grundlage für den Schwerpunkt der Algorithmik.

Inhaltliche Gliederung der Veranstaltung

1. Einführung
Sprachen, Rechenmodelle, Grammatiken, Simulationen
2. Berechenbarkeit:
Entscheidbare, unentscheidbare Sprachen, Diagonalisierung, Halteproblem, Reduktionen, Beispiele
3. Zeitkomplexität :
Laufzeiten, Klassen P und NP, polynomielle Reduktionen, NP-Vollständigkeit, SAT, Satz von Cook-Levin, Beispiele
4. Approximationsalgorithmen und Heuristiken
Approximationsalgorithmen, Approximationsgüte, Beispiele, Backtracking, Branch-and-Bound, Lokale Verbesserung
5. Formale Sprachen und Grammatiken
Grammatiktypen, Zusammenhang mit Entscheidbarkeit, reguläre und kontextfreie Sprachen, endliche Automaten, Kellerautomaten, Pumping Lemma

Inhaltliche Verwendbarkeit

Die erworbenen Kenntnisse und Fähigkeiten werden in vielen Gebieten angewandt und vertieft, z.B. in Hard- und Softwareentwurf, Computergraphik, Betriebssysteme und Informationssysteme, in den weiterführenden Vorlesungen über Algorithmen, Komplexitätstheorie, Kryptographie, Optimierung und Parallelität. Die Modellierungskonzepte der formalen Sprachen und Grammatiken ist für die Berufstätigkeit jedes Informatikers unerlässlich. Die Konzepte der Entscheidbarkeit sind als Hintergrundwissen auch in der Praxis wesentlich. Die Konzepte aus dem Bereich der Algorithmen und Komplexität finden bei jedem Informatiker, der im Bereich des Algorithmenentwurfs arbeitet, Anwendung

Voraussetzungen und Vorkenntnisse

Vorlesungen Modellierung sowie Datenstrukturen und Algorithmen, Bereitschaft und Fähigkeit intuitive Konzepte formal zu fassen und diese dann auf konkrete Probleme anzuwenden.

Lernziele der Veranstaltung

Vermittlung von Faktenwissen

- Konzepte und Methoden der Berechenbarkeitstheorie.
- Formale Sprachen, Grammatiken und die zugehörigen Rechenmodelle
- Konzepte und Methoden der Komplexitätstheorie und der Algorithmik

Vermittlung von methodischem Wissen

- Selbstständige Analyse und Klassifikation von Problemen, Entwickeln von Hypothesen und daran anschließende Verifikation oder Falsifikation und Neuformulierung der Hypothesen
- Einsetzen mathematischer Methoden zur Analyse und Klassifikation.
- Verständnis für die grundlegende Struktur von Komplexitätsaussagen
- Einschätzen der Komplexität von Problemen anhand der in der Vorlesung vorgestellten Komplexitätsklassen

Vermittlung von Transferkompetenz

In Übungen und Hausaufgaben werden Modellierung, Analyse und Klassifikation von Problemen an ausgewählten Beispielen eingeübt.

Vermittlung von normativ-bewertenden Kompetenzen

- Einschätzen der Komplexität von Problemen
- Einschätzen von Lösungen im Hinblick auf praktische Verwertbarkeit

Methodische Umsetzung

Für verschiedene Probleme wird erläutert wie

- sie mit Hilfe von Grammatiken und formale Sprachen modelliert werden können
- sie analysiert und in die verschiedenen Komplexitätsklassen der Vorlesung eingeordnet werden können
- wie Bezüge und Vergleiche zu anderen Problemen hergestellt werden können
- die Klassifikation Lösungsansätze vorgibt bzw. einschränkt
- bei schwer zu lösenden Problemen häufig doch noch praxisgerechte Lösungsansätze gefunden werden können.

Modulzugehörigkeit

Pflichtmodul im Gebiet Modelle und Algorithmen.

Modus

- Leistungspunkte pro Modul (Workload) : 8
- Leistungspunkte der Veranstaltung: 8
- SWS (4V, 2Ü)
- Häufigkeit des Angebotes: jedes WS
- Dauer (1 Semester)

Formen der Vermittlung / Medieneinsatz / Literaturangaben

- Vorlesung mit Beamer/Folien und Tafelanschrieb
- Übungen in Kleingruppen
- erwartete Aktivitäten der Studierenden: Mitarbeit bei Präsenzübungen, Hausaufgaben
- Übungsblätter, Musterlösungen werden in Zentralübungen vorgestellt
- webbasiertes Vorlesungsmaterial: WS 02/03, SS03: Johannes Blömer:
http://webserv.uni-paderborn.de/cs/ag-bloemer/lehre/bfs_WS2002
http://webserv.uni-paderborn.de/cs/ag-bloemer/lehre/auk_SS2003

Schlüsselqualifikationen

- Kooperations- und Teamfähigkeit in den Präsenzübungen
- Strategien des Wissenserwerbs: Kombination aus Vorlesung, Vor- und nachbereitung am Vorlesungsmaterial, Präsenzübungen mit betreuter Gruppenarbeit, Hausaufgaben und Zentralübung
- Kreatives Problemlösen am Beispiel der eigenständigen Modellierung, Analyse und Klassifikation von Problemen.

Prüfungsmodalitäten

Klausur

Modulverantwortliche(r)

Blömer

I.3 Gebiet Eingebettete Systeme und Systemsoftware

I.3.1 Grundlagen der technischen Informatik und Rechnerarchitektur

Rolle des Moduls im Studiengang

Informatik-Studierende jeglicher Ausprägung sollten ein Grundverständnis über die Grundlagen der technischen Informatik und die Grundprinzipien der Wirkungsweise von Digitalrechnern haben. Die Studierenden, die sich später in technischer Informatik aber auch in Gebieten wie Compilerbau oder Systemnahen Softwarebereichen vertiefen wollen, benötigen diese Grundkenntnisse ganz offensichtlich. Aber auch für andere Schwerpunkte der Informatik bildet die technische Informatik mit ihren Modellierungs- und Lösungstechniken (beispielsweise Boolesche Algebra, Automatentheorie, Optimierungsverfahren in der Booleschen Algebra und der Automatentheorie, Arithmetik Algorithmen, Prinzip des Caching, Parallelität) eine wesentliche Grundlage. Für die Entwicklung effizienter Software ist darüber hinaus ein Grundverständnis der Wirkungsweise moderner Digitalrechner unumgänglich.

Dieses Modul ist ein Pflichtmodul für Studierende der Informatik und der Ingenieurinformatik. Es besteht aus zwei Veranstaltungen „Grundlagen der Technischen Informatik“ und „Grundlagen der Rechnerarchitektur“. In der ersten Veranstaltung wird besonderer Wert auf die Modellierungstechniken der technischen Informatik gelegt. Aus diesen Modellen werden dann die Methoden des Entwurfs digitaler Systeme abgeleitet. Somit ist diese Veranstaltung in die Konzeption des Informatikstudiums, welches ganz wesentlich auf Modellbildung abgestützt wird, nahtlos eingebettet. Die zweite Veranstaltung greift diesen auf Modellbildung basierenden Ansatz auf, um schrittweise die Wirkungsprinzipien moderner Universalprozessoren zu entwickeln. Phänomenologische Aspekte (Beschreibung realer Prozessorarchitekturen) werden zwar auch behandelt, dienen aber nur zur Illustration der Prinzipien.

Inhaltliche Gliederung der Veranstaltung

Das Modul besteht aus den Lehrveranstaltungen Grundlagen der Technischen Informatik und Grundlagen der Rechnerarchitektur. Die Veranstaltung Grundlagen der Technischen Informatik vermittelt einen Einblick über die Modellierung kombinatorischer Schaltungen (Boolesche Algebra) und sequentieller Schaltwerke (endliche transformierende Automaten). In beiden Fällen werden Optimierungsverfahren behandelt. Aufbauend auf diesen Modellen werden Grundstrukturen digitaler Schaltungen dargestellt. Zusätzlich findet eine knappe Einführung in die zugrunde liegende Halbleitertechnologie und in Techniken der Anbindung kontinuierlicher Systeme statt. Alternative Zahldarstellungen und die darauf basierenden Arithmetikalgorithmen werden eingeführt. Die prinzipielle Vorgehensweise beim Entwurf digitaler Systeme bildet den Abschluss dieser Veranstaltung. Sie wird durch ein, weitgehend auf Simulation mittels VHDL abgestütztes Praktikum abgerundet. Die Veranstaltung Grundlagen der Rechnerarchitektur vermittelt zunächst ein Grundverständnis über die Wirkungsweise eines v. Neumann-Rechners. Dies geschieht auf der Basis einer vereinfachten MIPS-Architektur. Das so eingeführte Grundprinzip wird nun schrittweise verfeinert, bis die Prinzipien moderner Rechnerarchitekturen abgedeckt sind. Dabei werden die Aspekte Informationsspeicherung (Speicherhierarchie), Zugriff auf Information (Adressierungstechniken), Informationstransport (Bus-Systeme), Zugriff auf entfernte Information (E/A, Interrupts), parallele Informationsverarbeitung (Pipelining) angesprochen. Die Konzepte werden anhand aktueller Prozessorarchitekturen (Pentium als CISC-Beispiel, PowerPC als RISC-Ansatz) illustriert.

Inhaltliche Verwendbarkeit

Die Prinzipien der Technischen Informatik finden sich in weiten Bereichen der Informatik wieder. Somit sind die Kenntnisse, die aus diesem Pflichtmodul gewonnen werden, breit einsetzbar. Für Studierende, die im Gebiet der technischen Informatik, insbesondere den eingebetteten Systemen vertiefen wollen, wird ein unumgängliches Rüstzeug vermittelt. Aber auch für das Gebiet der systemnahen Softwareentwicklung werden wichtige Grundlagen gelegt. Entwickler von Anwendungssoftware und von Verfahren der Softwareentwicklung wird das Verständnis der zugrunde liegenden Prozessorarchitekturen vermittelt, ein Verständnis, das zur Entwicklung effizienter Software und von Entwurfsverfahren für effiziente Software unumgänglich ist.

Vorraussetzungen und Vorkenntnisse

Als Vorkenntnisse werden die Inhalte des Moduls Modellierung vorausgesetzt. Ansonsten wird nur von den mathematischen Grundkenntnissen ausgegangen, die durch die allgemeine Hochschulreife gegeben sein sollte.

Prüfbare Standards / Lernziele der Veranstaltung

Die Studierenden sollen das Verständnis der spezifischen Eigenschaften von digitalen Systemen, insbesondere von Prozessoren erlangen sowie die elementaren Konzepte zum Entwurf derartiger Systeme kennen lernen. Die Studierenden sollen die Methoden zur Modellierung solcher Systeme und den darauf aufbauenden Optimierungsverfahren verstehen. Sie sollen in der Lage sein, die spezifischen Restriktionen, die sich durch die physikalischen Gesetze technischer Systeme ergeben, einzuschätzen und lernen diese gezielt in den Entwurfsprozess einzubeziehen. Schließlich sollen sie verstehen, wie sich die Restriktionen, die sich aus der Digitaltechnik und der spezifischen Rechnerarchitekturen ergeben, auf höhere Abstraktionsebenen, insbesondere der Softwaretechnik auswirken.

Vermittlung von Faktenwissen – Inhaltskompetenz

- Modellierung digitaltechnischer Systemkomponenten
- Entwurfstechniken für digitale Systeme
- Grundprinzipien der Prozessorarchitektur
- Verständnis über die Interaktion Software/Hardware

Vermittlung von methodischem Wissen – Methodenkompetenz

- Methoden zur Modellierung kombinatorischer Systeme
- Methoden zur Modellierung sequentieller Systeme
- Methoden zur Optimierung komplexer Systeme
- Methoden zur Parallelarbeit
- Methoden für den Entwurf von digitalen Systemen

Vermittlung von Transferkompetenz

Übertragung der globalen Strategien auf vorgegebene Einzelsituationen, zum Beispiel im Rahmen von Übungsaufgaben

Vermittlung von normativ-bewertender Kompetenz

- Strategien zur Behandlung unterschiedlicher Strategien ausarbeiten
- Den praktischen Wert der Konzepte und Methoden der Prozessorarchitekturen erkennen

Schlüsselqualifikationen

Durch den Übungsbetrieb in kleinen Gruppen wird die Kooperations- und Teamfähigkeit gefördert.

Modulzugehörigkeit

Pflichtmodul

Modus

- Leistungspunkte pro Modul (Workload): 10
- SWS (GTI: 2V+1Ü+1 Praktikum, GRA: 2V+2Ü)
- Häufigkeit des Angebotes: GTI: jedes SS, GRA: jedes WS
- Dauer (2 Semester)

Methodische Umsetzung

Übungen in kleinen Gruppen fördern die praktische Anwendung der vorgestellten Methoden auf ausgewählte Beispiele. Insbesondere müssen Parameter und Strategien an die konkrete Situation angepasst werden. Die Aufgaben werden in Gruppen von 3 Studierenden bearbeitet, wodurch die Teamfähigkeit gefördert wird.

Organisationsformen / Medieneinsatz / Literaturangaben

- Folien-orientierte Vorlesung, Tafelanschrieb bei Beispielen, zusätzlichen Erläuterungen und zu potenzierenden Sachverhalten
- Wöchentliche Übungen in kleinen Gruppen. Dabei werden Präsenzaufgaben sowie die Musterlösungen zu den Hausaufgaben vorgerechnet
- Erwartete Aktivitäten der Studierenden: Mitarbeit bei Präsenzübungen, Hausaufgaben, selbständiges Studium von sekundärer Literatur
- Folienskript ist vorhanden auf der Homepage der Vorlesung

Prüfungsmodalitäten

Klausuren jeweils für die beiden Veranstaltungen

Modulverantwortliche(r)

Rammig

I.3.2 Konzepte und Methoden der Systemsoftware

Rolle der Veranstaltung im Studiengang

Betrachtet man Lehrinhalte klassischer Teilgebiete der Informatik wie Übersetzerbau, Betriebssysteme, Datenbanksysteme, Rechnernetze, Verteilte Systeme oder Rechnerarchitektur, so kann man feststellen, dass es immer wieder fundamentale Problemstellungen gibt, die in den einzelnen Gebieten als Varianten auftauchen und dort mit entsprechenden Verfahren gelöst werden. Es liegt daher nahe, diese Einzelphänomene aus ihrem Kontext herauszulösen, ihre Gemeinsamkeiten herauszuarbeiten und sie als allgemeine Phänomene einmalig und grundlegend zu behandeln. Ziel der Vorlesung ist die Vermittlung allgemeiner Prinzipien, Konzepte, Methoden und Techniken, wie sie in komplexen HW/SW-Systemen mit Nebenläufigkeit vorzufinden sind. Die Studenten sollen die Gemeinsamkeiten erkennen können und die Prinzipien als grundlegend für das Fach verstehen. Sie sollen insbesondere in Entwurfssituationen diese Methoden sinnvoll einsetzen können. Die Lehrveranstaltung bildet eine Brücke zwischen der technischen und der praktischen Informatik. Basierend auf den Grundlagen der Rechnerarchitektur werden die grundlegenden Komponenten der Systemsoftware vorgestellt. Nach einer Wiederholung der wesentlichen Komponenten der Rechnerarchitektur werden Prozesse eingeführt, die eine funktionale und strukturierende Beschreibungseinheit für System- und Anwendungssoftware darstellen. Die nebenläufige und parallele Ausführung von Prozessen trägt maßgeblich zur Effizienz des Gesamtsystems bei und ist entscheidend für die Auslastung der Betriebsmittel. Allerdings erfordert die verzahnte Verarbeitung den Einsatz von Synchronisationskonzepten, die auf Sperrvariablen, Semaphore, kritischen Abschnitten und Transaktionen basieren, um die Interaktion zwischen den Prozessen und den Zugriff auf die Betriebsmittel zu organisieren. Eine Betrachtung der grundlegenden Techniken zur Transaktionsverwaltung und Sicherstellung gewünschter Eigenschaften wie Rücksetzbarkeit, Striktheit, Wiederherstellung stellt die Verbindung zum Gebiet der Datenbanken her. Insbesondere werden allgemeine Methoden für die Verwaltung der Betriebsmittel vorgestellt, welche auf zentralisierten, kooperativen und optimistischen Techniken basieren. Techniken zur Erkennung und Vermeidung von Verklemmungen schließen den Bereich der Ressourcenverwaltung. Anschließend werden die vorgestellten Methoden im Kontext der Speicherverwaltung und Scheduling detailliert betrachtet. Umgang mit logischen und virtuellen Betriebsmitteln, Speicherhierarchien, Virtualisierung, Caching und auf dem Lokalisierungsprinzip basierenden Strategien gehören zu den Schwerpunkten. Bei Scheduling wird die Ablaufplanung für konventionelle Prozesse, Echtzeitsysteme sowie für abhängige Prozesse erläutert. Der letzte Abschnitt der Vorlesung beschäftigt sich mit der Prozessinteraktion über Rechengrenzen hinweg und führt die grundlegenden Konzepte des Kanals und der Brücke sowie der entfernten Prozeduraufrufe.

Die Lehrveranstaltung bietet die Grundlagen für weiterführende Veranstaltungen wie Betriebssysteme, Verteilte Systeme, Rechnernetze, Echtzeitsysteme und teilweise für Datenbanken und Compilerbau. Die Lehrveranstaltung ist insbesondere auch für Studierende der Ingenieur-Informatik. Studierenden der Wirtschaftsinformatik mit Interesse an technischen Gegebenheiten bietet sie einen umfassenden Überblick über die Grundzüge der Systemsoftware.

Inhaltliche Gliederung der Veranstaltung

Die Vorlesung gliedert sich in drei große Bereiche Grundlagen der Systemsoftware, Ressourcenverwaltung und Interprozesskommunikation. Die zeitliche Anordnung der Lehrveranstaltung besteht aus folgenden 7 Kapiteln:

1. Grundlagen von Rechnerarchitekturen
2. Prozesse und Nebenläufigkeit
3. Prozessscheduling
4. Prozesssynchronisation und Transaktionen
5. Betriebsmittelverwaltung und Verklemmungen
6. Speicherverwaltung
7. Kooperative Prozessinteraktion

Inhaltliche Verwendbarkeit

Typische Anwendungsfelder für Inhalte und Methoden sind hauptsächlich im Bereich Betriebssysteme zu finden. Die konkreten Mechanismen zur Speicherverwaltung oder Scheduling in modernen Betriebssystemen sind abgeleitet von den vorgestellten Grundmethoden. Die Techniken zur Betriebsmittelverwaltung werden in nahezu allen Bereichen der Informatik benötigt, zum Beispiel bei dem Entwurf und der Realisierung von effizienter, echtzeitfähiger Systemsoftware. Die Synchronisationsmechanismen werden zur Transaktionsverwaltung in Datenbanken benötigt. Die parallele und nebenläufige Verarbeitung ist bei leistungsfähigen und/oder ausfallsicheren Servern unerlässlich. Schließlich bilden die Konzepte Brücke und Kanal die Grundlage für die praktische Realisierung der Netzwerkkommunikation und entfernte Prozeduraufrufe, welche bei web-basierten Informationssystemen notwendig sind.

Voraussetzungen und Vorkenntnisse

Grundlegende Kenntnisse der Programmiersprachen sowie der Rechnerarchitektur sind erforderlich. Es wird die Bereitschaft erwartet, die Zusammenhänge zwischen Betriebsmitteln und Rechnerarchitektur herauszuarbeiten und die Grundprinzipien der Entwicklung von effizienter Software zu verinnerlichen. Insbesondere sollen die globalen Konzepte und Methoden selbstständig an konkrete Beispiele angewendet werden.

Prüfbare Standards / Lernziele der Veranstaltung

Vermittlung von Faktenwissen – Inhaltskompetenz

- Zusammenhang zwischen Hardware und Systemsoftware
- Aufbau, Verwaltung und Synchronisation von Prozessen
- Techniken zur Speicherverwaltung und für Scheduling
- Techniken zur Sicherung von kritischen Bereichen
- Techniken für den Entwurf von parallelen und nebenläufigen Programmen

Vermittlung von methodischem Wissen – Methodenkompetenz

- Methoden zur effizienten Verwaltung und Zuordnung von Betriebsmitteln
- Methoden zur Erkennung und Vermeidung von Verklemmungen
- Methoden zur Kooperation zwischen Prozessen in verteilten Systemen
- Methoden für Prozessinteraktion

Vermittlung von Transferkompetenz

Übertragung der globalen Strategien auf vorgegebene Einzelsituationen, zum Beispiel im Rahmen von Übungsaufgaben

Vermittlung von normativ-bewertender Kompetenz

- Strategien zur Behandlung unterschiedlicher Strategien ausarbeiten
- Den praktischen Wert der Konzepte und Methoden der Systemsoftware erkennen

Schlüsselqualifikationen

Durch den Übungsbetrieb in kleinen Gruppen wird die Kooperations- und Teamfähigkeit gefördert.

Modulzugehörigkeit

Pflichtmodul Konzepte und Methoden der Systemsoftware

Modus

- Leistungspunkte pro Modul (Workload) : 8
- SWS (4V, 2Ü)
- Häufigkeit des Angebotes: jedes SS
- Dauer (1 Semester)

Methodische Umsetzung

Übungen in kleinen Gruppen fördern die praktische Anwendung der vorgestellten Methoden auf ausgewählte Beispiele. Insbesondere müssen Parameter und Strategien an die konkrete Situation angepasst werden. Die Aufgaben werden in Gruppen von 3 Studierenden bearbeitet, wodurch die Teamfähigkeit gefördert wird. Der Aufbau der Aufgabenzettel bildet den Aufbau der Systemsoftware nach, beginnend von der Hardware über Prozesse bis zu Betriebsmittelverwaltung und Scheduling.

Organisationsformen / Medieneinsatz / Literaturangaben

- Folien-orientierte Vorlesung, Tafelanschrieb bei Beispielen, zusätzlichen Erläuterungen und zu potenzierenden Sachverhalten
- Wöchentliche Übungen in kleinen Gruppen. Dabei werden Präsenzaufgaben sowie die Musterlösungen zu den Hausaufgaben vorgerechnet
- Erwartete Aktivitäten der Studierenden: Mitarbeit bei Präsenzübungen, Hausaufgaben, selbständiges Studium von sekundärer Literatur
- Folienskript ist vorhanden auf der Homepage der Vorlesung

Prüfungsmodalitäten

- Klausur
- Die bei den Hausaufgaben erreichten Punkte können zur Verbesserung der Klausurnote eingesetzt werden. Bei Erreichen von mehr als 35% der Punkte wird die Note um 0.3, bei mehr als 60% um 0.7 und bei mehr als 90% um eine ganze Stufe verbessert. Diese Angaben dienen lediglich zur Orientierung und werden von Semester zu Semester angepasst und in der Lehrveranstaltung angegeben. Die Bonusstufen gelten nur dann, wenn die Klausur bestanden wird.

Modulverantwortliche(r)

Rammig

I.4 Mathematik

I.4.1 Grundlagen der Mathematik

Rolle der Veranstaltung im Studiengang

Einführung in die Grundlagen der Mathematik, die während des Informatikstudiums benötigt werden.

Inhaltliche Gliederung der Veranstaltung

Das Modul besteht aus den zwei Vorlesungen Mathematik für Informatik 1 und 2, jeweils V4+Ü2, die in zwei aufeinander folgenden Semestern gelesen werden.

Gliederung „Mathematik für Informatiker I (V4+Ü2)“

Kapitel I Grundbegriffe

1. Mengen und Abbildungen
2. Vollständige Induktion und Rekursion, Kombinatorik
3. Elementare Zahlentheorie
4. Reelle Zahlen, Körper
5. Die komplexen Zahlen

Kapitel II Lineare Algebra

1. Das Rechnen mit Matrizen
2. Lineare Gleichungssysteme
3. Determinanten
4. Vektorräume
5. Lineare Abbildungen und Matrizen
6. Eigenwerte

Gliederung „Mathematik für Informatiker II (V4+Ü2)“

1. Konvergenz von Folgen
2. Konvergenz von Reihen und Potenzreihen
3. Stetigkeit
4. Exponentialfunktion und trigonometrische Funktionen
5. Polarkoordinaten, Einheitswurzeln und der Fundamentalsatz der Algebra
6. Differenzierbarkeit
7. Lokale Extrema, Taylor-Formel, Taylor-Reihen
8. Integrierbarkeit (Riemann-Integral)
9. Approximation von Nullstellen und Fixpunkten. Das Newton-Verfahren

Inhaltliche Verwendbarkeit

Die in diesen Veranstaltungen vermittelten Kenntnisse werden als Verfahren bzw. Faktenwissen im Informatikstudium gebraucht; bzw. die mathematisch-methodische Denkweise (Definition, Satz, Beweis), die hier eingeübt wird.

Voraussetzungen und Vorkenntnisse

Keine besonderen Vorkenntnisse.

Lernziele der Veranstaltung

Die Studierenden sollen mit Grundbegriffen und Grundtechniken der Linearen Algebra, der Zahlentheorie und der Analysis umgehen lernen

Schlüsselqualifikationen

- Die Fähigkeit zum abstrakten mathematischen Denken und Schließen wird entwickelt.

Modulzugehörigkeit

- Pflichtmodul

Modus

- Leistungspunkte pro Modul (Workload): 16 Leistungspunkte, 8 je Veranstaltung
- Umfang und Form des Angebots, z.B. 12 SWS (2 mal 4V,+2Ü)
- Häufigkeit des Angebotes: Dieses Modul wird in jedem Wintersemester angeboten und dauert ein Jahr.

Prüfungsmodalitäten

Fachprüfung (Klausur)

Modulverantwortliche(r)

Die Dozenten der Mathematik

I.4.2 Stochastik

Rolle der Veranstaltung im Studiengang

Einführung in die Grundlagen der Stochastik, die während des Informatikstudiums benötigt werden.

Inhaltliche Gliederung der Veranstaltung

Das Modul besteht aus der Vorlesung Mathematik für Informatik 3.1.

Gliederung „Mathematik für Informatiker 3.1 (V2+Ü1)“

1. Einführung

Wozu Stochastik (und wozu Stochastik für Informatiker)? Historisches, Literatur

2. Klassische Wahrscheinlichkeitsmodelle

Kombinatorische Modelle: Grundprobleme in verschiedenen Terminologien, Urnenmodell: Ziehungen mit/ohne Berücksichtigung der Reihenfolge, Permutationen, Anwendungen, Geometrische Modelle: Prinzip, Problem der Modellbildung (Bertrand)

3. Grundbegriffe

Axiomatik: Begriff des Wahrscheinlichkeitsraumes, Modellbeispiele, Rechenregeln für Ereignisse und Wahrscheinlichkeiten, Diskrete Wahrscheinlichkeitsräume, Laplace- u.a. Räume, geometrische und Poisson-Verteilung; Bedingte Wahrscheinlichkeiten und Unabhängigkeit: Begriffe, Formeln (totale Wahrscheinlichkeit, Bayes, . . .)

Binomial- und verwandte Verteilungen: Bernoulli-Schema; Hypergeometrische, Polynomial-, negative Binomialverteilung; Grenzwertsatz von Poisson

Beispiele für nichtdiskrete Wahrscheinlichkeitsräume: Begriffe: Zufallsvariable, Verteilung; Beschreibung von Verteilungen durch Verteilungs- bzw. Dichtefunktionen

4. Kenngrößen von Zufallsgrößen und -vektoren

Median und andere Quantile

Momente: Begriffe: Erwartungen, Streuung, Kovarianz; Interpretation; Berechnung (Transformationssatz, Beispiele)

Ungleichungen für Momente: Markow- und Tschebyscheff-Ungleichung; Anwendungen (Wahrscheinlichkeitsabschätzungen, Konvergenz)

5. Gesetze der großen Zahlen

schwaches und starkes Gesetz der großen Zahlen; Anwendungen: Parameterschätzung, Monte-Carlo Methoden

6. Zentraler Grenzwertsatz

Zentraler Grenzwertsatz von de Moivre-Laplace und von Lindeberg-Feller; Anwendungen (Approximative Wahrscheinlichkeitsrechnungen)

7. Schätzen und Testen

Problemstellung, Parameterschätzung: Maximum-Likelihood-Schätzer und Momentenschätzer, einfache Güteeigenschaften, Signifikanztests: Struktur, Grundbeispiele (t-Test)

8. Zufallszahlen

Problemstellung; Beispiele für Generatoren; Transformationen von Zufallszahlen

Inhaltliche Verwendbarkeit

Die in diesen Veranstaltungen vermittelten Kenntnisse werden als Verfahren bzw. Faktenwissen im Informatikstudium gebraucht.

Voraussetzungen und Vorkenntnisse

Das Modul „Grundlagen der Mathematik“ muss vorher absolviert worden sein.

Lernziele der Veranstaltung

Die Studierenden sollen mit Grundbegriffen und Grundtechniken der Stochastik umgehen lernen.

Schlüsselqualifikationen

- Die Fähigkeit zum abstrakten mathematischen Denken und Schließen wird entwickelt.

Modulzugehörigkeit

- Pflichtmodul

Modus

- Leistungspunkte pro Modul (Workload): 4 Leistungspunkte
- Umfang und Form des Angebots, z.B. 3 SWS (2V,+1Ü)
- Häufigkeit des Angebotes: Dieses Modul wird in jedem Wintersemester angeboten und dauert ein Semester.

Prüfungsmodalitäten

Fachprüfung (Klausur)

Modulverantwortliche(r)

Die Dozenten der Mathematik

II. Module im 2. Studienabschnitt des Bachelorstudiengangs

II.1 Gebiet Softwaretechnik und Informationssysteme

II.1.1 Softwaretechnik und Informationssysteme

Rolle im Studiengang

Die Entwicklung, Inbetriebnahme und Wartung von Softwaresystemen gehören zu den wichtigsten Aufgaben heutiger Informatiker. Die größte Schwierigkeit bei diesen Aufgaben ist die Beherrschung der Größe und der Komplexität heutiger und zukünftiger Softwaresysteme. Zusätzlich werden diese Aufgaben noch dadurch erschwert, daß in bestimmten Bereichen Soft- und Hardware genau aufeinander abgestimmt sein müssen. Um diese Aufgaben zu bewältigen, benötigen Informatiker eine breite Palette von Kenntnissen und Fähigkeiten auf dem Gebiet der Softwaretechnik und der Informationssysteme.

Aufbauend auf den im 1. Studienabschnitt vermittelten grundlegenden Konzepten und Methoden der Softwaretechnik und den praktischen Erfahrungen aus dem Softwaretechnikpraktikum vermittelt dieser Modul einen breiten Überblick über die wichtigsten Konzepte, Notationen und Methoden der Softwaretechnik und ihrer formalen und mathematischen Grundlagen. Die vermittelten Kenntnisse sollen die Studierenden in die Lage versetzen, Softwaresysteme unter vorgegebenen technischen und ökonomischen Randbedingungen zu entwickeln. Darüber hinaus sollen die Studierenden das wissenschaftliche Handwerkszeug beherrschen, um sich im späteren Berufsleben in zukünftige Techniken einzuarbeiten.

Die Veranstaltungen dieses Moduls decken verschiedene Teilgebiete des Gebietes "Softwaretechnik und Informationssysteme" ab. Die Auswahl gibt einen repräsentativen Überblick über das gesamte Gebiet und über die verschiedenen Phasen der Softwareentwicklung.

Die Veranstaltungen dieses Moduls können später im Masterstudiengang Informatik durch Wahlpflichtmodule im Bereich Softwaretechnik in verschiedenen Schwerpunkten vertieft werden.

Inhaltliche Gliederung des Moduls

Zum Absolvieren dieses Moduls wählen die Studierenden zwei Veranstaltungen aus dem folgenden Katalog aus:

- Modellbasierte Softwareentwicklung (MSWE)
- Programmiersprachen und Übersetzer (PSÜ)
- Grundlagen der Wissensbasierten Systeme (GWBS)
- Software Modellierung mit Formalen Methoden (SMFM)
- Datenbanken und Informationssysteme 1 (DBIS1)

Die Ziele und der Inhalt der einzelnen Veranstaltungen sind nachfolgend beschrieben

Modellbasierte Softwareentwicklung (MSWE)

Die Studierenden sollen grundlegende Verfahren zur Konstruktion großer Softwaresysteme kennen und ihre Anwendung beherrschen. Sie sollen die Vor- und Nachteile formaler und informaler Spezifikationstechniken erfahren und die Notwendigkeit von Design erkennen und abstrakte Modelle zur Verbesserung der Softwarequalität einsetzen können. Insbesondere

wird auf das Paradigma der „Model Driven Development“ eingegangen, das einen wesentlichen Produktivitäts- und Qualitätsgewinn bei der Softwareentwicklung verspricht.

Inhalt:

1. Spezifikationstechniken für Analyse und Design:
Strukturorientierte, operationale und deskriptive Techniken
2. Automatische Codegenerierung aus dem Design
3. Validierung und Verifikation von Softwaresystemen:
Testen und Modelchecking

Programmiersprachen und Übersetzer (PSÜ)

Sprachen spielen in der Softwaretechnik vielfältige und wichtige Rollen: Als Programmiersprachen sind sie Ausdrucksmittel für die Programmentwicklung und dabei auf eine bestimmte Programmiermethode zugeschnitten. Als Spezifikationssprachen dienen sie zur Formulierung von Aufgabenbeschreibungen im allgemeinen oder sind für bestimmte Anwendungsgebiete Beschreibungsmethoden speziell zugeschnitten. Der Entwurf und die Implementierung solcher Sprachen durch Übersetzer oder Generatoren sind bedeutende Themengebiete der Softwaretechnik.

Diese Veranstaltung vermittelt Kenntnisse und Fähigkeiten zum vertieften Verständnis, zur Spezifikation und zur Implementierung von Programmier- und Spezifikationssprachen. Die Teilnehmer werden befähigt,

- grundlegende Kalküle zur präzisen Beschreibung von Spracheigenschaften anzuwenden,
- grundlegende Methoden zur Implementierung von Sprachen anzuwenden.

Inhalt:

1. Ebenen von Spracheigenschaften und Struktur von Übersetzern
2. Spezifikation von Grundsymbolen und lexikalische Analyse
3. Syntaktische Spezifikation und Analyse
4. Semantische Eigenschaften und Analyse
5. Spezifikation dynamischer Semantik und Übersetzung

Grundlagen der Wissensbasierten Systeme (GWBS)

Die Vorlesung „Grundlagen Wissensbasierte Systeme“ stellt Basiswissen zur Deduktion sowie Methoden der symbolischen Wissensverarbeitung vor. Wichtige Ziele sind die Vermittlung von Grenzen und Möglichkeiten gängiger Wissensrepräsentationsformen und die Vorstellung ihrer formalen Grundlagen.

Die in der Vorlesung erworbenen Kenntnisse und Fähigkeiten sollen Studenten in die Lage versetzen, Softwaresysteme zu entwickeln, in denen Aspekte wie Unsicherheit und Vagheit berücksichtigt werden müssen oder menschliches Problemlöseverhalten nachgebildet werden soll.

Inhalt:

1. Künstliche Intelligenz (Begriff, Geschichte, Gebiete)
2. Wissensformen (sub/symbolisch, Problemlösung), Expertensysteme
3. Aussagenlogische Deduktion und Entscheidungsprobleme

4. Prädikatenlogische Deduktion
5. Produktionsregelsysteme
6. Unschärfe und Vagheit (z.B. Fuzzy Logic)
7. Einordnung maschineller Lernverfahren

Software Modellierung mit Formalen Methoden (SMFM)

Formale Methoden sind Sprachen zur Modellierung von Softwaresystemen auf einer gewissen Abstraktionsebene. Da sie eine formale Semantik besitzen, können die so beschriebenen Modelle auf ihre Korrektheit analysiert werden. Dies ist insbesondere für sicherheitskritische Systeme wichtig.

Die Vorlesung vermittelt Kenntnisse und Fähigkeiten in der Modellierung und Analyse von Softwaresystemen. Die Studenten sollen ferner in die Lage versetzt werden, passende Sprachen und Analysetechniken für die Modellierung auszuwählen.

Inhalt:

1. Einführung in formale Spezifikationssprachen
2. Modellierung paralleler kommunizierender Systeme
3. Analysetechniken
4. Zustandsbasierte Formalismen

Datenbanken und Informationssysteme 1 (DBIS1):

Inhalt:

1. Anfragesprachen
2. Anfrageoptimierung
3. Deduktive Datenbanken
4. Web-Server-Technologien und Web-Datenbank-Kopplung
5. Semistrukturierte Daten, Werkzeuge und Standards um XML

Inhaltliche Verwendbarkeit

Absolventen dieses Moduls sollen fähig sein, Software- und Informationssysteme gemäß der gängigen Techniken zu entwickeln und sich neue Techniken anzueignen und sie zu bewerten.

Die Veranstaltung GWBS soll allgemeine formale Grundlagen der Softwaretechnik legen, während die Veranstaltungen MSWE, PSÜ, SMFM und DBIS1 die speziellen Konzepte und Methoden des jeweiligen Teilgebietes vermitteln.

Die Veranstaltungen aus DBIS1 und GWBS decken stärker den Bereich der Informationssysteme ab, die Veranstaltungen MSWE, PSÜ und SMFM stärker den Bereich Softwaretechnik.

Voraussetzungen und Vorkenntnisse

Die Voraussetzungen zum Absolvieren dieses Moduls sind die Fähigkeit zur Modellierung und Formalisierung von Sachverhalten, wie sie u.a. im Modul "Modellierung" vermittelt werden. Außerdem werden die Beherrschung einer Programmiersprache und der gängigen Notationen der Objektorientierten Modellierung sowie erste Erfahrungen in der

Softwareentwicklung vorausgesetzt, wie sie in den Modulen "Grundlagen der Programmierung", "Softwaretechnik" und im "Softwaretechnikpraktikum" vermittelt werden.

Lernziele

Die Studierenden sollen ...

Vermittlung von Faktenwissen

- einen breite Überblick über die grundlegenden Konzepte der Programmierung und der Softwaretechnik besitzen
- die gängigen Prinzipien, Notationen und Sprachen zur Modellierung und Entwicklung von Software kennen
- die Probleme, die bei der Entwicklung von Software auftreten, und Methoden und Verfahren zu ihrer Bewältigung kennen

Vermittlung von methodischem Wissen

- grundlegende Methoden zur Formalisierung und Modellierung anwenden können
- grundlegende Methoden der Softwareentwicklung anwenden können

Vermittlung von Transferkompetenz

- sich neue Methoden und Notationen der Softwaretechnik aneignen und bewerten können
- neue Methoden und Konzepte der Softwaretechnik definieren und formulieren können

Vermittlung von normativ-bewertender Kompetenz

- die Notwendigkeit der systematischen Softwareentwicklung erkennen

Schlüsselqualifikationen

- Kommunikations- und Teamfähigkeit in den Übungen
- Strategien des Wissenserwerbs:
Kombination aus Vorlesung, Vor- und Nachbereitung am Vorlesungsmaterial und ergänzender Literatur, Gruppenarbeit, Hausaufgaben.
- Bewertung und Hinterfragung neuer Konzepte

Modulzugehörigkeit

Pflichtmodul des 2. Studienabschnittes im Gebiet „Softwaretechnik und Informationssysteme“.

Modus

Leistungspunkte: 3+3 ECTS (3 pro Veranstaltung)

SWS: 2V+1Ü, 2V+1Ü

Häufigkeit:

Alle Veranstaltungen dieses Moduls werden im jährlichen Zyklus angeboten: Die Veranstaltungen MSWE, PSÜ und DBIS1 werden voraussichtlich jeweils im Wintersemester angeboten, die Veranstaltungen GWBS und SMFM im Sommersemester.

Methodische Umsetzung

- Methoden und Techniken werden an typischen Beispielen eingeführt und diskutiert

- in den Übungen werden sie mit Rechner- und Werkzeugunterstützung praktisch erprobt

Organisationsform, Medieneinsatz, Literaturangaben

- Vorlesungen mit Folienpräsentation oder Tafelanschrift
- ergänzende Materialien zur Vorlesung im Internet
- in den Übungen wird die Lösung der Aufgaben gemeinsam erarbeitet
- erwartete Aktivitäten der Studierenden:
 - Mitarbeit bei der Erarbeitung der Lösung in den Übungen,
 - Hausaufgaben, Vor- und Nacharbeit der Vorlesungen

Prüfungsmodalitäten

Die Veranstaltungen dieses Moduls werden jeweils einzeln geprüft (je nach Anzahl der Teilnehmer mündlich oder schriftlich).

Modulverantwortlicher

Kastens

II.2 Gebiet Modelle und Algorithmen

II.2.1 Modelle und Algorithmen

Rolle der Veranstaltung im Studiengang

Algorithmen bilden die Grundlage jeder Hardware und Software: Ein Schaltkreis setzt einen Algorithmus in Hardware um, ein Programm macht einen Algorithmus "für den Rechner verstehbar". Algorithmen spielen daher eine zentrale Rolle in der Informatik.

Deshalb steht im Mittelpunkt des Bachelormoduls Modelle und Algorithmen die Klassifizierung von Problemen bezüglich ihrer algorithmischen Komplexität. Als Maße für Komplexität werden insbesondere Laufzeit und Speicherbedarf, aber auch z.B. Parallelisierbarkeit herangezogen. Veranstaltungen dieses Moduls behandeln sowohl die Entwicklung und Analyse effizienter Algorithmen und algorithmischer Techniken, als auch die Untersuchung der Problem-inhärenten Komplexität, d.h. den Nachweis unterer Komplexitätsschranken und den Komplexitätsvergleich von Problemen. Weiter ergänzt wird der Modul durch eine Veranstaltung zur Kryptographie. Hier wird die inhärente Schwierigkeit von Problemen, wie sie die Komplexitätstheorie nachzuweisen versucht, positiv etwa für den Entwurf sicherer Verschlüsselungsverfahren genutzt.

Inhaltliche Gliederung der Veranstaltung

Der Modul besteht aus den folgenden Veranstaltungen

- Grundlegende Algorithmen
- Komplexitätstheorie
- Methoden des Algorithmenentwurfs
- Parallelität und Kommunikation
- Optimierung
- Einführung in Kryptographie

Grundsätzlich können zur Absolvierung des Moduls zwei beliebige dieser Veranstaltungen gehört werden. Als Einstiegsveranstaltungen werden jedoch Grundlegende Algorithmen, Komplexitätstheorie, Kryptographie und Parallelität und Kommunikation empfohlen. Dabei sollte eine der beiden Veranstaltungen Grundlegende Algorithmen und Komplexitätstheorie gehört werden. Die Vorlesungen Optimierung und Algorithmische Methoden sollten möglichst nur im Anschluss an die Vorlesung Grundlegende Algorithmen gehört werden.

Im Einzelnen sind die Vorlesungen inhaltlich wie folgt gegliedert.

1. Grundlegende Algorithmen

- Graphenalgorithmen wie kürzeste Wege, Flüsse in Netzwerken (Grundlagen) und Matchings
- Universelles und perfektes Hashing
- String matching

2. Komplexitätstheorie

- Hierarchiesätze
- Gödel'sche Unvollständigkeitssätze

- P-, NP- und PSPACE-Vollständigkeit
- Vergleiche zwischen Komplexitätsklassen
- 3. Algorithmische Methoden
 - Einführung in online-Algorithmen, Randomisierung und Approximation
 - Optimierungsheuristiken
- 4. Parallelität und Kommunikation
- 5. Optimierung
- 6. Einführung in Kryptographie
 - Aufgaben der Kryptographie
 - Symmetrische und asymmetrische Verfahren
 - Elementare Sicherheitskonzepte und Kryptanalyse
 - Die symmetrischen Chiffren DES und AES
 - Hashfunktionen und MACs
 - Diffie-Hellman Schlüsselaustauschverfahren und RSA

Inhaltliche Verwendbarkeit

Für gegebene Probleme nicht nur irgendwelche, sondern ressourcenschonende, d.h. effiziente Algorithmen zu entwerfen, und Probleme bezüglich ihrer inhärenten Komplexität einzuschätzen, beschreibt für viele Teilgebiete (nicht nur) der Informatik eine wichtige Fähigkeit. Datenbanken und Informationssysteme, Kommunikationsprotokolle und Ressourcenmanagement in Rechnernetzen, Computergrafik-Systeme und wissenschaftliches Rechnen sind wichtige Beispiele. In vielen Anwendungen sind es häufig Optimierungsprobleme, die effizient gelöst werden sollen. Auf diesen Aspekt geht die Vorlesung Optimierung intensiv ein. Die Inhalte der Kryptographie finden Anwendung u.a. im E-Banking und im E-Commerce.

Voraussetzungen und Vorkenntnisse

Die wesentlichen Konzepte aus den Bereichen Algorithmen, Datenstrukturen, Berechenbarkeit und Komplexitätstheorie wie sie in den ersten vier Semestern vermittelt werden vorausgesetzt. Untereinander sind die Abhängigkeiten der Veranstaltungen des Moduls wie im Abschnitt Inhaltliche Gliederung erläutert.

Lernziele der Veranstaltung

Ziel der Veranstaltung dieses Moduls ist es, den Studierenden Einblick in einige der wichtigsten Themen und Problemkreise aus dem Bereich Modelle und Algorithmen zu geben. Entsprechend dem Schwerpunkt Algorithmik kommen viele dieser Themen aus dem Bereich des Algorithmenentwurfs und der Algorithmenanalyse. Diese Themen werden vor allem in den Veranstaltungen Grundlegenden Algorithmen, Algorithmische Methoden, Optimierung und Parallelität und Kommunikation behandelt. Am Ende dieser Veranstaltungen sollten Studierende wichtige Algorithmen und algorithmische Techniken nicht nur kennen, sondern sie auch gezielt modifizieren und anwenden können. Die Veranstaltungen Komplexitätstheorie und Kryptographie sollen die Veranstaltungen aus dem Bereich Algorithmik ergänzen und komplementieren, indem sie nicht nur die Grenzen von (effizienten) Algorithmen aufzeigen, sondern auch die Nützlichkeit dieser Grenzen beim

Entwurf von kryptographischen Verfahren aufzeigen. Auch bei diesen Veranstaltungen sollen Studenten am Ende die gelernten Techniken auf neue Probleme anwenden können, um so etwa die Komplexität eines Problems oder die Sicherheit eines kryptographischen Verfahrens einschätzen zu können.

Modulzugehörigkeit

Wahlmodul im Gebiet Modelle und Algorithmen.

Modus

- Leistungspunkte : 3+3
- SWS 2+1, 2+1
- Häufigkeit des Angebotes: Jedes Semester werden 2-4 Veranstaltungen des Katalogs angeboten.

Prüfungsmodalitäten

Je nach Teilnehmerzahl mündliche Prüfung oder Klausur

Modulverantwortliche(r)

Blömer.

II.3 Gebiet Eingebettete Systeme und Systemsoftware

II.3.1 Eingebettete Systeme und Systemsoftware

Rolle des Moduls im Studiengang

Eingebettete Systeme und Systemsoftware spielen die zentrale Rolle im Zuge der sich ständig verstärkenden Informatisierung aller technischen Systeme. Die Systemsoftware ist jede grundlegende Softwareschicht, welche eine Verbindung zwischen der Rechnerhardware und der Software herstellt. Zusammen mit anderen Komponenten der Systemsoftware wird die Erstellung von Anwendungen ermöglicht und eine Schnittstelle zu den Hardwareressourcen zur Verfügung gestellt. Eine besondere Bedeutung kommt dabei den Rechnernetzen zu, dass diese die Brücke zwischen den räumlich getrennten Ressourcen aufbauen und die Grundlage für den Aufbau von verteilten Systemen bieten. Letztere umfassen unter anderem webbasierte Dienste, kooperative Anwendungen, effiziente und ausfallsichere Verarbeitung. In allen Fällen ist es jedoch notwendig, dass die Umsetzung für den Benutzer möglichst transparent, zuverlässig und sicher erfolgt.

Eines der wesentlichen Gebiete für den Entwurf und Einsatz von Systemsoftware findet sich im Bereich der Eingebetteten Systeme. Darunter werden die informationsverarbeitenden Anteile in Systemen verstanden, die in der Regel aus dedizierter Hardware und darauf aufsetzender Software bestehen. Beides wird mit den grundsätzlichen Methoden der Informatik entworfen, wobei das Zusammenspiel zwischen HW und SW eine besonders wichtige Rolle spielt und im Bereich HW/SW-Codesign betrachtet wird. Dabei müssen die physikalischen Gesetze des Gesamtsystems unbedingt berücksichtigt werden: Neben Realzeitanforderungen ist hier die Ressourcenbeschränktheit (z.B. bezüglich des Stromverbrauchs oder der verfügbaren Chipfläche) zu nennen. Dies führt dazu, dass der allgemeine Entwurfszyklus von Informatiksystemen in allen Phasen spezifisch angepasst werden muss, so dass spezifische Spezifikations- und Modellierungstechniken erforderlich sind.

Dieses Modul auf den in den Modulen „Konzepte und Methoden der Systemsoftware“ und „Technische Informatik“ vorgestellten Grundlagen auf und führt die Studierenden im Bachelorabschnitt in das Gebiet „Eingebettete Systeme und Systemsoftware“ ein. Die enthaltenen Veranstaltungen sind so ausgewählt, dass ein möglichst breiter Überblick gewährleistet wird. Dabei kann eine grobe Aufteilung in SW-lastige Aspekte (Verteilte Systeme und Rechnernetze) sowie HW-lastige Aspekte (Eingebettete Systeme und HW/SW-Codesign) identifiziert werden. Da in diesem Modul nur eine Teilmenge der Vorlesungen gehört werden kann, finden sich die Veranstaltungen auch in einzelnen Modulen für den Masterteil wieder, so dass die Einführung in den Bereich ESS vollständig abgedeckt werden kann. Allerdings wird eine gemäß den eigenen Interessen gezielte Wahl von Veranstaltungen nachdrücklich unterstützt, da dadurch eine höhere Spezialisierung im bestimmten Teilen im Masterabschnitt erreicht werden kann.

Inhaltliche Gliederung des Moduls

Das Modul vermittelt eine breite Einführung in die Gebiete Eingebettete Systeme, HW/SW-Codesign, Verteilte Systeme und in Rechnernetze.

Im Bereich HW/SW-Codesign wird der Entwurfszyklus eines integrierten HW/SW-Entwurfs von der Spezifikation und Modellierung über Analyse und Validierung, die HW/SW-Partitionierung, die HW/SW-Synthese bis hin zur Systemintegration und Verifikation

erläutert. Die Veranstaltung über Eingebettete Systeme liefert einen Überblick über die Aufgabenstellung und grundsätzliche Lösungsansätze. Insbesondere wird auf die Terminologie und Klassifikation von eingebetteten Systemen, Entwurfsumgebungen sowie den Echtzeitaspekten eingegangen. In der einführenden Veranstaltung über Rechnernetze werden die Aufgaben und Architektur von Kommunikationssystemen erörtert und Fragestellung hinsichtlich Aufbau und Klassifikation von Rechnernetzen, Adressierung, Routing, Protokollen, Netzwerk-Topologien und Technologien analysiert. Die Veranstaltung über Verteilte Systeme betrachtet schließlich charakteristische Eigenschaften, System-, Architektur und Programmiermodelle, Namens- und Erkennungsdienste sowie grundlegende Algorithmen zum Design und zur Erstellung von verteilten Systemen als wichtigen Bestandteil moderner Informationssysteme.

Inhaltliche Verwendbarkeit

Typische Anwendungsfelder sind in der Anwendungsentwicklung, Systemadministration sowie Entwurf- und Umsetzung von Spezialsystemen zu finden. Die vorgestellten Mechanismen zur Ressourcenverwaltung, Sicherheit, rechnerübergreifenden Kommunikation werden sowohl bei klassischen Informationssystemen als auch – in angepasster Form – bei speziellen Hardwareressourcen angewendet. Die vorgestellten Methoden zur Spezifikation, Modellierung, Analyse, Synthese und Verifikation werden im gesamten Bereich technischer Systeme benötigt. Realzeitanwendungen finden aber auch im nichttechnischen Umfeld Verwendung, z.B. in der Wetterprognose oder in der strategischen Planung von Finanzdienstleistungen. Die Grundlagen der Rechnernetze und die Grundbausteine zur Erstellung verteilter Systeme werden bei Internetanwendungen, Webdiensten, Unternehmenssoftware, usw. benötigt. Schließlich soll das vermittelte Wissen, die Bewertung, Auswahl und Anpassung verschiedener Lösungswege und -komponenten an eine spezifische Aufgabenstellung ermöglichen.

Vorraussetzungen und Vorkenntnisse

Als Vorkenntnisse werden die Inhalte der Module Technische Informatik und KMS vorausgesetzt. Ferner sind grundlegende Kenntnisse von Modellierungsprinzipien aus dem Modul Modellierung und der Programmiersprachen aus SWE zwingend erforderlich. Darüber hinaus wird die Bereitschaft erwartet, sich in systemnahe Programmiersprachen und Hardwarebeschreibungssprachen einzuarbeiten.

Prüfbare Standards / Lernziele der Veranstaltung

Die Studierenden sollen das Verständnis der spezifischen Eigenschaften von Systemsoftware erlangen sowie die elementaren Bausteine zum Aufbau von Betriebs- und verteilten Systemen kennen lernen. Außerdem sollen die Studierenden grundlegende Konzepte und unterschiedliche Funktionsweisen von Rechnernetzen und deren Nutzung kennen lernen und so verstehen, dass eine Übertragung der Grundkonzepte auf neue Netzwerkstrukturen und –technologien möglich ist. Im Bereich der eingebetteten System sollen die Studierenden das Verständnis der spezifischen Eigenschaften von Eingebetteten Systemen erlangen, die elementaren Konzepte zum Entwurf derartiger Systeme als gemischte HW/SW-Implementierungen und Kriterien für die Partitionierung in HW/SW kennen lernen. Sie sollen in der Lage sein, die spezifischen Restriktionen, die sich durch die physikalischen Gesetze des umgebenden Systems ergeben, einzuschätzen und lernen, diese gezielt in den Entwurfsprozess einzubeziehen. Schließlich sollen sie lernen, wie spezifische Methoden aus der Softwaretechnik einerseits und dem Hardwareentwurf andererseits zu einer leistungsfähigen Entwurfsmethodik kombiniert werden können.

Vermittlung von Faktenwissen – Inhaltskompetenz

- Zusammenhang zwischen Hardware und Systemsoftware
- Aufbau, Verwaltung und Synchronisation von Prozessen
- Techniken zur Speicherverwaltung und für Scheduling
- Techniken zur Sicherung von kritischen Bereichen
- Techniken für den Entwurf von parallelen und nebenläufigen Programmen

Vermittlung von methodischem Wissen – Methodenkompetenz

- Methoden zur effizienten Verwaltung und Zuordnung von Betriebsmitteln
- Methoden zur Erkennung und Vermeidung von Verklemmungen
- Methoden zur Kooperation zwischen Prozessen in verteilten Systemen
- Methoden für Prozessinteraktion

Vermittlung von Transferkompetenz

Übertragung der globalen Strategien auf vorgegebene Einzelsituationen, zum Beispiel im Rahmen von Übungsaufgaben

Vermittlung von normativ-bewertender Kompetenz

- Strategien zur Behandlung unterschiedlicher Strategien ausarbeiten
- Den praktischen Wert der Konzepte und Methoden der Systemsoftware erkennen

Schlüsselqualifikationen

Durch den Übungsbetrieb in kleinen Gruppen wird die Kooperations- und Teamfähigkeit gefördert.

Modulzugehörigkeit

Pflichtmodul im Gebiet „Eingebettete Systeme und Systemsoftware“

Modus

- Leistungspunkte pro Modul (Workload) : 6
- Leistungspunkte der Veranstaltung: jeweils 3
- SWS (2V+1Ü, 2V+1Ü)
- Häufigkeit des Angebotes: 2-4 Veranstaltungen pro Jahr im WS und SS
- Dauer (2 Semester)

Methodische Umsetzung

Übungen in kleinen Gruppen fördern die praktische Anwendung der vorgestellten Methoden auf ausgewählte Beispiele. Insbesondere müssen Parameter und Strategien an die konkrete Situation angepasst werden. Die Aufgaben werden in Gruppen von 3 Studierenden bearbeitet, wodurch die Teamfähigkeit gefördert wird. Der Aufbau der Aufgabenzettel bildet den Aufbau der Systemsoftware nach, beginnend von der Hardware über Prozesse bis zu Betriebsmittelverwaltung und Scheduling.

Organisationsformen / Medieneinsatz / Literaturangaben

- Folien-orientierte Vorlesung, Tafelanschrieb bei Beispielen, zusätzlichen Erläuterungen und zu potenzierenden Sachverhalten
- Wöchentliche Übungen in kleinen Gruppen. Dabei werden Präsenzaufgaben sowie die Musterlösungen zu den Hausaufgaben vorgerechnet

- Erwartete Aktivitäten der Studierenden: Mitarbeit bei Präsenzübungen, Hausaufgaben, selbständiges Studium von sekundärer Literatur
- Folienskript ist vorhanden auf der Homepage der Vorlesung

Prüfungsmodalitäten

- In der Regel Einzelprüfungen pro Veranstaltung und anschließende Berechnung des Mittelwertes der beiden Einzelleistungen. Die Prüfungen zu den einzelnen Veranstaltungen im Modul werden je nach Teilnehmerzahl als Klausur oder als Fachgespräch durchgeführt. Abweichungen von diesem Schema sowie zusätzliche Forderungen werden bei der Ankündigung der entsprechenden Vorlesungen angegeben.

Modulverantwortliche(r)

Rammig

II.4 Gebiet Mensch-Maschine-Wechselwirkung

II.4.1 Mensch-Maschine-Wechselwirkung

Rolle der Veranstaltung im Studiengang

Das Modul „Mensch-Maschine-Wechselwirkung“ führt in die wesentlichen Aspekte der Kommunikation zwischen Mensch und Maschine ein. Aus Informatik-sicht sind dabei die Themen der Entwicklung benutzergerechter Benutzungsschnittstellen, sowie die gestalterische Sicht auf Computerbedienung generell und Internetnutzung im speziellen, sowie die Einordnung von Computern in den gesellschaftlichen Kontext die zentralen Themen. Auf der Entwicklungsseite werden einerseits die konzeptuellen und mathematischen Grundlagen, sowie Algorithmen und Werkzeuge für die Erzeugung grafischer Darstellungen behandelt. Andererseits werden die Modelle und Techniken aus dem Bereich des Usability Engineering vermittelt, die im Rahmen der industriellen Softwareentwicklung für die Berufspraxis des Informatikers inzwischen an erheblichem Einfluß gewonnen hat. Insbesondere werden auch die Notwendigkeiten und Fragestellungen des Entwurfs von Webauftritten als erheblich relevantes Thema aufgegriffen. Dies komplementiert die in den technisch orientierten Informatikbereichen vermittelten Kenntnisse und Fähigkeiten des Erstellens und Umsetzens. Da Technologie im gesellschaftlichen Kontext existiert, werden in einer weiteren Veranstaltung die Rolle der Informatik, zu beachtende Konsequenzen, Problemfelder und Berührungspunkte mit anderen Disziplinen und Berufen behandelt. Dies vermittelt den Studierenden der Informatik die Verantwortlichkeit ihrer Tätigkeit und macht sensibel für Aspekte, die über das rein Technische hinausgehen.

Inhaltliche Gliederung der Veranstaltung

Der Modul besteht aus den folgenden Veranstaltungen

- Computergrafik I
- Kontextuelle Informatik
- Usability Engineering
- Gestaltung von Webauftritten (geplant)

Im Einzelnen sind die Vorlesungen inhaltlich wie folgt gegliedert:

1. Computergrafik I
 - 1. Was ist Computergrafik (Einführung und Geschichte)
 - 2. Mathematik der Computergrafik
 - 3. Die Grafik-Pipeline
 - 4. OpenGL
 - 5. Input und Interaktivität
 - 6. Transformationen in 2d und 3d
 - 7. Modellieren dreidimensionaler Szenen
 - 8. Projektionen
 - 9. Schattierungsberechnung
 - 10. Clipping

- 11. Entfernen verdeckter Oberflächen
 - 12. Rastern von Linien
 - 13. Farbe
2. Kontextuelle Informatik
 - Besonderheiten von Software als Ingenieurprodukt
 - Maschinelle DV-Modelle kognitiver Prozesse
 - Biologische Informationsverarbeitung
 - Fehler in technischen und natürlichen Systemen
 - Produkt-Prozess-Komplementarität
 - Software-Entwicklung als Lernprozess
 - Informationsethik
 - Verantwortung des Informatikers
 3. Usability Engineering (engl. Veranstaltung)
 - Usability Engineering: Basic Definition and Examples
 - The Human User
 - Modelling Rational Human Behaviour
 - Rules for the Design of Usable User Interfaces
 - The Usability Development Process
 - Web Site Usability
 4. Gestaltung von Webauftritten (vorl. Gliederung)
 - Web Design Probleme
 - Gestaltung des Inhalts
 - Gestaltung der Seitenstrukturen
 - Navigation
 - Layout, Grafik, Typografie
 - Internationalisierung
 - Personalisierung

Inhaltliche Verwendbarkeit

Bis auf die in der Vorlesung Computergrafik I vermittelten Inhalte werden wohl alle Informatiker in ihrer Berufspraxis mit den Inhalten der Vorlesungen dieses Moduls konfrontiert werden. Benutzertests werden mittlerweile in erheblichem Umfang durchgeführt und sind als absolut unverzichtbar erkannt worden; auch die benutzergerechte Gestaltung von Webauftritten und die Einordnung der Informatik-Tätigkeiten in das gesellschaftliche und arbeitsorganisatorische Umfeld sind Inhalte von großer Relevanz für jeden Informatik-Absolventen. Dies betrifft sowohl die Tätigkeit als Systementwickler, als auch im beratenden Umfeld. Die Computergrafik ist ein Spezialgebiet, das erhebliche Relevanz im Entwicklungsbereich hat, da durch die Verbreitung von leistungsfähigen Computern und

Bildschirmen mittlerweile eine ausgereifte Grafikdarstellung in fast jedem Kontext benötigt wird.

Voraussetzungen und Vorkenntnisse

Dieses Modul ist das erste aus dem Gebiet Mensch-Maschine-Wechselwirkung, das die Studierenden absolvieren müssen. Neben den im ersten Abschnitt erworbenen Grundkenntnissen der Informatik sind daher keine weiteren Vorkenntnisse erforderlich.

Lernziele der Veranstaltung

Ziel der Veranstaltung dieses Moduls ist es, den Studierenden Einblick in einige der wichtigsten Themen und Problemkreise aus dem Bereich Mensch-Maschine-Wechselwirkung zu geben. Innerhalb der Computergrafik-Vorlesung sollen die Studierenden die mathematischen Grundlagen der Grafikerzeugung, die dabei auftretenden Probleme der Softwaretechnik und ihre algorithmische Lösung kennenlernen. Außerdem werden in der Vorlesung Fähigkeiten und Kenntnisse vermittelt, die es den Studierenden erlauben einschlägige Grafiksysteme zu benutzen und zu bewerten. In der Veranstaltung Kontextuelle Informatik wird den Studierenden die Rolle der Informatik in der Gesellschaft deutlich gemacht – es geht um soziologische, psychologische, ökonomische, arbeitsorganisatorische und rechtliche Aspekte dieser Technologie. Dabei sollen die Studierenden sensibilisiert werden für die Wirkung, die ihre spätere Berufstätigkeit in verschiedenen Bereichen des menschlichen Lebens haben kann (Technikfolgenabschätzung) und zu verantwortlichem Umgang mit der Informatik führen. Die Vorlesung Usability Engineering vermittelt den Studierenden die grundlegenden Kenntnisse und methodischen Vorgehensweisen zum Entwickeln von Bedienoberflächen, die benutzergerecht und aufgabengerecht gestaltet sind. Dabei wird auch die Entwicklung, Planung und Durchführung von Benutzertests vermittelt und praktisch erprobt. Den Studierenden wird dadurch bewusst gemacht, wie man den menschlichen Bediener von Software in den Entwicklungsprozess einbezieht und wie wenig voraussagbar die Benutzungsprobleme, selbst bei sorgfältiger Entwicklungsarbeit sind. Etwas spezieller werden in der Veranstaltung Gestaltung von Webauftritten die Benutzungsprobleme im Internet behandelt. Durch die hochgradige Verbreitung des Internet in der Gesellschaft treten hier noch weit andere Probleme und Fragestellungen als bei „klassischer“ interaktiver Software auf. Die Studierenden sollen hier in Ergänzung zu den anderweitig vermittelten technischen Fähigkeiten die notwendigen Kenntnisse erlernen, um gut benutzbare und auf die Informationsbedürfnisse des Menschen abgestimmte Webseiten entwickeln zu können.

Modulzugehörigkeit

Pflichtmodul im Gebiet Mensch-Maschine-Wechselwirkung.

Modus

- Leistungspunkte : 3+3
- SWS 2+1, 2+1
- Häufigkeit des Angebotes: Jedes Semester werden mindestens 2 Veranstaltungen des Katalogs angeboten.

Prüfungsmodalitäten

Je nach Teilnehmerzahl mündliche Prüfung oder Klausur

Modulverantwortliche(r)
Szwilius

II.5 Gebietsübergreifend

II.5.1 Informatik im Kontext

Rolle der Veranstaltung im Studiengang

Der Einsatz von Informations- und Kommunikationstechnik hat Auswirkungen, die nicht immer nur positiv zu bewerten sind, er beinhaltet Risiken und negative Folgen. Um diese „beherrschbar“ zu machen und eventuell auftretende negative Folgen „abfangen“ zu können, brauchen wir Orientierungshilfen, die gewissermaßen Ordnung in den Umgang mit den Systemen bringen. Die in der Vorlesung „Ordnungsmäßigkeit der Systeme“ behandelten Ordnungsmäßigkeiten sind teilweise durch Gesetze untermauert, manchmal aber nur als ethische Richtlinien vorhanden.

In der Vorlesung werden systematisch die Ordnungsmäßigkeiten behandelt, die Datenschutz, Urheberschutz, Arbeitsschutz, Ethik und ähnliche soziale/rechtliche Themen für die Gesellschaft diskutierbar, einsatzfähig, eben „handhabbar“ machen. Eine Fülle von Fallstudien beschreibt die praktische Anwendung der Ordnungsmäßigkeiten und ist gleichzeitig eine Grundlage für die diese Vorlesung begleitenden Diskussionen. Diese Vorlesung behält die Sicht der Informatiker (nicht der Juristen) auf die genannten Themen und ist auch für zukünftige Führungskräfte von besonderem Interesse.

Die Veranstaltung zur Ordnungsmäßigkeit ist dem Gebiet Mensch-Maschine-Wechselwirkung (MMWW) zuzuordnen; das Proseminar kann frei aus jedem Gebiet gewählt werden, weswegen das gesamte Modul als gebietsübergreifend charakterisiert werden kann.

Inhaltliche Gliederung des Moduls

Das Modul umfasst die Veranstaltung „Ordnungsmäßigkeit der Systeme“ im Umfang von 2 Leistungspunkten und ein Proseminar (3 Leistungspunkte), das aus einem der vier Gebiete frei wählbar ist.

Inhaltliche Verwendbarkeit

Die in diesem Modul erworbenen Kenntnisse und Kompetenzen gehören zum allgemeinen Rüstzeug der Absolventen und werden entsprechend stark von der Industrie auch gefordert. Neben der Vertiefung und Einbettung tradierter Informatikinhalte erhalten die Studierenden einen Einblick in Themenfelder, die für ein Verständnis der Berufspraxis wichtig sind. Zugleich sind die erworbenen Schlüsselqualifikationen auf viele Lebens- und Arbeitsbereiche übertragbar. Sie bieten zugleich Anknüpfungspunkte für eine interdisziplinäre Arbeit und bereichern dadurch das Ausbildungsrepertoire der Informatik.

Voraussetzungen und Vorkenntnisse

Soweit nicht einzelne Proseminare entsprechende Voraussetzungen erfordern werden für dieses Modul keine spezifischen Kenntnisse oder Kompetenzen vorausgesetzt.

Lernziele

Die Veranstaltung soll den Studierenden helfen, ihre Arbeit und technischen Systeme im Kontext zu betrachten. Damit sollen zum einen Schlüsselqualifikationen erworben werden, die in einer durch digitale Medien geprägten Lebenswelt einen entscheidenden Erfolgsfaktor darstellen. Zum anderen soll ein vertieftes Verständnis der Berufspraxis sowohl die eigene Orientierung verbessern als auch ein Grundverständnis für die Rolle von Informatikern in der Gesellschaft schaffen. Dieses Wissen soll zugleich die Anschlussfähigkeit informatischer Kenntnisse an andere interdisziplinäre Bereiche sichern. Wissen und Kompetenzen aus dieser Veranstaltung vermitteln insofern wichtige Grundlagen für die Wahrnehmung der Verantwortung des Informatikers.

Modulzugehörigkeit

- Gebietsübergreifendes Pflichtmodul

Modus

- Leistungspunkte: 2 (aus Vorlesung V1+Ü1) + 3 (aus Proseminar S2)
- SWS: 2+2
- Häufigkeit des Angebotes: Die Pflichtveranstaltung wird jedes Wintersemester angeboten. Die Proseminare je nach Angebot der einzelnen Bereiche.

Prüfungsmodalitäten

Referate mit schriftlicher Ausarbeitung in den Proseminaren und eine Klausur in der Pflichtveranstaltung.

Modulverantwortliche(r)

Domik

III. Module im Masterstudiengang

III.1 Gebiet Softwaretechnik und Informationssysteme

III.1.1 Modellbasierte Softwareentwicklung

Rolle im Studiengang Informatik

Bei der modellbasierten Softwareentwicklung ist das Modell, die abstrakte Beschreibung des zu entwerfenden Systems, der Kern. Modelle entstehen in frühen Entwurfsphasen und auf unterschiedlichen Abstraktionsebenen, die unterschiedlich detailliert das zu entwerfende System beschreiben. Modelltransformationen überführen die Modelle unterschiedlicher Ebenen ineinander. Eine solche Transformation ist auch die letztendliche Codegenerierung. Ein modellbasierter Softwareentwicklungsprozess unterstützt die grundlegenden Konzepte Abstraktion und Strukturierung bereits im Entwurf. Er ermöglicht ferner eine Analyse in frühen Phasen des Entwurfs und dient damit der Qualitätsverbesserung.

Inhaltliche Gliederung der Veranstaltung

Der Modul besteht aus zwei weiterführenden Veranstaltungen aus folgendem Katalog

- Generating Software from Specifications
- Software Quality Assurance
- Web-Engineering
- Modelchecking
- Deductive Verification
- Software Safety

Die Vorlesungen sind wie folgt gegliedert:

- Generating Software from Specifications (in Englisch):
siehe Modul Sprachen und Programmiermethoden
- Software Quality Assurance (in Englisch):
 1. Standards (ISO 9126, CMM-I, SPICE, ISTQB, ...)
 2. Constructive Approaches (Domain-specific Languages, Meta-Modelling, Architectural Styles, Patterns, MDA, ...)
 3. Analytical Approaches (Reviews, Inspections, Black-Box, White-Box-Testing, Model-based Testing,...)
- Web Engineering (in Englisch):
 1. Web Applications
 - Categories / Characteristics
 - Modeling Approaches (WebML, UWE, ...)
 - Web Technologies (XML, CGI, JSP, JSF, PHP, AJAX, ...)
 2. Web Services
 - Standards (WSDL, SOAP, UDDI)
 - Visual Contracts
 3. Service-Oriented Architectures (SOA)
 - Concepts, Notions
 - Development Methods

- Modelchecking (in Englisch) :
siehe Modul Analytische Methoden des Software Engineering
- Deductive Verification (in Englisch):
siehe Modul Analytische Methoden des Software Engineering
- Software Safety (in Englisch):
siehe Modul Konstruktive Methoden des Software Engineering

Inhaltliche Verwendbarkeit

Dieses Modul vermittelt Kenntnisse und Fähigkeiten, die einerseits der Qualitätssicherung in einem modellbasierten Entwurf dienen, und andererseits Entwurfstechniken selber betreffen. Insbesondere sollen sie die Studierenden in die Lage versetzen, qualitätssichernde Maßnahmen selber auswählen und anwenden zu können. Die erworbenen Kenntnisse können daher überall dort im Studium und Beruf eingesetzt werden, wo es um den Entwurf von qualitativ hochwertiger Software geht.

Voraussetzungen und Vorkenntnisse

Grundlegende Kenntnisse im Softwareentwurf, wie sie etwa in der Vorlesung „Softwaretechnik“ und „Modellbasierter Softwareentwurf“ vermittelt werden und/oder Kenntnisse in der Modellierung mit formalen Methoden, wie sie in den Veranstaltungen „Modellierung“ und „Softwareentwurf mit formalen Methoden“ gelehrt werden.

Lernziele der Veranstaltung

Studierende lernen

Vermittlung von Faktenwissen

- Qualitätssicherungsverfahren im Softwareentwurf
- Entwurfstechniken für Webanwendungen
- Entwurfsverfahren für sicherheitskritische Systeme

Vermittlung von methodischem Wissen

- Problembezogene Auswahl und Einsatz von Sprachen und Entwicklungsmethoden
- Umgang mit industriellen Standards
- Techniken einer modellbasierten Softwareentwicklung

Vermittlung von Transferkompetenz

- die erworbenen Kompetenzen und Fertigkeiten auch in anderen Anwendungsbereichen als den im Modul vorkommenden

Vermittlung von normativ-bewertenden Kompetenzen

- Bewertung und Verbesserung von vorliegenden Entwurfsverfahren
- Die Eignung verschiedener Entwurfsverfahren und Qualitätssicherungsverfahren für einen Anwendungsbereich bewerten können

Schlüsselqualifikationen

- Kooperations- und Teamfähigkeit in den Präsenzübungen
- Strategien des Wissenserwerbs: Kombination aus Vorlesung, Vor- und Nachbereitung am Vorlesungsmaterial, Präsenzübungen mit betreuter Gruppenarbeit, Hausaufgaben.

Modulzugehörigkeit

Wahlpflichtmodul im Gebiet Softwaretechnik und Informationssysteme des Masterstudiums Informatik.

Modus

- Leistungspunkte pro Modul (Workload) : 8
- SWS: je zweimal V2 + Ü1
- Häufigkeit des Angebotes: jährlich
- Dauer: 1-2 Semester (je nach gewählter Katalog-Veranstaltung)

Methodische Umsetzung

- Methoden und Techniken werden an typischen Beispielen eingeführt und erläutert.
- Konzepte werden in Präsenzübungen in Kleingruppen vertieft.

Organisationsformen / Medieneinsatz / Literaturangaben

- Vorlesung mit Folienpräsentation
- Präsenzübungen in Kleingruppen
- erwartete Aktivitäten der Studierenden: Mitarbeit bei Präsenzübungen, Hausaufgaben, Vor- und Nacharbeit der Vorlesungen
- Vorlesungsmaterial im Web

Prüfungsmodalitäten

mündliche Prüfung falls weniger als 40 Teilnehmer, sonst Klausur in jeder Katalogveranstaltung

Modulverantwortlicher

Engels

III.1.2 Sprachen und Programmiermethoden

Rolle im Studiengang Informatik

Sprachen spielen in der Softwaretechnik vielfältige und wichtige Rollen. Als Programmiersprachen sind sie Ausdrucksmittel für die Programmentwicklung und dabei auf eine bestimmte Programmiermethode zugeschnitten. Als Spezifikationssprachen dienen sie zur Formulierung von Aufgabenbeschreibungen im Allgemeinen oder sind für bestimmte Anwendungsgebiete Beschreibungsmethoden speziell zugeschnitten. Nicht nur die methodisch fundierte Benutzung, sondern auch der Entwurf und die Implementierung solcher Sprachen durch Übersetzer oder Generatoren sind bedeutende Themengebiete der Softwaretechnik.

Dieser Modul vermittelt Kenntnisse und Fähigkeiten zum vertieften Verständnis, zur Spezifikation und zur Implementierung von Programmier- und Spezifikationssprachen. Er bietet die Weiterführung dieser Thematik wahlweise in zwei Gebieten der Sprachimplementierung oder der Programmiermethoden. Dieser Modul soll die Teilnehmer befähigen

- spezielle Verfahren zur Analyse oder Synthese von Programmen zu erlernen oder
- Programmiermethoden zu objektorientierten, parallelen, funktionalen, logischen oder Web-basierten Paradigmen anzuwenden oder
- Spezifikationssprachen für anwendungsspezifische Software-Generatoren zu entwerfen und zu implementieren.

Dabei wird auf Kenntnissen von Kalkülen zur Beschreibung von Spracheigenschaften und von Methoden zur Implementierung von Sprachen aufgebaut.

Inhaltliche Gliederung der Veranstaltung

Der Modul besteht aus zwei weiterführenden Veranstaltungen aus folgendem Katalog

- Generating Software from Specifications
- Compilation Methods
- Objektorientierte Programmierung
- Parallel Programming in Java
- Funktionale Programmierung
- Logische Programmierung
- Skriptsprachen
- Prolog mit Anwendungen

Die Vorlesungen sind wie folgt gegliedert

- Compilation Methods (in Englisch):
 1. Optimierung von Zwischen-Code
 2. Code-Generierung
 3. Registerzuteilung
 4. Code-Parallelisierung
- Generating Software from Specifications (in Englisch):
 1. Wiederverwendung und Generatoren
 2. Strukturierte Texte generieren

3. Bäume aufbauen
 4. Berechnungen in Bäumen
 5. Namen und Eigenschaften
 6. Sprachentwurf
 7. Projekte
- Objektorientierte Programmierung:
 1. Paradigmen zum Einsatz von Vererbung
 2. Separater Entwurf durch Entwurfsmuster
 3. Bibliotheken und Programmgerüste
 4. Entwurfsfehler
 5. Jenseits von Java
 - Parallel Programming in Java (in Englisch):
 1. Monitore und ihre systematische Entwicklung
 2. Barrieren: Anwendung und Implementierung
 3. Schleifenparallelisierung
 4. Programmierung mit asynchronen Botschaften
 5. Programmierung mit synchronen Botschaften
 - Funktionale Programmierung:
 1. Rekursionsparadigmen
 2. Funktionsschemata
 3. Typstrukturen
 4. Funktionen als Daten
 5. Datenströme und Lazy Evaluation
 6. Fixpunkte, Funktionsalgebra
 - Skriptsprachen:
 1. Statische und dynamische Web-Anwendungen, HTML
 2. Client-seitiges Skripting mit JavaScript
 3. Server-seitiges Skripting mit Perl
 4. Server-seitiges Skripting mit PH
 - Prolog mit Anwendungen in Textverstehen und Interpreterbau
(siehe Veranstaltungsbeschreibung im Modul Datenbanken und Informationssysteme)

Inhaltliche Verwendbarkeit

Die in diesem Modul erworbenen Kenntnisse und Fähigkeiten können überall im Studium und Beruf eingesetzt werden, wo ein vertieftes Verständnis von Sprachen zur Programmierung oder zur Spezifikation benötigt wird. Dabei ist die Veranstaltung Generating Software from Specifications stärker auf die Entwicklung sprachbasierter Werkzeuge ausgerichtet, während die anderen Veranstaltungen Methoden zum Einsatz von Sprachen vermitteln.

Voraussetzungen und Vorkenntnisse

Grundlegende Kenntnisse in einer zur Software-Entwicklung geeigneten Sprache sowie eigene praktische Erfahrung damit in der Programmentwicklung, wie sie die Veranstaltungen GP1, GP2 sowie das Softwaretechnikpraktikum vermitteln, Verständnis von allgemeinen Spracheigenschaften und von nicht-imperativen Programmierparadigmen wie es in GPS vermittelt wird, Kenntnisse von grundlegenden Methoden zur Spezifikation und Implementierung von Spracheigenschaften wie sie in Programmiersprachen und Übersetzer (PSÜ/PLaC) vermittelt werden.

Lernziele

Die Studierenden sollen ...

Vermittlung von Faktenwissen

- fortgeschrittene Techniken zur Implementierung von Sprachen verstehen
- Sprachkonstrukte für spezielle Programmierparadigmen und Spezifikationskalküle erlernen

Vermittlung von methodischem Wissen

- Generatoren und Standardlösungen zur Sprachimplementierung anwenden
- Methoden bestimmter Programmierparadigmen systematisch anwenden

Vermittlung von Transferkompetenz

- Sprachen für neue Anwendungsaufgaben spezifizieren und mit Generatoren implementieren
- Programmiermethoden auf zukünftige Sprachen übertragen

Vermittlung von normativ-bewertender Kompetenz

- die Klarheit und Problemnähe funktionaler Programm- und Datenformulierungen erkennen
- den Wert systematischer Methoden der Programmentwicklung erkennen

Schlüsselqualifikationen

- Kooperations- und Teamfähigkeit in den Präsenzübungen
- Strategien des Wissenserwerbs: Kombination aus Vorlesung, Vor- und Nachbereitung am Vorlesungsmaterial, Präsenzübungen mit betreuter Gruppenarbeit, Hausaufgaben.

Modulzugehörigkeit

Wahlmodul im Gebiet Softwaretechnik

Modus

Leistungspunkte: 4+4 ECTS (2 Katalogveranstaltungen)

SWS: 2+1, 2+1

Häufigkeit: 2-3 Katalogveranstaltungen pro Jahr im WS und SS

Methodische Umsetzung

- Methoden und Techniken werden an typischen Beispielen eingeführt, erläutert und in den Übungen praktisch erprobt (alle Veranstaltungen),
- in den Übungen werden Projekte in Kleingruppen unter Anleitung durchgeführt

Organisationsform, Medieneinsatz, Literaturangaben

- Vorlesungen mit Folienpräsentation
- Präsenzübungen in Kleingruppen
- einige Übungsstunden unter Anleitung an Rechnern
- erwartete Aktivitäten der Studierenden: Mitarbeit bei Präsenzübungen, Hausaufgaben, Vor- und Nacharbeit der Vorlesungen
- Web-basiertes Vorlesungsmaterial

Prüfungsmodalitäten

mündliche Prüfung falls weniger als 60 Teilnehmer, sonst Klausur in jeder Katalogveranstaltung

Modulverantwortlicher

Kastens

III.1.3 Datenbanken und Informationssysteme

Rolle im Studiengang

Das Modul Datenbanken und Informationssysteme ist ein Wahlpflichtmodul des Masterstudiums Informatik im Bereich Software-Technik und Informationssysteme. Die erworbenen Kenntnisse und Fertigkeiten können Seminaren vertieft werden und bilden eine ideale Grundlage für Masterarbeiten.

Inhaltliche Gliederung der Veranstaltungen des Moduls

Der Modul besteht aus zwei der folgenden Veranstaltungen

- Datenbanken und Informationssysteme 2
- Prolog mit Anwendungen
- Skriptsprachen
- Propositional Proof Systems
- Web-Engineering

wovon mindestens eine der folgenden Veranstaltungen in das Modul eingebracht werden soll:

- Datenbanken und Informationssysteme 2
- Prolog mit Anwendungen

Diese Vorlesungen werden laufend den aktuellen Inhalten angepasst und sind zum Zeitpunkt des Entwurfs dieses Modulhandbuches wie folgt gegliedert:

- Datenbanken und Informationssysteme 2
 1. XML-Technologien für den effizienten Informationsaustausch im Web
 2. XML-Kompression und Algorithmen auf massiven komplexen Daten
 3. Suche in und Transformation von Datenströmen
 4. Zugriffskontrolle, Security, Privacy
 5. Mobile Datenbanken und Informationssysteme
- Prolog mit Anwendungen
 - Natürlichsprachlicher Zugang zu Informationssystemen (Frage-Antwortsysteme, usw.)
 - Anwendungen aus der Computerlinguistik (Textverstehen, Automatisches Übersetzen)
 - Inferenzmethoden (Constraint-Solver, Termersetzungssysteme)
 - Parser und Interpreterbau (Spracherweiterungen und Meta-Interpreter)
- Skriptsprachen
 - siehe Veranstaltungsbeschreibung im Modul Sprachen und Programmiermethoden
- Web-Engineering
 - siehe Veranstaltungsbeschreibung im Modul Modellbasierte Softwareentwicklung
- Propositional Proof Systems
 - siehe Veranstaltungsbeschreibung im Modul Wissensbasierte Systeme

Inhaltliche Verwendbarkeit

Datenbanken und Informationssysteme spielen eine zentrale Rolle in Unternehmen, weil ein Großteil des „Wissen“ der Unternehmen als Daten in Datenbanken effizient zugreifbar gespeichert wird bzw. in Web-basierten Informationssystemen zur Verfügung gestellt wird.

Darüber hinaus ist es entscheidend, große Mengen von Daten und Wissen, z.B. aus dem Web effizient und sicher auszutauschen, zu durchsuchen, zu transformieren und an die jeweiligen

Benutzerwünsche anpassen zu können. Deshalb befassen sich die Veranstaltungen dieses Moduls unter anderem mit den jeweils modernen Verfahren in diesen Bereichen.

Zum Zeitpunkt des Verfassens dieser Modul-Beschreibung sind dies Algorithmen und Verfahren für Datenaustauschformate (u.a. komprimiertes XML), mobile und service-orientierte Architekturen. Zudem geht es um die Anwendung von Inferenz-Techniken in Informationssystemen, Security und Privacy, sowie um natürlich-sprachlichen Zugang zu Informationssystemen.

Die erworbenen Kenntnisse und Fertigkeiten werden in vielen Unternehmen in der Praxis angewandt und sind deshalb über das Studium hinaus nützlich.

Voraussetzungen und Vorkenntnisse

Die Inhalte der Veranstaltungen “Datenbankgrundlagen“ und “Datenbanken und Informationssysteme 1“ sowie solide Programmierkenntnisse in Java, wie sie in den Übungen zur Vorlesung Grundlagen der Programmierung gelehrt werden, werden vorausgesetzt.

Lernziele der Veranstaltung

Studierende lernen

Vermittlung von Faktenwissen

- Grundkonzepte, Architektur und Aufbau von Datenbanksystemen und Informationssystemen kennen

Vermittlung von methodischem Wissen

in Kleingruppen-Präsenz-Übungen:

- Systemkomponenten in Datenbanksystemen und Informationssystemen (z.B. Anfrageoptimierung) richtig bzw. sinnvoll zu nutzen
- beliebige Anfragen und Schreiboperationen in beliebigen Datenmodellen zu formulieren

in praktischen Übungen am Rechner:

- eigene Datenbanken und Informationssysteme aufzubauen
- Schnittstellen und Datenquellen eines Web-Informationssystems zu gestalten und zu ändern
- den sinnvollen Umgang mit wesentlichen in der Industrie benutzten Standards, z.B. SQL/XML, XPath, XSLT, DOM, SAX, XQuery, XML-Schema, Webservices, etc.

Vermittlung von Transferkompetenz

- die erworbenen Kompetenzen und Fertigkeiten auf andere Datenmodelle oder Datenstrukturen und auf andere Datenbanksysteme, Web-Informationssysteme, andere Netzwerke und andere Software-Architekturen zu übertragen

Vermittlung von normativ-bewertenden Kompetenzen

- die Eignung verschiedener Datenmodelle (relational, OO, XML) und verschiedener Softwareentwicklungsmodelle (Regel-basiert, Grammatik-basiert, etc.) für verschiedenen Anwendungen beurteilen können
- Einarbeitungszeiten in Datenbank- und Webtechnologien sowie Entwicklungszeiten für Informationssysteme abschätzen können

Schlüsselqualifikationen

Studierende lernen in praktischen Übungen den Umgang mit wesentlichen in der Industrie benutzten Standards, z.B. SQL/XML, XPath, XSLT, XQuery, XML Schema, SOAP. Sie erwerben durch eigene Rechnerübungen mit diesen Standards und den zugehörigen Technologien zudem die notwendige Praxis zur selbständigen Erschließung einer Vielzahl ganz ähnlicher Datenbank- und Internet-Technologien.

Modulzugehörigkeit

Wahlpflichtmodul im Gebiet Softwaretechnik und Informationssysteme des Masterstudiums Informatik.

Modus

- Leistungspunkte pro Modul (Workload) : 8
- Leistungspunkte der Veranstaltung : 8
- SWS: je zweimal V2 + Ü1
- Häufigkeit des Angebotes: jährlich
- Dauer: 1-2 Semester (je nach gewählter Katalog-Veranstaltung)

Methodische Umsetzung

- Die theoretischen Konzepte werden in Präsenzübungen in Kleingruppen vertieft.
- Die Vorlesung wird unterbrochen durch Übungen an praktischen Beispielen.
- Wesentliche Ergebnisse werden so in der Vorlesung entwickelt, dass sie quasi „neu entdeckt“ bzw. „erfunden“ werden können.
- Die praktischen Fertigkeiten werden erlernt anhand von Übungen am Rechner, bei denen ausgehend von lauffähigen Demoprogrammen aus der Vorlesung eigene Informationssysteme zu entwickeln sind.
- Die Vorlesung (insbesondere DBIS 2) orientiert sich an der aktuellen Forschung

Organisationsformen / Medieneinsatz / Literaturangaben

- Vorlesung mit Tafel bzw. Folienpräsentation und aktuellen wissenschaftlichen Publikationen bzw. minimierten lauffähigen Beispielprogrammen am Rechner
- Die Übungen sind je nach Veranstaltung unterschiedlich organisiert, z.B. für DBIS 2: ca. 40% als Präsenzübungen in Kleingruppen mit Übungsblättern und Hausaufgaben, ca. 60% praktische Übungen am Rechner.
- erwartete Aktivitäten der Studierenden: Mitarbeit bei Präsenzübungen, Hausaufgaben, Entwicklung eigener Software am Rechner
- Aktuelle Literatur wird in den Lehrveranstaltungen bekannt gegeben.

Prüfungsmodalitäten

Bei DBIS 2: Klausur, sonst abhängig von der gewählten Veranstaltung

Modulverantwortliche(r)

Böttcher

III.1.4 Wissensbasierte Systeme

Rolle im Studiengang MSc Informatik

Das Modul Wissensbasierte Systeme enthält Vorlesungen aus dem Bereich der intelligenten Datenverarbeitung zum Lösen wissensintensiver Aufgaben. Inhaltlich werden unterschiedliche Wissensrepräsentationen und Verfahren zu ihrer Verarbeitung angesprochen. Sie sollen die Hörer in die Lage versetzen, schwer strukturierbare Probleme zu modellieren und effizienten Lösungsverfahren zugänglich zu machen.

Wissensbasierte Verfahren sind keine „Stand-alone-Technologie“, sondern dazu gedacht, in Kombination mit den klassischen Gebieten der Informatik bzw. für Anwendungen aus den Ingenieurwissenschaften oder der Betriebswirtschaftslehre eine neue Qualität der Problemlösung zu erzielen.

Inhaltliche Gliederung

In diesem Modul sind bisher die folgenden Veranstaltungen geplant:

- Verteiltes Problemlösen
- Maschinelles Lernen
- Propositional Proof Systems
- Heuristische Suchverfahren
- Prolog mit Anwendungen im Textverstehen und Interpreterbau
- Theorembeweisen

Die genannten Veranstaltungen sind so aufgebaut, dass sie in sich geschlossene Einheiten bilden und daher größtenteils unabhängig voneinander gehört werden können.

Inhaltliche Verwendbarkeit

Die in diesem Modul erworbenen Kenntnisse und Fähigkeiten können in der beruflichen Praxis dort eingesetzt werden, wo keine Standardverfahren zur Problemlösung existieren, die Aspekte Unsicherheit und Vagheit berücksichtigt werden müssen, menschliches Problemlöseverhalten nachgebildet werden soll etc.

Voraussetzungen und Vorkenntnisse

Die Hörer sollten Interesse an Algorithmen, abstraktem Modellieren haben und gute Kenntnisse und praktische Erfahrung in einer Programmiersprache besitzen.

Lernziele

Die Studenten sollen in der Lage sein, eine Auswahl von Problemlösungstechniken sicher zu beherrschen und Probleme selbständig zu analysieren, den Grad der möglichen Automatisierbarkeit realistisch abzuschätzen und auf Basis ihrer Analyse eine adäquate Lösung zu entwickeln.

Insbesondere sollen die Studierenden ...

Vermittlung von Faktenwissen

- verschiedene Fragestellungen aus dem Bereich der Wissensbasierten Systeme kennenlernen,
- Techniken zur Modellierung erlernen,
- die passenden Problemlösungsmethoden mit ihren Vorteilen und Beschränkungen begreifen,

Vermittlung von methodischem Wissen

- Problemstellungen mit vorgestellten Formalismen modellieren, mit den zugehörigen Methoden bearbeiten und Resultate interpretieren,

Vermittlung von Transferkompetenz

- Anwendungsmöglichkeiten für wissensbasierte Technologien in neuen Problemstellungen entdecken,
- weitere Techniken selbständig erarbeiten,

Vermittlung von normativ-bewertender Kompetenz

- die Eignung verschiedener Formalismen zur Wissensrepräsentation in konkreten Problemstellungen beurteilen,
- die Anwendbarkeit von Problemlösungsmethoden abschätzen.

Schlüsselqualifikationen

- Kooperations- und Teamfähigkeit in den Präsenzübungen
- Strategien des Wissenserwerbs: Kombination aus Vorlesung, Vor- und Nachbereitung am Vorlesungsmaterial, Präsenzübungen mit betreuter Gruppenarbeit, Hausaufgaben.

Modulzugehörigkeit

Wahlpflichtmodul im Gebiet Softwaretechnik und Informationssysteme

Modus

Leistungspunkte: 4+4 ECTS (2 Katalogveranstaltungen)

SWS: 2+1, 2+1

Häufigkeit: 2-3 Katalogveranstaltungen pro Jahr im WS und SS

Methodische Umsetzung

- Methoden, Techniken und ihre Umsetzung werden in den Vorlesungen an typischen Beispielen eingeführt, erläutert und in den Übungen praktisch erprobt.
- Teilweise werden in den Übungen prototypische Implementierungen vorgenommen oder evaluiert.

Organisationsformen / Medieneinsatz / Literaturangaben

- Vorlesungen mit Folienpräsentation
- Präsenzübungen in Kleingruppen
- Hausaufgaben, teilweise Musterlösungen in Übungen vorgestellt
- Erwartete Mitarbeit der Studierenden:
Mitarbeit bei Präsenzübungen, Hausaufgaben, Vor- und Nacharbeit der Vorlesungen
- Weiterführende Literatur und Lehrmaterialien im Web (Folien, Hausaufgaben)

Prüfungsmodalitäten

Mündliche Prüfung, bei sehr hohen Teilnehmerzahlen ggf. Klausur

Modulverantwortlicher

Kleine Büning

III.1.5 Analytische Methoden des Software Engineering

Rolle im Studiengang

Dieser Modul vertieft die im Bachelorstudiengang erworbenen Kenntnisse im Softwareentwurf auf dem Gebiet der analytischen Methoden. Analytische Methoden dienen der Qualitätssicherung der entworfenen Software, im Gegensatz zu konstruktiven Verfahren wird hierbei aber die Qualität nicht per Konstruktion sondern durch anschließende Analyse erreicht. Der Modul beschäftigt sich insbesondere mit dem Gebiet der mathematischen und formalen Grundlagen der Softwaretechnik, die zur Sicherung der Korrektheit von Software eingesetzt werden.

Dazu werden Konzepte und Methoden der Semantik von Sprachen, der semiautomatischen und automatischen Verifikationstechniken sowie der klassischen Qualitätssicherung erarbeitet. Die Beherrschung dieser Methoden verhilft einerseits zu einem besseren Verständnis der Konzepte der Softwaretechnik und ermöglicht andererseits die wissenschaftliche Untersuchung, Verbesserung und Fundierung neuer Softwaretechniken.

Je nach dem gewählten Schwerpunkt innerhalb dieses Moduls sollen die Studierenden nach Absolvieren dieses Moduls in der Lage sein,

- verschiedene analytische Qualitätssicherungsverfahren bewerten, vergleichen und in einem Anwendungsgebiet einsetzen zu können und über den Einsatz eines passenden Verfahrens entscheiden zu können,
- Methoden des semi-automatischen und automatischen Korrektheitsnachweises zu benutzen.

Dieser Modul ist ein Wahlpflichtmodul im Bereich „Softwaretechnik und Informationssysteme“.

Inhaltliche Gliederung der Veranstaltung

Zum Absolvieren dieses Moduls müssen zwei Veranstaltungen aus dem folgenden Katalog ausgewählt werden:

- Propositional Proof Systems
- Theorembeweiser
- Modelchecking
- Deductive Verification
- Compilation Methods
- Software Quality Assurance

Diese Veranstaltungen sind wie folgt gegliedert:

- Propositional Proof Systems:
(siehe Veranstaltungsbeschreibung im Modul Wissensbasierte Systeme)
- Theorembeweiser:
(siehe Veranstaltungsbeschreibung im Modul Wissensbasierte Systeme)
- Modelchecking (in Englisch)
 1. Modellierung und Eigenschaften reaktiver Systeme
 2. Temporale Logiken: LTL und CTL
 3. Fairness
 4. LTL Modelchecking mit Automaten
 5. BDDs, Symbolisches Modelchecking
 6. Reduktions- und Abstraktionstechniken, Bisimulation

- Deductive Verification (in Englisch)
 1. Semantik einer Programmiersprache
 2. Beweissysteme
 3. Partielle/Totale Korrektheit
 4. Sicherheit und Lebendigkeit
 5. Vollständigkeit und Korrektheit von Beweissystemen
- Compilation Methods (in Englisch)
(siehe Veranstaltungsbeschreibung im Modul Sprachen und Programmiermethoden)
- Software Quality Assurance (in Englisch)
(siehe Veranstaltungsbeschreibung im Modul Modellbasierte Softwareentwicklung)

Inhaltliche Verwendbarkeit

In diesem Modul werden Kenntnisse und Fähigkeiten vermittelt, die das Verstehen, Formulieren, Formalisieren und Argumentieren über komplexe Zusammenhänge mit Hilfe formaler und mathematischer Modelle ermöglichen. Diese können insbesondere bei der Analyse von sicherheitskritischen Systemen und zuverlässiger Software eingesetzt werden.

Voraussetzungen und Vorkenntnisse

Die Voraussetzungen zum Absolvieren dieses Moduls sind die Fähigkeit zur Modellierung und Formalisierung von Sachverhalten mit Hilfe von mathematischen und informatischen Notationen, wie sie im Modul Modellierung sowie Softwareentwurf vermittelt werden. Außerdem wird die Beherrschung wenigstens einer Programmiersprache vorausgesetzt, wie sie in dem Modul Grundlagen der Programmierung vermittelt wird. Darüber hinaus sollten die Studierenden die grundlegenden Techniken des formalen Definierens und Schließens beherrschen, wie es in den Lehrveranstaltungen „Softwareentwurf mit formalen Methoden“, „Grundlagen Wissensbasierter Systeme“ sowie in „Mathematik“ vermittelt wird.

Lernziele

Die Studierenden sollen ...

Vermittlung von Faktenwissen

- die Techniken und mathematischen Strukturen zur Formalisierung der Semantik von Programmier- und Modellierungssprachen kennen,
- verschiedene analytische Qualitätssicherungsverfahren, angefangen von statischen Analysen über Testen bis zur Verifikation kennen,
- die Unterschiede und Vor- und Nachteile der verschiedenen Techniken kennen,

Vermittlung von methodischem Wissen

- Systeme formal modellieren und deren Eigenschaften formulieren können,
- einschätzen können, welche Techniken und Verfahren zu welchem Zweck eingesetzt werden sollten,
- Mathematik und Logik korrekt und zweckmäßig einsetzen können,
- bestehende Softwaresysteme auf gewünschte Qualitätseigenschaften untersuchen können.

Vermittlung von Transferkompetenz

- selbständig mathematische Modelle aufstellen und über deren Eigenschaften argumentieren können,
- sich neue Konzepte und Techniken aneignen und sie bewerten und ggf. anpassen können.

Vermittlung von normativ-bewertender Kompetenz

- die Bedeutung semantischer Fundierung von Techniken erkennen,
- ein Bewusstsein dafür besitzen, dass die Auswahl geeigneter Analyseverfahren eine genaue Analyse der Charakteristik des spezifischen Anwendungsgebietes erfordert.

Schlüsselqualifikationen

- Kommunikations- und Teamfähigkeit in den Übungen
- Strategien des Wissenserwerbs:
Kombination aus Vorlesung, Vor- und Nachbereitung von Vorlesungsmaterial und ergänzender Literatur, Gruppenarbeit, Hausaufgaben.
- Bewertung und Hinterfragung neuer Konzepte
- Querverbindungen und Bezüge zwischen ähnlichen Konzepten entdecken und herstellen können

Modulzugehörigkeit

Wahlpflichtmodul im Gebiet „Softwaretechnik und Informationssysteme“

Modus

Leistungspunkte: 4+4 ECTS (4 pro Veranstaltung)

SWS: 2V+1Ü, 2V+1Ü

Häufigkeit: Es wird pro Semester mindestens eine Veranstaltung aus dem Modul angeboten.

Methodische Umsetzung

- Methoden und Techniken werden an typischen Beispielen eingeführt und diskutiert
- in den Übungen werden sie praktisch erprobt; teilweise werden Computerwerkzeuge eingesetzt

Organisationsform, Medieneinsatz, Literaturangaben

- Vorlesungen mit Folienpräsentation oder Tafelanschrift
- ergänzende Materialien zur Vorlesung im Internet
- in den Übungen wird die Lösung der Aufgaben gemeinsam erarbeitet
- erwartete Aktivitäten der Studierenden:
Mitarbeit bei der Erarbeitung der Lösung in den Übungen, Hausaufgaben, Vor- und Nacharbeit der Vorlesungen

Prüfungsmodalitäten

Die Veranstaltungen dieses Moduls werden jeweils einzeln geprüft (je nach Anzahl der Teilnehmer mündlich oder schriftlich).

Modulverantwortlicher

Wehrheim

III.1.6 Konstruktive Methoden des Software Engineering

Rolle im Studiengang Informatik

Dieser Modul vertieft die im Bachelorstudiengang erworbenen Kenntnisse im Softwareentwurf auf dem Gebiet der konstruktiven Methoden. Konstruktive Methoden sollen eine hohe Softwarequalität direkt durch den Softwareengineering-Prozess sicherstellen (im Gegensatz zu analytischen Methoden, die Qualität durch eine nach der Entwicklung stattfindende Analyse erreichen).

Nach dem Absolvieren dieses Moduls sollen die Studierenden in der Lage sein verschiedene konstruktive Qualitätssicherungsverfahren bewerten, vergleichen und einsetzen zu können. Dabei sollen sie insbesondere in der Lage sein, ein zum jeweiligen Einsatzgebiet passendes Verfahren auszuwählen und anzuwenden.

Inhaltliche Gliederung der Veranstaltung

Der Modul besteht aus zwei weiterführenden Veranstaltungen aus folgendem Katalog

- Web-Engineering
- Generating Software from Specifications
- Parallel Programming in Java
- Objektorientierte Programmierung
- Software Safety
- Software Quality Assurance
- Prolog mit Anwendungen
- Skriptsprachen
- Compilation Methods
- Funktionale Programmierung

Die Vorlesungen sind wie folgt gegliedert:

Web-Engineering:

siehe Modul Modellbasierte Softwareentwicklung

Generating Software from Specifications (in Englisch):

siehe Modul Sprachen und Programmiermethoden

Parallel Programming in Java:

siehe Modul Sprachen und Programmiermethoden

Objektorientierte Programmierung:

siehe Modul Sprachen und Programmiermethoden

Funktionale Programmierung:

siehe Modul Sprachen und Programmiermethoden

Software Safety :

1. Eigenschaften sicherheitskritischer Systeme
2. Modellbasierte Methoden und domänenspezifische Architekturen für sicherheitskritische Systeme
3. Gefahrenanalyse und Fehlertoleranz
4. Entwurf zuverlässiger Software

Software Quality Assurance (in Englisch):

siehe Modul Modellbasierte Softwareentwicklung

Prolog mit Anwendungen

siehe Veranstaltungsbeschreibung im Modul Datenbanken und Informationssysteme

Skriptsprachen

siehe Modul Sprachen und Programmiermethoden

Compilation Methods

siehe Modul Sprachen und Programmiermethoden

Inhaltliche Verwendbarkeit

Die Inhalte dieses Moduls können in der Praxis bei Entwurf und Umsetzung komplexer Softwaresysteme eingesetzt werden. Insbesondere geht es dabei um die Kenntnis der verschiedenen Paradigmen und die Auswahl des jeweils geeigneten Verfahrens für ein bestimmtes System in einer vorgegebenen Domäne. Dabei wird besonderes Augenmerk auf software-intensive und sicherheitskritische Systeme gelegt.

Voraussetzungen und Vorkenntnisse

Grundlegende Kenntnisse im Softwareentwurf, wie sie etwa in der Vorlesung „Softwaretechnik“ und „Modellbasierter Softwareentwurf“ vermittelt werden. Ferner grundlegende Kenntnisse von Programmiersprachen und Fähigkeiten in der Programmierung, wie sie etwa durch Besuch der Veranstaltungen „Grundlagen der Programmierung 1 und 2“ sowie „Grundlagen Programmiersprachen“ erworben werden können.

Lernziele der Veranstaltung

Studierende sollen ...

Vermittlung von Faktenwissen

- gängige Paradigmen der Softwareentwicklung kennenlernen.
- die Anwendbarkeit der Paradigmen in verschiedenen Kontexten verstehen.

Vermittlung von methodischem Wissen

- geeigneter Methoden bei der Erstellung und Wartung von Softwaresystemen einsetzen können.

Vermittlung von Transferkompetenz

- selbstständig komplexe Softwaresysteme unter Berücksichtigung der domänenspezifischen Anforderungen entwerfen können.
- sich neue Methoden des Softwareentwurfs aneignen und diese einsetzen können.

Vermittlung von normativ-bewertenden Kompetenzen

- die Auswirkungen von Designentscheidungen auf Softwarearchitekturen im Kontext eines Systems beurteilen können.
- die Eignung von Entwurfskonzepten für ein gegebenes System beurteilen können.
- Die Tragweite von Entscheidungen im Bereich sicherheitskritischer Systeme begreifen.

Schlüsselqualifikationen

- Kooperations- und Teamfähigkeit in den Präsenzübungen

- Strategien des Wissenserwerbs: Kombination aus Vorlesung, Vor- und Nachbereitung am Vorlesungsmaterial, Präsenzübungen mit betreuter Gruppenarbeit, Hausaufgaben.

Modulzugehörigkeit

Wahlpflichtmodul im Gebiet Softwaretechnik und Informationssysteme des Masterstudiums Informatik.

Modus

- Leistungspunkte pro Modul (Workload) : 8
- SWS: je zweimal V2 + Ü1
- Häufigkeit des Angebotes: jährlich
- Dauer: 1-2 Semester (je nach gewählter Katalog-Veranstaltung)

Methodische Umsetzung

- Methoden und Techniken werden an typischen Beispielen eingeführt und erläutert.
- Konzepte werden in Präsenzübungen in Kleingruppen vertieft.

Organisationsformen / Medieneinsatz / Literaturangaben

- Vorlesung mit Folienpräsentation
- Präsenzübungen in Kleingruppen
- erwartete Aktivitäten der Studierenden: Mitarbeit bei Präsenzübungen, Hausaufgaben, Vor- und Nacharbeit der Vorlesungen
- Vorlesungsmaterial im Web

Prüfungsmodalitäten

mündliche Prüfung falls weniger als 40 Teilnehmer, sonst Klausur in jeder Katalogveranstaltung

Modulverantwortlicher

Schäfer

III.2 Gebiet Modelle und Algorithmen

III.2.1 Algorithmen I

Rolle der Veranstaltung im Studiengang

Algorithmen bilden die Grundlage jeder Hardware und Software: Ein Schaltkreis setzt einen Algorithmus in Hardware um, ein Programm macht einen Algorithmus "für den Rechner verstehbar". Algorithmen spielen daher eine zentrale Rolle in der Informatik. Wesentliches Ziel des Algorithmenentwurfs ist die (Ressourcen-) Effizienz, d.h. die Entwicklung von Algorithmen, die ein gegebenes Problem möglichst schnell oder mit möglichst geringem Speicherbedarf lösen. Gegenstand dieses Moduls ist die Behandlung unterschiedlicher methodischer und anwendungsspezifischer algorithmischer Fragestellungen. Es werden z.B. Online Algorithmen, Approximation und Randomisierung angesprochen, und ihre Anwendungen in Algorithmen für Graphen, für Codierungsprobleme und geometrische Probleme vorgestellt.

Entsprechend der Breite und Bedeutung des Gebiets und seines Stellenwertes in der Paderborner Informatik bieten wir an, dieses Gebiet im Umfang zweier Module zu studieren.

Deshalb bieten wir mit Algorithmen I und Algorithmen II zwei Module mit identischem Veranstaltungskatalog an.

Inhaltliche Gliederung der Veranstaltung

Folgende Veranstaltungen werden angeboten:

- Approximationsalgorithmen
- Randomisierte Algorithmen
- Online Algorithmen
- Algorithmische Spieltheorie
- Algorithmische Codierungstheorie I
- Netzwerkfluss-Algorithmen
- Graphenalgorithmen
- Geometrische Algorithmen

Inhaltliche Verwendbarkeit

Für gegebene Probleme nicht nur irgendwelche, sondern ressourcenschonende, d.h. effiziente Algorithmen zu entwerfen, und Probleme bezüglich ihrer inhärenten Komplexität einzuschätzen, beschreibt für viele Teilgebiete (nicht nur) der Informatik eine wichtige Fähigkeit. Datenbanken und Informationssysteme, Computergrafik-Systeme und wissenschaftliches Rechnen sind wichtige Beispiele.

Voraussetzungen und Vorkenntnisse

Die wesentlichen Konzepte aus der Algorithmentheorie und Komplexitätstheorie wie sie etwa in den Vorlesungen *Einführung in Berechenbarkeit, Komplexität und Formale Sprachen* sowie *Effiziente Algorithmen* behandelt werden, werden vorausgesetzt.

Neben mathematischem Grundwissen, wie es im Grundstudium vermittelt wird, ist Interesse an kreativem Problemlösen mit mathematisch exakten Methoden notwendig.

Lernziele der Veranstaltung

Die wesentlichen Aufgaben, Konzepte und Verfahren der Algorithmenentwicklung und

-analyse sollen vermittelt werden. Studierende sollen am Ende in der Lage sein, Problem gemäß ihrer Komplexität einzuschätzen und adäquate algorithmische Techniken für ihre Lösung einzusetzen.

Modulzugehörigkeit

Wahlmodul im Gebiet Modelle und Algorithmen.

Modus

- Leistungspunkte : 4+4
- SWS 2+1,2+1
- Häufigkeit des Angebotes: Das Modul wird jedes Jahr angeboten.

Prüfungsmodalitäten

Je nach Teilnehmerzahl mündliche Prüfung oder Klausur

Modulverantwortlicher

Meyer auf der Heide

III.2.2 Algorithmen II

Identisch zu „Algorithmen I“.

III.2.3 Berechenbarkeit und Komplexität

Rolle der Veranstaltung im Studiengang

Im Zentrum dieses Moduls stehen die Frage nach den Grenzen der Berechenbarkeit und die Klassifizierung von Problemen bezüglich ihrer algorithmischen Komplexität. Als Maße für Komplexität werden insbesondere Laufzeit und Speicherbedarf, aber auch z.B. Parallelisierbarkeit herangezogen. Es beinhaltet den Nachweis sowohl der Nichtentscheidbarkeit z.B. der Arithmetik als auch die Untersuchung der Problem-inhärenten Komplexität, d.h. den Beweis unterer Komplexitätsschranken und den Komplexitätsvergleich von Problemen. Unter komplexitätstheoretischen Aspekten werden auch Formale Sprachen untersucht.

Die Grundlagen über Algorithmen und Komplexität werde ergänzt durch Methoden der algorithmischen Behandlung sehr komplexer Probleme, z.B. Approximationsalgorithmen.

Inhaltliche Gliederung der Veranstaltung

Folgende Veranstaltungen werden angeboten:

- Berechenbarkeit und Komplexität
- Formale Sprachen
- Approximationsalgorithmen

Inhaltliche Verwendbarkeit

Dieses Modul befähigt die Studierenden, die grundsätzlichen und die durch Ressourcenschranken gegebenen Grenzen der algorithmischen Lösbarkeit von Problem einzuschätzen und diese Fähigkeit auf konkrete Probleme anzuwenden. Diese Fähigkeit ist in Allen Gebieten des Bereichs MuA, aber auch überall da, wo Algorithmen für komplexe Problem entwickelt werden, nützlich.

Voraussetzungen und Vorkenntnisse

Die wesentlichen Konzepte aus der Komplexitätstheorie wie sie etwa in den Vorlesungen *Einführung in Berechenbarkeit, Komplexität und Formale Sprachen* sowie *Komplexitätstheorie* behandelt werden, werden vorausgesetzt.

Neben mathematischem Grundwissen, wie es im Grundstudium vermittelt wird, ist Interesse an kreativem Problemlösen mit mathematisch exakten Methoden notwendig.

Lernziele der Veranstaltung

Die wesentlichen Aufgaben, Konzepte und Verfahren der Berechenbarkeits- und Komplexitätstheorie sollen vermittelt werden. Studierende sollen am Ende in der Lage sein, Problem gemäß ihrer Komplexität einzuschätzen und adäquate algorithmische Techniken für ihre Lösung einzusetzen.

Modulzugehörigkeit

Wahlmodul im Gebiet Modelle und Algorithmen.

Modus

- Leistungspunkte : 4+4
- SWS 2+1,2+1
- Häufigkeit des Angebotes: Das Modul wird in Jahr angeboten.

Prüfungsmodalitäten

Je nach Teilnehmerzahl mündliche Prüfung oder Klausur

Modulverantwortlicher

Meyer auf der Heide

III.2.4 Algorithmen in Rechnernetzen

Rolle der Veranstaltung im Studiengang

Die Theorie der parallelen Algorithmen und Architekturen hat es in den letzten Jahren ermöglicht, massiv parallele Rechner mit tausend und mehr Prozessen zu bauen und effizient einzusetzen. Die großen Herausforderungen für die Informatik, wie z.B. die Wettervorhersage, Ozeansimulation, astrophysikalische Simulationen und Medikamentendesign, aber auch schwere Optimierungsprobleme erfordern den Einsatz massiv paralleler Supercomputer. Neben den Supercomputern ist der Einsatz von Parallelrechnern in Form von Mehrprozessor-PCs, oder Prozessorclustern heute schon Standard in vielen wissenschaftlichen, kommerziellen oder industriellen Anwendungen.

Das Internet, in seiner Gesamtheit ebenfalls ein Parallelrechner, wird heute schon als ein solcher genutzt, wenn z.B. Gridcomputing Anwendungen implementiert werden.

Die theoretische Informatik hat mit der Modellbildung paralleler Computer und mit der Entwicklung von effizienten Algorithmen für diese Modelle einen entscheidenden Schritt dazu beigetragen, dass Parallelrechner effizient in vielen Bereichen eingesetzt werden können, wo große Rechenleistungen erforderlich sind. Wesentliches Ziel der Veranstaltungen in diesem Modul ist es, anhand von analysierbaren parallelen Algorithmen und Architekturen ein allgemeines Verständnis für parallele Vorgänge zu erreichen.

Inhaltliche Gliederung der Veranstaltung

Der Modul umfasst sowohl Veranstaltungen, in denen effiziente Algorithmen zur Problemlösung mit Rechnernetzen vorgestellt werden, als auch Veranstaltungen, die Problemlösungen vorstellen, die eine effiziente Nutzung von Rechnernetzen ermöglichen.

Das Modul besteht aus den folgenden Veranstaltungen:

- Algorithmen für synchrone Rechnernetze
- Nichtkooperative Netze
- Ressourcenverwaltung in Rechnernetzen
- Algorithmische Grundlagen des Internets
- Peer-to-Peer Netzwerke
- Algorithmische Probleme in Funknetzwerken
- Kommunikation in Netzwerken

Inhaltliche Verwendbarkeit

Die in diesem Modul erworbenen Grundkenntnisse über parallele Algorithmen und Architekturen sind für jeden, der in der Forschung, im kommerziellen oder im industriellen Umfeld mit Parallelrechnern arbeitet, unerlässlich. Durch die immer noch stark wachsenden Anwendungsfelder für parallele Supercomputer, speziell im Bereich des wissenschaftlichen Rechnens, aber auch durch die wachsende Zahl von Multiprozessor-PCs oder PC-Clustern im kommerziellen oder industriellen Umfeld bilden die vermittelten Kenntnisse ein zukunftsträchtiges Wissen.

Neben diesen Formen von Parallelrechnern wird auch das Internet durch die in steigender Form angebotenen Servicedienste immer mehr als Parallelrechner genutzt. Auch für diese Wachstumsbranche werden die in diesem Modul vermittelten Kenntnisse von Bedeutung sein.

Voraussetzungen und Vorkenntnisse

Die wesentlichen Konzepte aus den Bereichen Algorithmen, Datenstrukturen, Berechenbarkeit und Komplexitätstheorie wie sie in den ersten vier Semestern vermittelt

werden werden vorausgesetzt. Kenntnisse über Algorithmen und ihre Analysen wie sie in der Vorlesung Effiziente Algorithmen vermittelt werden sind von Vorteil.

Lernziele der Veranstaltung

Die Veranstaltung soll den Studenten wichtige parallele algorithmische Techniken und Architekturen vorstellen. Das Ziel ist es dabei einerseits, einen Grundstock paralleler Algorithmen für immer wieder in Anwendungen vorkommende Probleme bereitzustellen, und andererseits, die Studenten in die Lage zu versetzen, effiziente parallele Algorithmen für neue Problemstellungen entwickeln und auf real existierenden Parallelrechnern -- in all ihren Formen vom Supercomputer bis zum Internet -- implementieren zu können.

Modulzugehörigkeit

Wahlmodul im Gebiet Modelle und Algorithmen.

Modus

- Leistungspunkte : 4+4
- SWS 2+1,2+1
- Häufigkeit des Angebotes: Eine Basisveranstaltung jedes WS, die weiterführenden Veranstaltungen jedes SS.

Prüfungsmodalitäten

Je nach Teilnehmerzahl mündliche Prüfung oder Klausur

Modulverantwortliche(r)

Monien

III.2.5 Codes und Kryptographie

Rolle der Veranstaltung im Studiengang

Datenkompression und Datenverschlüsselung spielen in der Datenübertragung eine große Rolle. Datenkompression ist notwendig, da immer größere Datenmengen gespeichert und verschickt werden. Datensicherung ist unerlässlich, wenn wichtige Daten auf unsicheren (nicht abhörsicheren) Kanälen verschickt werden. Dieses kann z.B. eine Kreditkartennummer sein, die über das Internet übertragen wird. In dem Modul Codes und Kryptographie werden wichtige Techniken der Datenkompression, Datensicherung und Datenverschlüsselung behandelt. In der Datenkompression werden vor allem die wichtigsten Methoden der verlustfreien Datenkompression behandelt. Im Bereich der Datensicherung wird die Theorie der fehlerkorrigierenden Codes eingeführt. Weiter werden wichtige fehlerkorrigierende Codes sowie ihre algorithmische Umsetzung untersucht. Die Veranstaltungen zur Datenverschlüsselung umfassen neben den wichtigsten Verschlüsselungsverfahren wie AES und RSA, auch Verfahren zur Datenintegrität und Authentizität. Es werden die wesentlichen Sicherheitskonzepte und Methoden der Kryptanalyse behandelt. Schließlich werden die wesentlichen mathematischen Grundlagen von Verfahren wie RSA und DSA untersucht.

Inhaltliche Gliederung der Veranstaltung

Veranstaltungen:

1. Kryptographie
2. Kryptographische Protokolle
3. Algorithmen in der Zahlentheorie
4. Algorithmische Codierungstheorie I
5. Algorithmische Codierungstheorie II

Der Modul umfasst sowohl Veranstaltungen aus dem Bereich der Codierungstheorie als auch aus dem Bereich der Kryptographie. Der Modul kann in drei unterschiedlichen Varianten studiert werden. Der Modul erlaubt eine Konzentration auf die Codierungstheorie. Dann werden die Veranstaltungen Algorithmische Codierungstheorie I und II gehört. Bei einer Konzentration auf den Bereich Kryptographie sollte die Veranstaltung Kryptographie sowie eine der Veranstaltung gehört werden, die nicht zur Codierungstheorie zählen. Die Veranstaltung Kryptographische Protokolle und Das kryptographische Verfahren RSA vertiefen dabei Kenntnisse in einzelne kryptographische Verfahren. Die Vorlesung Algorithmen in der Zahlentheorie vertieft mehr die mathematischen Grundlagen der Kryptographie. Es ist aber auch möglich Veranstaltungen aus der Codierungstheorie und der Kryptographie zu kombinieren. Eine sinnvolle Kombination sind hierbei die Veranstaltungen Algorithmische Codierungstheorie und Kryptographie. Wird dieses Modul jedoch als Vertiefungsmodul gewählt, so sollten zwei Veranstaltungen aus dem Bereich Codierungstheorie oder dem Bereich Kryptographie gehört werden.

Die beiden grundlegenden Veranstaltungen Algorithmische Codierungstheorie I und Kryptographie gliedern sich im Einzelnen sich folgendermaßen.

- Algorithmische Codierungstheorie I

1. Grundlegende Konzepte - Präfix-Codes, erwartete Codewortlänge, kompakte Codes, Entropie
2. Huffman-Codes
3. Shannon-Fano-Elias Codes und arithmetische Codierung

4. Lempel-Ziv Codierung
5. Vorhersagendes Codieren– PPM, Burrows-Wheeler, JPEG-LS, CALIC
6. JPEG als Beispiel für verlustbehaftete Kompression

- Kryptographie

1. Symmetrische und asymmetrische Verschlüsselung
2. Digitale Unterschriften
3. Authentisierungsverfahren
4. Kryptographisches Hashing
5. Semantische Sicherheit

Inhaltliche Verwendbarkeit

Viele Methoden und Techniken aus der Codierungstheorie I finden Anwendung in der Computergrafik, so tauchen auf in Verfahren wie JPEG und MPEG. Fehlerkorrigierende Codes finden Anwendung in vielen Bereichen der Datenübertragung. Es gibt aber auch Anwendungen von fehlerkorrigierenden Codes in der Komplexitätstheorie. Die Kryptographie findet viele Anwendung etwa im E-Banking und im E-Commerce.

Voraussetzungen und Vorkenntnisse

Die wesentlichen Konzepte aus der Algorithmentheorie werden vorausgesetzt.

Neben mathematischem Grundwissen, wie es im Grundstudium vermittelt wird, ist insbesondere für Algorithmische Codierungstheorie Kenntnis von Gruppen und endlichen Körpern unerlässlich. Im Bereich der Kryptographie werden die Inhalte der Vorlesung Einführung in Kryptographie vorausgesetzt. Weiter ist Interesse an kreativem Problemlösen mit mathematisch exakten Methoden notwendig.

Lernziele der Veranstaltung

Die wesentlichen Konzepte und Verfahren der (verlustfreien) Datenkompression, der Theorie der fehlerkorrigierenden Codes und/oder der Kryptographie sollen am Ende der Veranstaltungen bekannt sein. Die Verfahren werden in Übungen und Präsenzübungen eingeübt. Studenten sollten am Ende auch in der Lage sein, bekannt Verfahren unterschiedlichen Anwendungen gemäß anzupassen.

Modulzugehörigkeit

Wahlmodul im Gebiet Modelle und Algorithmen.

Modus

- Leistungspunkte : 4+4
- SWS 2+1,2+1
- Häufigkeit des Angebotes: Das Modul wird in unterschiedlichen Ausprägungen jedes Jahr angeboten.

Prüfungsmodalitäten

Je nach Teilnehmerzahl mündliche Prüfung oder Klausur

Modulverantwortlicher

Blömer

III.2.6 Optimierung

Rolle des Moduls im Studiengang

Optimierung ist eine zentrale Aufgabenstellung sowohl innerhalb der Informatik (z. B. Hardwareentwurf, Datenbanken, Betriebssysteme, Lastverteilung) als auch in insbesondere betriebswirtschaftlichen Anwendungen (z. B. Verschnittprobleme, Planungsaufgaben, Logistik, „Supply Chain Management“). In Abgrenzung zu numerischen Verfahren auf der Grundlage einer Modellierung mittels Differenzialgleichungen beschäftigen sich die meisten Veranstaltungen dieses Moduls mit kombinatorischen Verfahren auf der Grundlage diskreter Modelle.

Themen der zugehörigen Veranstaltungen sind:

- die gemeinsamen (auch historischen) Wurzeln von Simplexalgorithmus und Graphalgorithmen,
- die neuesten Entwicklungen, die aus den Ideen von Khachian und Karmarkar zu Polynomialzeit-Algorithmen für die Lineare Optimierung erwachsen sind,
- exakte und heuristische Verfahren zur Lösung von NP-schweren Optimierungsproblemen unter besonderer Berücksichtigung so genannter Metaheuristiken (z. B. Branch-and-Bound, Simulated Annealing, Tabu Search),
- Netzwerkfluss-Probleme wie z. B. maximale oder kostenminimale Flüsse und Mehrgüterflüsse und deren vielfältige Anwendungen.

Die Veranstaltungen dieses Moduls liefern einen passgenauen Einstieg in immer wieder aktuelle, praxisbezogene Projektarbeiten, die im Rahmen von Projektgruppen und Masterarbeiten behandelt werden. Die Themen spannen einen attraktiven Bogen von aktuellster Forschung zu anspruchsvollen Problemen der industriellen Praxis.

Katalog der Wahlpflichtveranstaltungen

Für das Modul Optimierung werden regelmäßig pro Studienjahr mindestens drei der folgenden Veranstaltungen angeboten:

- Optimierung und Graphalgorithmen
- Innere-Punkte-Verfahren
- Kombinatorische Optimierung
- Netzwerkfluss-Algorithmen
- Algorithmische Spieltheorie

Darüber hinaus werden in unregelmäßigen Abständen forschungsnahe Spezialveranstaltungen, u. a. auch als Seminar, im kommentierten Vorlesungsverzeichnis diesem Modul zugeordnet.

Inhaltliche Verwendbarkeit

In diesem Modul lernen die Studierenden grundlegende Bausteine für das Verständnis bzw. die Entwicklung großer Planungs- und Entscheidungsunterstützungssysteme kennen und anwenden.

Voraussetzungen und Vorkenntnisse

Die beste Voraussetzung liefern die Veranstaltungen Grundlegende Algorithmen und Optimierung des Pflichtmoduls „Modelle und Algorithmen“ im Bachelorstudiengang

Informatik. Ein gutes Verständnis von „Mathematik I für Informatiker“ (Lineare Algebra) und von „Datenstrukturen und Algorithmen“ ist aber ebenfalls eine gute Voraussetzung.

Lernziele des Moduls

Die Studierenden sollen die grundlegenden Methoden der Optimierung, besonders der linearen, als Bausteine für die Lösung hochkomplexer Probleme begreifen und sie auf neue, verwandte Problemstellungen anpassen und anwenden. Sie sollen erkennen, dass Lösungen praktisch relevanter Probleme wie Gewinnmaximierung oder Ressourcenminimierung soziale Auswirkungen haben. Die Lösung von Optimierungsproblemen soll als Unterstützung unternehmerischer Entscheidungen nicht als Entscheidung selbst begriffen werden.

Schlüsselqualifikationen

Das Lösen hochkomplexer Aufgaben ist prädestiniert für die Einübung von Arbeiten im Team. Die Übungen sollen dem Rechnung tragen.

Modulzugehörigkeit

Dieses Modul ist Teil des Katalogs von Wahlpflichtmodulen im Gebiet „Modelle und Algorithmen“ (MUA).

Modus

Für das Modul wird ein Arbeitsaufwand entsprechend 8 Leistungspunkten erwartet. Jede der aufgeführten Wahlpflichtveranstaltungen wird in der Form 2V, 1Ü mit 4 Leistungspunkten angeboten. Das Modul wird in der Regel jedes Jahr angeboten. Während der Dauer des Moduls von 2 Semestern werden mindestens drei Veranstaltungen abgehalten.

Prüfungsmodalitäten

Die Prüfungsmodalitäten werden für die einzelnen Veranstaltungen im kommentierten Vorlesungsverzeichnis veröffentlicht. Die Prüfungen sind in der Regel mündliche Prüfungen. Die Modulnote ergibt sich als arithmetischen Mittel der beiden Veranstaltungsnoten.

Modulverantwortlicher

Hauenschild

III.3 Gebiet Eingebettete Systeme und Systemsoftware

III.3.1 Verteilte Rechnersysteme

Rolle des Moduls im Studiengang

Im Umfeld einer globalen, arbeitsteiligen Wirtschaft gehören vernetzte Computersysteme zu unverzichtbaren Infrastrukturen für moderne Informationssysteme und andere Systeme der Informatik; ihr sicheres und schnelles Funktionieren ist an vielen Stellen kritisch für den unternehmerischen Erfolg.

Solche vernetzten Computersystemen beruhenden auf den grundlegenden Konzepten der Rechnernetze, der Betriebs- und der verteilte Systeme. Die Betriebssysteme stellen eine Verbindung zwischen der Rechnerhardware und der Software her und stellen eine Schnittstelle zu den Hardwareressourcen zur Verfügung. Rechnernetze ermöglichen den Transport von Daten zwischen getrennten Rechnern. Dabei werden unterschiedlichste Kommunikationskanäle genutzt (drahtgebunden, Glasfaser, drahtlos), Geräte unterschiedlicher Leistungsklassen verbunden und unterschiedliche Dienstgarantien abgegeben (korrekte, verlässliche, effiziente Kommunikation). Bei verteilten Systemen wird – basierend auf der existierenden Rechnernetz – eine Interaktion über Rechnergrenzen hinweg ermöglicht, so dass z.B. unterschiedliche, räumlich getrennte Abteilungen eines Unternehmens verbunden oder allgemeine Webdienste realisiert werden können. Systeme zur verteilten Verarbeitung werden auch dann eingesetzt, wenn eine Beschleunigung oder eine Ausfallsicherheit erzielt werden soll. In allen Fällen ist es jedoch notwendig, dass die Umsetzung für den Benutzer möglichst transparent, zuverlässig und sicher erfolgt. Insbesondere die Sicherheitsaspekte spielen eine große Rolle, da die Verarbeitung über in der Regel unsichere Netzwerkstrukturen erfolgt.

In diesem Modul werden zunächst allgemeine Prinzipien, die für die Verwirklichung solcher Systeme notwendig sind, erarbeitet. Die allgemeinen Prinzipien werden auf konkrete Systemsoftware, Rechnerressourcen sowie Programmiermodelle übertragen und durch Fallstudien verdeutlicht.

Dieses Modul ist für Studierende konzipiert, die sich im Software-lastigen Teil des Bereichs „Eingebettete Systeme und Systemsoftware“ (ESS) vertiefen, aber keine weiteren ESS-Module absolvieren wollen. Dieses Modul baut auf den in den Modulen „Konzepte und Methoden der Systemsoftware“ und „Eingebettete Systeme und Systemsoftware“ vorgestellten Grundlagen auf und setzt mindestens eine der beiden Veranstaltungen „Einführung in Verteilte Systeme“ oder „Rechnernetze“ aus dem 2. SA des BSc-Studiengangs voraus.

Inhaltliche Gliederung der Veranstaltung

Das Modul beinhaltet die Gebiete Betriebssysteme, Rechnernetze, Systemaspekte und Algorithmen der verteilten Systeme, Sicherheit in Rechnersystemen, mobile Kommunikation in unterschiedlichen Systemkontexten, Hochleistungsnetzwerke und verschiedene Aspekte des Cluster Computings und Performance-optimierter Programmierung; zusätzlich wird die Leistungsbewertung und Optimierung solcher Systeme behandelt. Die Veranstaltung über Betriebssysteme stellt den Aufbau und die Grundkomponenten moderner Betriebssysteme dar und behandelt Algorithmen und Strategien zur effizienten Ressourcenverwaltung. Die Umsetzung wesentlicher Mechanismen wird am Beispiel aktueller Betriebssysteme vorgestellt. Die Methoden der Rechnerkommunikation, fortgeschrittenen Konzepte der Rechnernetz und Netzwerktechnologien sowie moderne Formen der mobilen

Kommunikation, der integrierten Sprach- und Datennetze und exemplarische Mehrwertdienste wie Videokonferenzen werden in einer Reihe von Veranstaltung über Rechnernetze vermittelt; zum Beispiel wird sowohl auf klassische mobile Kommunikationssysteme für Sprachübermittlung wie GSM eingegangen wie auch moderne Systemkonzepte wie drahtlose Sensornetze behandelt. Die Veranstaltung über verteilte Systeme vermittelt Kenntnisse über grundlegende verteilte Algorithmen sowie unterstützende Aspekte aus den Bereichen Rechnerkommunikation, Betriebssysteme, Sicherheit und verteiltes Datenmanagement. Im Gebiet Sicherheit in Rechnersystemen werden Gefährdungen betrachtet, denen informationsverarbeitende Systeme ausgesetzt sind sowie entsprechende Mechanismen zur Abwehr solcher Gefahren. Dazu gehören Themen wie Vertraulichkeit, Integrität, Verfügbarkeit, Autorisierung, Zugangs- und Nutzungskontrolle, Sicherheitsmodelle und -architekturen. Die Veranstaltungen über Cluster Computing, Architektur paralleler Rechnersysteme und Performance-optimierte Programmierung führen in die Welt des Hochleistungsrechnens. Dabei werden sowohl der prinzipielle Aufbau als auch Fall Beispiele von Hochleistungsrechnern vorgestellt. Ferner wird auf die entsprechende Systemsoftware, auf die Netzwerkstrukturen und insbesondere auf die Programmiermodelle für Parallelrechner eingegangen. In methodisch orientierten Veranstaltungen werden unterschiedliche Verfahren der Leistungsbewertung von Systemen, z.B. durch Analyse oder Simulation, Fragen der Experimentplanung und der Leistungsoptimierung behandelt.

Inhaltliche Verwendbarkeit

Typische Anwendungsfelder sind in der Anwendungsentwicklung, Systemadministration sowie Entwurf und Umsetzung von Spezialsystemen zu finden. Die vorgestellten Mechanismen zur Ressourcenverwaltung, Sicherheit, rechnerübergreifenden Kommunikation werden sowohl bei klassischen Informationssystemen als auch – in angepasster Form – bei speziellen Hardwareressourcen angewendet. Ferner hilft das Wissen über die detaillierte Funktionsweise von Rechnernetzen dem Informatiker, den komplexen Anforderungen moderner Informationssysteme gerecht zu werden und neue Anwendungsfelder zu erschließen. Oft spielen zeitabhängige Vorgänge in kommerziellen und technischen Systemen eine wichtige Rolle. Schließlich werden hilfreiche Grundkenntnisse für die Netzwerkadministration erlangt.

Die Grundbausteine zur Erstellung verteilter Systeme werden bei Internetanwendungen, Webdiensten, Unternehmenssoftware, usw. benötigt. Schließlich soll das vermittelte Wissen, die Bewertung, Auswahl und Anpassung verschiedener Lösungswege und -komponenten an eine spezifische Aufgabenstellung ermöglichen. Schließlich werden die Kenntnisse über Hochleistungsrechnen auch in vielen verwandten Wissenschaften benötigt, in denen komplexe, rechenintensive Aufgaben zu lösen sind.

Voraussetzungen und Vorkenntnisse

Als Voraussetzung dienen die Veranstaltung „Einführung in verteilte Systeme“ und/oder „Rechnernetze“ aus dem 2. SA des BSc-Studiengangs Informatik. Als weitere Vorkenntnisse werden die Inhalte des Moduls KMS des 1. SA vorausgesetzt. Für Veranstaltungen wie performance-orientierte Programmierung wird die Bereitschaft erwartet, sich in systemnahe Programmiersprachen einzuarbeiten. Ferner sind grundlegende Kenntnisse der Programmiersprachen aus SWE zwingend erforderlich.

Prüfbare Standards / Lernziele der Veranstaltung

Die Studierenden sollen das Verständnis der spezifischen Eigenschaften von Systemsoftware und Rechnernetze erlangen sowie die elementaren Bausteine zum Aufbau von Betriebs- und verteilten Systemen kennen lernen. Die Studierenden sollen mögliche Gefahren für den

Rechnerbetrieb durch einen nicht-authorisierten Zugriff auf die Ressourcen erkennen und entsprechende Maßnahmen ergreifen können. Sie sollen in der Lage sein, Möglichkeiten, Grenzen und Risiken offener verteilter Systeme sowie von Hochleistungsrechnern einschätzen und evaluieren lernen. Schließlich sollen die Kernmethoden für effiziente Bearbeitung und Ressourcenverwaltung verstanden und an konkreten Beispielen angewendet werden.

Vermittlung von Faktenwissen – Inhaltskompetenz

- Zusammenhang zwischen Hardware und Systemsoftware
- Aufbau, Verwaltung und Synchronisation von Prozessen
- Techniken zur Speicherverwaltung und für Scheduling
- Techniken zur Sicherung von kritischen Bereichen
- Techniken für den Entwurf von parallelen und nebenläufigen Programmen
- Techniken der effizienten, problem- und anforderungsgerechten Übertragung von Daten in drahtlosen und mobilen Kommunikationssystemen
- Erweiterte und spezialisierte Verfahren und Techniken des Internets

Vermittlung von methodischem Wissen – Methodenkompetenz

- Methoden zur effizienten Verwaltung und Zuordnung von Betriebsmitteln
- Methoden zur Erkennung und Vermeidung von Verklemmungen
- Methoden zur Kooperation zwischen Prozessen in verteilten Systemen
- Methoden für Prozessinteraktion
- Methoden der Leistungsbewertung und –optimierung von Kommunikationssystemen und ähnlichen technischen Systemen

Vermittlung von Transferkompetenz

Übertragung der globalen Strategien auf vorgegebene Einzelsituationen, zum Beispiel im Rahmen von Übungsaufgaben

Vermittlung von normativ-bewertender Kompetenz

- Strategien zur Behandlung unterschiedlicher Strategien ausarbeiten
- Auswahl einer für eine gegebene Aufgabe geeigneten Strategie anhand des Optimierungsziels und zu berücksichtigender Rahmenbedingungen.

Schlüsselqualifikationen

Durch den Übungsbetrieb in kleinen Gruppen wird die Kooperations- und Teamfähigkeit gefördert.

Modulzugehörigkeit

Wahlmodul im Gebiet „Eingebettete Systeme und Systemsoftware“

Modus

- Leistungspunkte pro Modul (Workload) : 8
- Leistungspunkte der Veranstaltung: jeweils 4
- SWS: 2V+1Ü, 2V+1Ü
- Häufigkeit des Angebotes: 2-4 Veranstaltungen pro Jahr im WS und SS
- Dauer: 2 Semester

Methodische Umsetzung

Neben klassischen Vorlesungen und Tafelübungen werden Übungen in kleinen Gruppen eingesetzt. Diese fördern die praktische Anwendung der vorgestellten Methoden auf ausgewählte Beispiele. Insbesondere müssen Parameter und Strategien an eine vorgegebene, konkrete Situation angepasst werden. Die Aufgaben werden in Gruppen von 3 Studierenden bearbeitet, wodurch die Teamfähigkeit gefördert wird. Dieser Ansatz wird durch das Angebot von Projektgruppen verstärkt und erweitert, bei denen reale Problem aus dem Forschungsbetrieb durch Studierende in Gruppenarbeit über einen längeren Zeitraum zu lösen sind. Eine eigenständige Vertiefung in die behandelten Themen wird Studierenden durch Seminare ermöglicht.

Organisationsformen / Medieneinsatz / Literaturangaben

- Folien-orientierte Vorlesung, Tafelanschrieb bei Beispielen, zusätzlichen Erläuterungen und zu potenzierenden Sachverhalten
- Wöchentliche Übungen in kleinen Gruppen. Dabei werden Präsenzaufgaben sowie die Musterlösungen zu den Hausaufgaben vorgerechnet
- Erwartete Aktivitäten der Studierenden: Mitarbeit bei Präsenzübungen, Hausaufgaben, selbständiges Studium von sekundärer Literatur
- Folienskript ist vorhanden auf der Homepage der Vorlesung

Prüfungsmodalitäten

In der Regel Einzelprüfungen pro Veranstaltung und anschließende Berechnung des Mittelwertes der beiden Einzelleistungen. Die Prüfungen zu den einzelnen Veranstaltungen im Modul werden je nach Teilnehmerzahl als Klausur oder als Fachgespräch durchgeführt. Abweichungen von diesem Schema sowie zusätzliche Forderungen werden bei der Ankündigung der entsprechenden Vorlesungen angegeben.

Modulverantwortliche(r)

Karl

III.3.2 Systemsoftware

Rolle des Moduls im Studiengang

Betriebssysteme bilden die grundlegende Softwareschicht, welche eine Verbindung zwischen der Rechnerhardware und der Software herstellt. Zusammen mit anderen Komponenten der Systemsoftware wird die Erstellung von Anwendungen ermöglicht und eine Schnittstelle zu den Hardwareressourcen zur Verfügung gestellt. Bei verteilten Systemen wird hingegen eine Interaktion über Rechengrenzen hinweg ermöglicht, so dass unterschiedliche, räumlich getrennte Abteilungen eines Unternehmens verbunden bzw. allgemeine Webdienste realisiert werden können. Systeme zur verteilten Verarbeitung werden auch dann eingesetzt, wenn eine Beschleunigung oder eine Ausfallsicherheit erzielt werden soll. In allen Fällen ist es jedoch notwendig, dass die Umsetzung für den Benutzer möglichst transparent, zuverlässig und sicher erfolgt. Insbesondere die Sicherheitsaspekte spielen eine große Rolle, da die Verarbeitung über unsichere Netzwerkstrukturen erfolgt. Die aktuelle Entwicklung führt zu einer weitgehenden Verschmelzung von Betriebssystemen und verteilten Systemen, so dass viele Zusammenhänge zu erkennen sind.

Dieses Modul ist für Studierende konzipiert, die sich im SW-lastigen Teil des Bereichs „Eingebettete Systeme und Systemsoftware“ (ESS) vertiefen wollen. Die spezifische Ausrichtung auf Betriebs- und Verteilte Systeme ermöglicht die Kombination mit allen anderen ESS-Aspekten, wie zum Beispiel Rechnernetzen oder eingebetteten und Echtzeitsystemen im Rahmen des Vertiefungsgebiets. Dieses Modul baut auf den in den Modulen „Konzepte und Methoden der Systemsoftware“ und „Technische Informatik“ vorgestellten Grundlagen auf und setzt die Veranstaltung „Verteilte Systeme 1“ aus dem 2. SA des BSc-Studiengangs. Die allgemeinen Prinzipien werden nun auf konkrete Systemsoftware, Rechnerressourcen sowie Programmiermodelle übertragen und durch Fallstudien verdeutlicht.

Inhaltliche Gliederung der Veranstaltung

Das Modul beinhaltet die Gebiete Betriebssysteme, Systemaspekte und Algorithmen der Verteilten Systeme, Sicherheit in Rechnersystemen, Cluster Computing, Architektur paralleler Rechnersysteme und Performance-optimierte Programmierung. Die Veranstaltung über Betriebssysteme stellt den Aufbau und die Grundkomponenten moderner Betriebssysteme dar und behandelt Algorithmen und Strategien zur effizienten Ressourcenverwaltung. Die Umsetzung wesentlicher Mechanismen wird am Beispiel aktueller Betriebssysteme vorgestellt. Die Veranstaltung über Verteilte Systeme vermittelt Kenntnisse über grundlegende verteilte Algorithmen sowie unterstützende Aspekte aus den Bereichen Rechnerkommunikation, Betriebssysteme, Sicherheit und verteiltes Datenmanagement. Im Gebiet Sicherheit in Rechnersystemen werden Gefährdungen betrachtet, denen informationsverarbeitende Systeme ausgesetzt sind sowie entsprechende Mechanismen zur Abwehr solcher Gefahren. Dazu gehören Themen wie Vertraulichkeit, Integrität, Verfügbarkeit, Autorisierung, Zugangs- und Nutzungskontrolle, Sicherheitsmodelle und –architekturen, Verschlüsselung, Schlüsselverwaltung, sichere Protokolle, Firewalls und Sicherheit in Netzen. Die Veranstaltungen über Cluster Computing, Architektur paralleler Rechnersysteme und Performance-optimierte Programmierung führen in die Welt des Hochleistungsrechnens. Dabei werden sowohl der prinzipielle Aufbau als auch Fall Beispiele von Hochleistungsrechnern vorgestellt. Ferner wird auf die entsprechende Systemsoftware, auf die Netzwerkstrukturen und insbesondere auf die Programmiermodelle für Parallelrechner eingegangen.

Inhaltliche Verwendbarkeit

Typische Anwendungsfelder sind in der Anwendungsentwicklung, Systemadministration sowie Entwurf- und Umsetzung von Spezialexsystemen zu finden. Die vorgestellten Mechanismen zur Ressourcenverwaltung, Sicherheit, rechnerübergreifende Kommunikation werden sowohl bei klassischen Informationssystemen als auch – in angepasster Form – bei speziellen Hardwareressourcen angewendet. Die Kenntnisse über Hochleistungsrechnen werden auch in vielen verwandten Wissenschaften benötigt, in denen komplexe, rechenintensive Aufgaben zu lösen sind. Die Grundbausteine zur Erstellung verteilter Systeme werden bei Internetanwendungen, Webdiensten, Unternehmenssoftware, usw. benötigt. Schließlich soll das vermittelte Wissen, die Bewertung, Auswahl und Anpassung verschiedener Lösungswege und -komponenten an eine spezifische Aufgabenstellung ermöglichen.

Voraussetzungen und Vorkenntnisse

Als Voraussetzung dient die Veranstaltung Einführung in Verteilte Systeme aus dem 2. SA des BSc-Studiengangs. Als weitere Vorkenntnisse werden die Inhalte der Module Technische Informatik und KMS vorausgesetzt. Für Veranstaltungen wie performance-orientierte Programmierung wird die Bereitschaft erwartet, sich in systemnahe Programmiersprachen einzuarbeiten. Ferner sind grundlegende Kenntnisse der Programmiersprachen aus SWE zwingend erforderlich.

Prüfbare Standards / Lernziele der Veranstaltung

Die Studierenden sollen das Verständnis der spezifischen Eigenschaften von Systemsoftware erlangen sowie die elementaren Bausteine zum Aufbau von Betriebs- und verteilten Systemen kennen lernen. Die Studierenden sollen mögliche Gefahren für den Rechnerbetrieb durch einen nicht-authorisierten Zugriff auf die Ressourcen erkennen und entsprechende Maßnahmen ergreifen können. Sie sollen in der Lage sein, Möglichkeiten, Grenzen und Risiken offener verteilter Systeme sowie von Hochleistungsrechnern einschätzen und evaluieren lernen. Schließlich sollen die Kernmethoden für effiziente Bearbeitung und Ressourcenverwaltung verstanden und an konkreten Beispielen angewendet werden.

Vermittlung von Faktenwissen – Inhaltskompetenz

- Zusammenhang zwischen Hardware und Systemsoftware
- Aufbau, Verwaltung und Synchronisation von Prozessen
- Techniken zur Speicherverwaltung und für Scheduling
- Techniken zur Sicherung von kritischen Bereichen
- Techniken für den Entwurf von parallelen und nebenläufigen Programmen

Vermittlung von methodischem Wissen – Methodenkompetenz

- Methoden zur effizienten Verwaltung und Zuordnung von Betriebsmitteln
- Methoden zur Erkennung und Vermeidung von Verklemmungen
- Methoden zur Kooperation zwischen Prozessen in verteilten Systemen
- Methoden für Prozessinteraktion

Vermittlung von Transferkompetenz

Übertragung der globalen Strategien auf vorgegebene Einzelsituationen, zum Beispiel im Rahmen von Übungsaufgaben

Vermittlung von normativ-bewertender Kompetenz

- Strategien zur Behandlung unterschiedlicher Strategien ausarbeiten
- Den praktischen Wert der Konzepte und Methoden der Systemsoftware erkennen

Schlüsselqualifikationen

Durch den Übungsbetrieb in kleinen Gruppen wird die Kooperations- und Teamfähigkeit gefördert.

Modulzugehörigkeit

Wahlmodul im Gebiet „Eingebettete Systeme und Systemsoftware“

Modus

- Leistungspunkte pro Modul (Workload) : 8
- Leistungspunkte der Veranstaltung: jeweils 4
- SWS (2V+1Ü, 2V+1Ü)
- Häufigkeit des Angebotes: 2-4 Veranstaltungen pro Jahr im WS und SS
- Dauer (2 Semester)

Methodische Umsetzung

Übungen in kleinen Gruppen fördern die praktische Anwendung der vorgestellten Methoden auf ausgewählte Beispiele. Insbesondere müssen Parameter und Strategien an die konkrete Situation angepasst werden. Die Aufgaben werden in Gruppen von 3 Studierenden bearbeitet, wodurch die Teamfähigkeit gefördert wird. Der Aufbau der Aufgabenzettel bildet den Aufbau der Systemsoftware nach, beginnend von der Hardware über Prozesse bis zu Betriebsmittelverwaltung und Scheduling.

Organisationsformen / Medieneinsatz / Literaturangaben

- Folien-orientierte Vorlesung, Tafelanschrieb bei Beispielen, zusätzlichen Erläuterungen und zu potenzierenden Sachverhalten
- Wöchentliche Übungen in kleinen Gruppen. Dabei werden Präsenzaufgaben sowie die Musterlösungen zu den Hausaufgaben vorgerechnet
- Erwartete Aktivitäten der Studierenden: Mitarbeit bei Präsenzübungen, Hausaufgaben, selbständiges Studium von sekundärer Literatur
- Folienskript ist vorhanden auf der Homepage der Vorlesung

Prüfungsmodalitäten

In der Regel Einzelprüfungen pro Veranstaltung und anschließende Berechnung des Mittelwertes der beiden Einzelleistungen. Die Prüfungen zu den einzelnen Veranstaltungen im Modul werden je nach Teilnehmerzahl als Klausur oder als Fachgespräch durchgeführt. Abweichungen von diesem Schema sowie zusätzliche Forderungen werden bei der Ankündigung der entsprechenden Vorlesungen angegeben.

Modulverantwortliche(r)

Rammig

III.3.3 Rechnernetze

Rolle des Moduls im Studiengang

Die Übertragung von Daten zwischen unterschiedlichen Systemen über unterschiedliche Übertragungsmedien hinweg stellt einen grundlegenden Baustein für nahezu alle modernen Informationssysteme dar. Diese Übertragung ermöglicht erst die verteilten Systeme; sie schafft die Möglichkeiten der mobilen Kommunikation durch die Benutzung drahtloser Übertragungsmedien; sie bewirkt in unterschiedlichen Systemstrukturen unterschiedlichste Anforderungen und erfährt unterschiedlichste Ausprägungen – von hoch verlässlichen aber kleinen Automatisierungsnetzen über das Internet bis zu selbst organisierten, drahtlos kommunizierenden ad hoc Netzen. Dieser Themenkreis wird im Modul „Rechnernetze“ behandelt.

Das Modul „Rechnernetze“ ist für Studierende konzipiert, die sich im Bereich „Eingebettete Systeme und Systemsoftware“ (ESS) vertiefen und das Modul „Rechnernetze“ mit einem der drei andern ESS-Schwerpunkte „Betriebs- und verteilte Systeme“, „Eingebettete und Echtzeitsysteme“, oder „HW/SW-Codesign“ kombinieren wollen. Dieses Modul baut auf den im Modul „Eingebettete Systeme und Systemsoftware“ aus dem 2. SA des Bachelorstudiengangs auf und setzt die Veranstaltung über die „Rechnernetze“ voraus.

Inhaltliche Gliederung des Moduls

Das Modul besteht aus einem wechselnden Kanon von Lehrveranstaltungen über vertiefende Aspekte von Rechnernetzen/Internet, Einführung und Vertiefung in Mobilkommunikation, Hochleistungsnetzwerke, oder Sicherheit von Rechnernetzen (der aktuelle Katalog ist den Webseiten des Modulverantwortlichen zu entnehmen). Dabei werden die Aufgaben und Architektur von Kommunikationssystemen erörtert und Fragestellung hinsichtlich Aufbau und Klassifikation von Rechnernetzen, Adressierung, Routing, ... anhand von konkreten Konzepten und Protokollen von drahtgebundenen und drahtlosen Kommunikationssystemen verdeutlicht. Zusätzlich werden Fallbeispiele der Hochgeschwindigkeitsnetze, mobiler Netze, integrierter Sprach- und Datennetze, Funkübertragung und Vermittlung, Videokonferenzen, Multimediasysteme, Mehrpunktkommunikation, Netzwerksicherheit und Netzwerkmanagement behandelt.

Zu diesen technologisch orientierten Veranstaltungen treten Veranstaltungen zur Leistungsbewertung und –optimierung von Kommunikationssystemen (die aber auch auf technische Systeme weitgehend anwendbar sind). Hier werden Verfahren der Analyse und Simulation, der Experimentplanung und Parameteroptimierung behandelt.

Inhaltliche Verwendbarkeit

Das Wissen über die detaillierte Funktionsweise von Rechnernetzen hilft dem Informatiker, den komplexen Anforderungen moderner Informationssysteme gerecht zu werden und neue Anwendungsfelder zu erschließen. Neben einer theoretischen Durchdringung dieser Kommunikationssysteme wird praktische Kompetenz in der Nutzung, Planung, Konfiguration, Programmierung, und Administration dieser Netze vermittelt, die in vielfältigen Berufsbildern eines Informatikers Verwendung finden. Ferner ist die detailgenaue Modellierung relevanter Aspekte und Vorgänge in einem Rechnernetz eine wesentliche Grundlage für eine simulative Leistungsbewertung – vor allem von noch nicht existierenden Systemen oder Protokollen. Die formale Spezifikation von Kommunikationssystemen dient der (ggf. halb-) automatischen Implementierung von Protokollen mit Hilfe von entsprechenden Programmierertools sowie dem Testen der Systeme. Die Implementierung mündet in einer Leistungsbewertung in Form von Messungen in einer Laborumgebung.

Voraussetzungen und Vorkenntnisse

Als Voraussetzung dient die Veranstaltung „Rechnernetze“ aus dem 2. SA des Bachelorstudiengangs. Als weitere Vorkenntnisse werden die Inhalte des Moduls „Konzepte und Methoden der Systemsoftware“ des 1. SA vorausgesetzt. Ferner sind grundlegende Kenntnisse der Programmiersprachen aus SWE zwingend erforderlich.

Prüfbare Standards / Lernziele der Veranstaltung

Ausgehend von bekannten Grundlagen der Informatik sollen die Studierenden grundlegende Konzepte und unterschiedliche Funktionsweisen von Rechnernetzen und deren Nutzung kennen lernen und verstehen. Studierende, die sich in diesem Bereich spezialisieren, sollen sich mit den Kernkonzepten und Protokollen von Kommunikationssystemen vertraut machen und auch Gründe für Entwurfsentscheidungen verstehen. Von Spezialisten wird erwartet, dass sie die Methoden und Verfahren zur Modellierung/formalen Spezifikation von Kommunikationssystemen sowie zur Leistungsbewertung durch Simulation/Messung kennen und an eine spezifische Fragestellung anpassen können.

Vermittlung von Faktenwissen – Inhaltskompetenz

- Techniken der effizienten, problem- und anforderungsgerechten Übertragung von Daten in drahtlosen und mobilen Kommunikationssystemen
- Erweiterte und spezialisierte Verfahren und Techniken des Internets

Vermittlung von methodischem Wissen – Methodenkompetenz

- Methoden der Leistungsbewertung und –optimierung von Kommunikationssystemen und ähnlichen technischen Systemen
- Spezifikation von Kommunikationssystemen und –protokollen
- Ansätze zur systematischen Protokollimplementierung

Vermittlung von Transferkompetenz

Übertragung der globalen Strategien auf vorgegebene Einzelsituationen, zum Beispiel im Rahmen von Übungsaufgaben

Vermittlung von normativ-bewertender Kompetenz

- Strategien zur Behandlung unterschiedlicher Strategien ausarbeiten
- Auswahl einer für eine gegebene Aufgabe geeigneten Strategie anhand des Optimierungsziels und zu berücksichtigender Rahmenbedingungen.

Schlüsselqualifikationen

Durch den Übungsbetrieb in kleinen Gruppen wird die Kooperations- und Teamfähigkeit gefördert. Praktische Erfahrung kann durch Implementierungen in Praktika und Projektgruppen erworben werden; vertiefte Kenntnisse werden in Seminaren zu aktuellen Themen erworben.

Modulzugehörigkeit

Wahlmodul im Gebiet „Eingebettete Systeme und Systemsoftware“

Modus

- Leistungspunkte pro Modul (Workload) : 8
- Leistungspunkte der Veranstaltung: jeweils 4
- SWS: 2V+1Ü, 2V+1Ü, Projektgruppen, Seminare

- Häufigkeit des Angebotes: 2-4 Veranstaltungen pro Jahr im WS und SS
- Dauer: 2 Semester

Methodische Umsetzung

Neben klassischen Vorlesungen und Tafelübungen werden Übungen in kleinen Gruppen eingesetzt. Diese fördern die praktische Anwendung der vorgestellten Methoden auf ausgewählte Beispiele. Insbesondere müssen Parameter und Strategien an eine vorgegebene, konkrete Situation angepasst werden. Die Aufgaben werden in Gruppen von 3 Studierenden bearbeitet, wodurch die Teamfähigkeit gefördert wird. Dieser Ansatz wird durch das Angebot von Projektgruppen verstärkt und erweitert, bei denen reale Problem aus dem Forschungsbetrieb durch Studierende in Gruppenarbeit über einen längeren Zeitraum zu lösen sind. Eine eigenständige Vertiefung in die behandelten Themen wird Studierenden durch Seminare ermöglicht.

Organisationsformen / Medieneinsatz / Literaturangaben

- Folien-orientierte Vorlesung, Tafelanschrieb bei Beispielen, zusätzlichen Erläuterungen und zu potenzierenden Sachverhalten
- Wöchentliche Übungen in kleinen Gruppen. Dabei werden Präsenzaufgaben sowie die Musterlösungen zu den Hausaufgaben vorgerechnet
- Erwartete Aktivitäten der Studierenden: Mitarbeit bei Präsenzübungen, Hausaufgaben, selbständiges Studium von sekundärer Literatur
- Folienskript ist vorhanden auf der Homepage der Vorlesung

Prüfungsmodalitäten

In der Regel Einzelprüfungen pro Veranstaltung und anschließende Berechnung des Mittelwertes der beiden Einzelleistungen. Die Prüfungen zu den einzelnen Veranstaltungen im Modul werden je nach Teilnehmerzahl als Klausur oder als Fachgespräch durchgeführt. Abweichungen von diesem Schema sowie zusätzliche Forderungen werden bei der Ankündigung der entsprechenden Vorlesungen angegeben.

Modulverantwortliche(r)

Karl

III.3.4 Eingebettete Systeme

Rolle des Moduls im Studiengang

Eingebettete Systeme spielen die zentrale Rolle im Zuge der sich ständig verstärkenden Informatisierung aller technischen Systeme. Große Bereiche der Maschinen- Automobil-, Luft- u. Raumfahrtstechnik, aber auch der Kommunikationstechnik sind ohne eingebettete Systeme nicht mehr realisierbar. Unter Eingebetteten Systemen werden die informationsverarbeitenden Anteile in derartigen Systemen verstanden. Sie bestehen in der Regel aus dedizierter Hardware und darauf aufsetzender Software. Beides wird mit den grundsätzlichen Methoden der Informatik entworfen, wobei auch das Zusammenspiel zwischen HW und SW eine wichtige Rolle spielt. Eine wesentliche Besonderheit der Eingebetteten Systeme ist aber, dass die physikalischen Gesetze des Gesamtsystems eine bestimmende Rolle spielen und beim Entwurf berücksichtigt werden müssen. Neben Realzeitanforderungen sind hier auch die Ressourcenbeschränktheit (z.B. bezüglich des Stromverbrauchs) zu berücksichtigen. Dies führt dazu, dass der allgemeine Entwurfszyklus von Informatiksystemen in allen Phasen spezifisch angepasst werden muss. So müssen bei der Spezifikation und der Modellierung Realzeit- und Ressourcenbeschränkung beschreibbar sein, was zu spezifischen Formalismen führt. Die abstrakten Modelle müssen im Zusammenspiel mit den umgebenden (z.T. ebenfalls modellierten oder real existierenden) Systemteilen validiert und analysiert werden. Die Partitionierung in Hardware und Software geschieht bei Eingebetteten Systemen auf der Basis der zu erfüllenden Restriktionen. Der Syntheseprozess wird ebenfalls von der Vorgabe dominiert, diese Restriktionen zu respektieren. Da Eingebettete Systeme in der Regel sicherheitsrelevante Anteile beinhalten, teilweise die Sicherheit von Systemen erst gewährleisten, sind hier besonders stringente Verifikationstechniken anzuwenden, die zudem, nicht zuletzt wegen der zwingend erforderlichen Berücksichtigung von Realzeit besonders komplex sind.

Dieses Modul ist für Studierende konzipiert, die sich in den Aspekten des Bereichs „Eingebettete Systeme und Systemsoftware“ (ESS) vertiefen wollen, die sich mit der Interaktion mit physikalischen Systemen auseinandersetzen. Die spezifische Ausrichtung auf Eingebettete Systeme ermöglicht die Kombination mit allen anderen ESS-Aspekten, wie zum Beispiel Rechnernetzen oder Betriebssystemen und Verteilten Systemen im Rahmen des Vertiefungsgebiets. Dieses Modul baut auf den in den Modulen „Konzepte und Methoden der Systemsoftware“ und „Technische Informatik“ vorgestellten Grundlagen auf. Die allgemeinen Prinzipien werden nun auf realzeitfähige Systemsoftware, Abbildung auf Hardwareressourcen sowie anwendungsspezifische Programmiermodelle übertragen und durch Fallstudien verdeutlicht.

Inhaltliche Gliederung der Veranstaltung

Das Modul besteht aus Lehrveranstaltungen über Eingebettete Systeme, Echtzeitbetriebssysteme (Real Time Operating Systems), Distributed and Parallel Embedded Systems, Intelligenz in Eingebetteten Systemen, HW/SW-Codesign, Sensorik, Entwurfsmethoden für Eingebettete Systeme und über Fortgeschrittene Konzepte der Rechnerarchitektur und des HW/SW-Codesigns. Die Veranstaltung über Eingebettete Systeme liefert einen Überblick über die Aufgabenstellung und grundsätzliche Lösungsansätze. Dabei wird fallbasiert vorgegangen. Eine zentrale Bedeutung im Bereich der Eingebetteten Systeme kommt den Realzeitbetriebssystemen dar. Dieser Bereich wird daher in zwei aufeinander aufbauenden Veranstaltungen vertieft. Zunächst werden aus den Konzepten allgemeiner Betriebssysteme die Besonderheiten der Realzeitbetriebssysteme abgeleitet und die grundsätzlichen Konzepte dargestellt. Anschließend werden diese Konzepte verfeinert und

mathematisch präzise unterfüttert. Die spezifischen Aspekte, die bei der zunehmenden Realisierung Eingebetteter Systeme in Form verteilter Systeme zu berücksichtigen sind, werden in einer Veranstaltung über Distributed and Parallel Embedded Systems behandelt. Die Veranstaltung über Intelligenz in eingebetteten Systemen spannt den Bogen zu dem erkennbaren Trend zu mehr autonomen, selbstorganisierenden Systemen. Mit den Veranstaltungen über HW/SW-Codesign und Sensorik werden spezielle Aufgabenstellungen behandelt, die bei Eingebetteten Systemen besonderes Gewicht und eine spezielle Ausprägung haben. Den wichtigen Gesichtspunkt der Entwurfsmethoden für Eingebettete Systeme werden in einer weiteren Veranstaltung abgedeckt. Eingebettete Systeme setzen oft auf spezielle und fortgeschrittene Konzepte der Rechnerarchitektur, z.B. unter Verwendung rekonfigurierbarer Hardware auf. Darauf wird in Veranstaltungen über Fortgeschrittene Konzepte der Rechnerarchitektur eingegangen, während die Veranstaltung über Fortgeschrittenen Konzepte von HW/SW-Codesign die Grundprinzipien dieser Technik ebenfalls verfeinert und erweitert.

Inhaltliche Verwendbarkeit

Typische Anwendungsfelder sind in der Anwendungsentwicklung technischer Systeme sowie beim Entwurf- und der Umsetzung von Spezialsystemen zu finden. Die vorgestellten Methoden zur Spezifikation, Modellierung, Analyse, Synthese und Verifikation werden in allen Anwendungsbeichen eingebetteter Systeme, d.h. im gesamten Bereich technischer Systeme benötigt. Realzeitanwendungen finden aber auch im nichttechnischen Umfeld Verwendung, z.B. in der Wetterprognose oder in der strategischen Planung von Finanzdienstleistungen. Über diesen Anwendungsbezug hinaus eröffnet die Beschäftigung mit eingebetteten Systemen aber auch einen nicht zu vernachlässigenden Erkenntnisgewinn, da man gezwungen ist, sich von der Fiktion des Idealismus im Platonschen Sinn zu lösen und sich mit physikalischen Randbedingungen auseinanderzusetzen.

Voraussetzungen und Vorkenntnisse

Als Voraussetzung dienen die Veranstaltungen über Eingebettete Systeme oder HW/SW-Codesign aus dem 2. SA des BSc-Studiengangs. Als weitere Vorkenntnisse werden die Inhalte der Module Technische Informatik und KMS vorausgesetzt. Ferner sind grundlegende Kenntnisse von Modellierungsprinzipien aus dem Modul Modellierung und der Programmiersprachen aus SWE zwingend erforderlich. Darüber hinaus wird die Bereitschaft erwartet, sich in systemnahe Programmiersprachen einzuarbeiten. In einigen Veranstaltungen, insb. HW/SW-Codesign wird zudem erwartet, sich in Hardwarebeschreibungssprachen einzuarbeiten.

Prüfbare Standards / Lernziele der Veranstaltung

Die Studierenden sollen das Verständnis der spezifischen Eigenschaften von Eingebetteten Systemen erlangen sowie die elementaren Konzepte zum Entwurf derartiger Systeme kennen lernen. Die Studierenden sollen mögliche Gefahren im Falle des fehlerhaften Entwurfs eingebetteter Systeme erkennen und das Instrumentarium zur Vermeidung solcher Fehler beherrschen können. Sie sollen in der Lage sein, die spezifischen Restriktionen, die sich durch die physikalischen Gesetze des umgebenden Systems ergeben, einzuschätzen und lernen diese gezielt in den Entwurfsprozess einzubeziehen. Schließlich sollen die Kernmethoden für die präzise vorhersagbare Nutzung knapper Ressourcen verstanden und an konkreten Beispielen angewendet werden.

Vermittlung von Faktenwissen – Inhaltskompetenz

- Zusammenhang zwischen informatischen und physikalischen Systemkomponenten

- Architekturvarianten für eingebettete Systeme
- Techniken der Realzeitverwaltung
- Techniken zur Validierung und Verifikation
- Techniken für den Entwurf von eingebetteten Systemen

Vermittlung von methodischem Wissen – Methodenkompetenz

- Methoden zur vorhersagbaren Einplanung von Betriebsmitteln
- Methoden zur Interaktion mit physikalischen Systemen
- Methoden zur Verifikation zeitbehafteter Systeme
- Methoden für die gezielte Partitionierung von Aufgaben in HW und SW

Vermittlung von Transferkompetenz

Übertragung der globalen Strategien auf vorgegebene Einzelsituationen, zum Beispiel im Rahmen von Übungsaufgaben

Vermittlung von normativ-bewertender Kompetenz

- Strategien zur Behandlung unterschiedlicher Strategien ausarbeiten
- Den praktischen Wert der Konzepte und Methoden der eingebetteten Systeme erkennen

Schlüsselqualifikationen

Durch den Übungsbetrieb in kleinen Gruppen wird die Kooperations- und Teamfähigkeit gefördert.

Modulzugehörigkeit

Wahlmodul im Gebiet „Eingebettete Systeme und Systemsoftware“

Modus

- Leistungspunkte pro Modul (Workload) : 8
- Leistungspunkte der Veranstaltung: jeweils 4
- SWS (2V+1Ü, 2V+1Ü)
- Häufigkeit des Angebotes: 2-4 Veranstaltungen pro Jahr im WS und SS
- Dauer (2 Semester)

Methodische Umsetzung

Übungen in kleinen Gruppen fördern die praktische Anwendung der vorgestellten Methoden auf ausgewählte Beispiele. Insbesondere müssen Parameter und Strategien an die konkrete Situation angepasst werden. Die Aufgaben werden in Gruppen von 3 Studierenden bearbeitet, wodurch die Teamfähigkeit gefördert wird. Der Aufbau der Aufgabenzettel bildet den Aufbau der Systemsoftware nach, beginnend von der Hardware über Prozesse bis zu Betriebsmittelverwaltung und Scheduling.

Organisationsformen / Medieneinsatz / Literaturangaben

- Folien-orientierte Vorlesung, Tafelanschrieb bei Beispielen, zusätzlichen Erläuterungen und zu potenzierenden Sachverhalten
- Wöchentliche Übungen in kleinen Gruppen. Dabei werden Präsenzaufgaben sowie die Musterlösungen zu den Hausaufgaben vorgerechnet
- Erwartete Aktivitäten der Studierenden: Mitarbeit bei Präsenzübungen, Hausaufgaben, selbständiges Studium von sekundärer Literatur

- Folienskript ist vorhanden auf der Homepage der Vorlesung

Prüfungsmodalitäten

In der Regel Einzelprüfungen pro Veranstaltung und anschließende Berechnung des Mittelwertes der beiden Einzelleistungen. Die Prüfungen zu den einzelnen Veranstaltungen im Modul werden je nach Teilnehmerzahl als Klausur oder als Fachgespräch durchgeführt. Abweichungen von diesem Schema sowie zusätzliche Forderungen werden bei der Ankündigung der entsprechenden Vorlesungen angegeben.

Modulverantwortliche(r)

Platzner

III.3.5 HW/SW-Codesign

Rolle des Moduls im Studiengang

Eingebettete Systeme spielen die zentrale Rolle im Zuge der sich ständig verstärkenden Informatisierung aller technischen Systeme. Unter Eingebetteten Systemen werden die informationsverarbeitenden Anteile in derartigen Systemen verstanden. Sie bestehen in der Regel aus dedizierter Hardware und darauf aufsetzender Software. Beides wird mit den grundsätzlichen Methoden der Informatik entworfen, wobei das Zusammenspiel zwischen HW und SW eine besonders wichtige Rolle spielt. Eine wesentliche Besonderheit der Eingebetteten Systeme ist aber, dass die physikalischen Gesetze des Gesamtsystems eine bestimmende Rolle spielen und beim Entwurf berücksichtigt werden müssen. Neben Realzeitanforderungen ist hier die Ressourcenbeschränkung (z.B. bezüglich des Stromverbrauchs oder der verfügbaren Chipfläche) zu berücksichtigen. Dies führt dazu, dass der allgemeine Entwurfszyklus von Informatiksystemen in allen Phasen spezifisch angepasst werden muss. So müssen bei der Spezifikation und der Modellierung Realzeit- und Ressourcenbeschränkung beschreibbar sein, was zu spezifischen Formalismen führt. Die abstrakten Modelle müssen im Zusammenspiel mit den umgebenden (z.T. ebenfalls modellierten oder real existierenden) Systemteilen validiert und analysiert werden. Die Partitionierung in Hardware und Software geschieht bei Eingebetteten Systemen weniger im Sinne einer allgemeinen Optimierung sondern gezielt in Richtung der zu erfüllenden Restriktionen. Der Prozess der Synthese von Hardware und Software wird ebenfalls von der Vorgabe dominiert, diese Restriktionen zu respektieren. Da Eingebettete Systeme in der Regel sicherheitsrelevante Anteile beinhalten, teilweise die Sicherheit von Systemen erst gewährleisten, sind hier besonders stringente Verifikationstechniken anzuwenden, die zudem, nicht zuletzt wegen der zwingend erforderlichen Berücksichtigung von Realzeit besonders komplex sind. Da andererseits aber meist vordefinierte endliche Systeme vorliegen, können Verfahren aus der Hardwareverifikation zugrunde gelegt werden.

Dieses Modul ist für Studierende konzipiert, die sich in den Aspekten des Bereichs „Eingebettete Systeme und Systemsoftware“ (ESS) vertiefen wollen, die sich mit dem Zusammenspiel von Hardware- und Softwarekomponenten aber auch mit der Interaktion mit physikalischen Systemen auseinandersetzen. Die spezifische Ausrichtung auf HW/SW-Codesign ermöglicht die Kombination mit allen anderen ESS-Aspekten, wie zum Beispiel Rechnernetzen, Betriebssystemen und Verteilten Systemen oder Eingebettete und Realzeitsysteme im Rahmen des Vertiefungsgebiets. Dieses Modul baut auf den in den Modulen „Konzepte und Methoden der Systemsoftware“ und „Technische Informatik“ vorgestellten Grundlagen auf. Die allgemeinen Prinzipien werden nun auf den gesamtheitlichen Entwurf gemischter HW/SW-Systeme übertragen und durch Fallstudien verdeutlicht. Die besondere Berücksichtigung der durch die physikalischen Gesetze der umgebenden nicht informatischen Systemteile ergeben dabei spezifische Fragestellungen.

Inhaltliche Gliederung der Veranstaltung

Das Modul besteht aus Lehrveranstaltungen über HW/SW-Codesign, Sensorik, Entwurfsmethoden für Eingebettete Systeme, Fortgeschrittene Konzepte der Rechnerarchitektur und fortgeschrittene Konzepte von HW/SW-Codesign. Die Veranstaltung über HW/SW-Codesign behandelt den Entwurfszyklus eines integrierten HW/SW-Entwurfs von der Spezifikation und Modellierung über Analyse und Validierung, die HW/SW-Partitionierung, die HW/SW-Synthese bis hin zur Systemintegration und Verifikation. Der Partitionierung kommt dabei besondere Bedeutung zu. Hier gilt es auch geeignete Schnittstellen zu entwerfen. Mit einer Veranstaltung über Sensorik werden spezielle

Aufgabenstellungen behandelt, die bei Eingebetteten Systemen besonderes Gewicht und eine spezielle Ausprägung haben. Es gilt, den Zustand des umgebenden Systems möglichst lückenlos aufzunehmen und einer darauf resultierenden Verarbeitung zuzuführen. Hier müssen alle Aspekte von der elementaren Messdatenaufnahme bis hin zu komplexen Filteralgorithmen berücksichtigt werden. Natürlich hat die Sensorik eine spiegelbildliche Entsprechung in der Aktorik, die ebenfalls behandelt wird. Den wichtigen Gesichtspunkt der Entwurfsmethoden für Eingebettete Systeme deckt eine weitere Veranstaltung ab. Hier werden über die speziellen Fragestellungen des HW/SW_Codesign hinaus Aspekte des gesamten Entwurfszyklus behandelt. Eingebettete Systeme setzen oft auf spezielle und fortgeschrittene Konzepte der Rechnerarchitektur, z.B. unter Verwendung rekonfigurierbarer Hardware auf. Darauf wird in einer Veranstaltung über Fortgeschrittene Konzepte der Rechnerarchitektur eingegangen, während die Veranstaltung über Fortgeschrittenen Konzepte von HW/SW-Codesign die Grundprinzipien dieser Technik ebenfalls verfeinert und erweitert. Auch hier spielen rekonfigurierbare Hardwarebausteine eine zunehmend wichtige Rolle.

Inhaltliche Verwendbarkeit

Typische Anwendungsfelder sind in der Anwendungsentwicklung technischer Systeme sowie beim Entwurf- und der Umsetzung von Spezialsystemen zu finden. Die vorgestellten Methoden zur Spezifikation, Modellierung, Analyse, HW/SW-Partitionierung, Synthese und Verifikation werden in allen Anwendungsbeichen eingebetteter Systeme, dh. im gesamten Bereich technischer Systeme benötigt. Auch Lösungen im klassischen Umfeld der Informationsverarbeitung lassen sich aufgabenspezifisch durch eine geschickte Partitionierung in HW- und SW-Anteile optimieren. Grundsätzlich lässt sich ein Algorithmus nicht nur in SW sondern auch mittels einer dedizierten HW-Lösung implementieren. Dies stellt für die Studierenden einen nicht zu unterschätzenden Erkenntnisgewinn dar.

Voraussetzungen und Vorkenntnisse

Als Voraussetzung dient die Veranstaltung über HW/SW-Codesign aus dem 2. SA des BSc-Studiengangs. Als weitere Vorkenntnisse Als werden die Inhalte der Module Technische Informatik vorausgesetzt. Ferner sind grundlegende Kenntnisse von Modellierungsprinzipien aus dem Modul Modellierung und der Programmiersprachen aus SWE zwingend erforderlich. Darüber hinaus wird die Bereitschaft erwartet, sich in systemnahe Programmiersprachen und in Hardwarebeschreibungssprachen einzuarbeiten.

Prüfbare Standards / Lernziele der Veranstaltung

Die Studierenden sollen das Verständnis der spezifischen Eigenschaften von Eingebetteten Systemen erlangen sowie die elementaren Konzepte zum Entwurf derartiger Systeme als gemischte HW/SW-Implementierungen kennen lernen. Die Studierenden sollen Kriterien für die Partitionierung in HW/SW kennen lernen und das Instrumentarium zur Durchführung dieser Partitionierung beherrschen können. Sie sollen in der Lage sein, die spezifischen Restriktionen, die sich durch die physikalischen Gesetze des umbebenden Systems ergeben, einzuschätzen und lernen, diese gezielt in den Entwurfsprozess einzubeziehen. Schließlich sollen sie lernen, wie spezifische Methoden aus der Softwaretechnik einerseits und dem Hardwareentwurf andererseits zu einer leistungsfähigen Entwurfsmethodik kombiniert werden können.

Vermittlung von Faktenwissen – Inhaltskompetenz

- Zusammenhang zwischen informatischen und physikalischen Systemkomponenten
- HW/SW-Architekturvarianten für eingebettete und Realzeitsysteme

- Techniken der HW/SW-Partitionierung
- Techniken zur Validierung und Verifikation
- Techniken für den integrierten Entwurf gemischter HW/SW-Systeme

Vermittlung von methodischem Wissen – Methodenkompetenz

- Methoden zur Charakterisierung von Algorithmen bzgl. Implementierungstechnik
- Methoden zur technischen Interaktion mit physikalischen Systemen
- Methoden zur Verifikation zeitbehafteter HW/SW-Systeme
- Methoden für den zielgerichteten Entwurf dedizierter HW-Architekturen

Vermittlung von Transferkompetenz

Übertragung der globalen Strategien auf vorgegebene Einzelsituationen, zum Beispiel im Rahmen von Übungsaufgaben

Vermittlung von normativ-bewertender Kompetenz

- Strategien zur Behandlung unterschiedlicher Strategien ausarbeiten
- Den praktischen Wert der Konzepte und Methoden der eingebetteten Systeme erkennen

Schlüsselqualifikationen

Durch den Übungsbetrieb in kleinen Gruppen wird die Kooperations- und Teamfähigkeit gefördert.

Modulzugehörigkeit

Wahlmodul im Gebiet „Eingebettete Systeme und Systemsoftware“

Modus

- Leistungspunkte pro Modul (Workload) : 8
- Leistungspunkte der Veranstaltung: jeweils 4
- SWS (2V+1Ü, 2V+1Ü)
- Häufigkeit des Angebotes: 2-4 Veranstaltungen pro Jahr im WS und SS
- Dauer (2 Semester)

Methodische Umsetzung

Übungen in kleinen Gruppen fördern die praktische Anwendung der vorgestellten Methoden auf ausgewählte Beispiele. Insbesondere müssen Parameter und Strategien an die konkrete Situation angepasst werden. Die Aufgaben werden in Gruppen von 3 Studierenden bearbeitet, wodurch die Teamfähigkeit gefördert wird. Der Aufbau der Aufgabenzettel bildet den Aufbau der Systemsoftware nach, beginnend von der Hardware über Prozesse bis zu Betriebsmittelverwaltung und Scheduling.

Organisationsformen / Medieneinsatz / Literaturangaben

- Folien-orientierte Vorlesung, Tafelanschrieb bei Beispielen, zusätzlichen Erläuterungen und zu potenzierenden Sachverhalten
- Wöchentliche Übungen in kleinen Gruppen. Dabei werden Präsenzaufgaben sowie die Musterlösungen zu den Hausaufgaben vorgerechnet
- Erwartete Aktivitäten der Studierenden: Mitarbeit bei Präsenzübungen, Hausaufgaben, selbständiges Studium von sekundärer Literatur
- Folienskript ist vorhanden auf der Homepage der Vorlesung

Prüfungsmodalitäten

In der Regel Einzelprüfungen pro Veranstaltung und anschließende Berechnung des Mittelwertes der beiden Einzelleistungen. Die Prüfungen zu den einzelnen Veranstaltungen im Modul werden je nach Teilnehmerzahl als Klausur oder als Fachgespräch durchgeführt. Abweichungen von diesem Schema sowie zusätzliche Forderungen werden bei der Ankündigung der entsprechenden Vorlesungen angegeben.

Modulverantwortliche(r)

Platzner

III.3.6 Eingebettete- und Echtzeitsysteme

Rolle des Moduls im Studiengang

Eingebettete Systeme spielen die zentrale Rolle im Zuge der sich ständig verstärkenden Informatisierung aller technischen Systeme. Große Bereiche der Maschinen- Automobil-, Luft- u. Raumfahrtstechnik, aber auch der Kommunikationstechnik sind ohne eingebettete Systeme nicht mehr realisierbar. Unter Eingebetteten Systemen werden die informationsverarbeitenden Anteile in derartigen Systemen verstanden. Sie bestehen in der Regel aus dedizierter Hardware und darauf aufsetzender Software. Beides wird mit den grundsätzlichen Methoden der Informatik entworfen. Eine wesentliche Besonderheit der Eingebetteten Systeme ist aber, dass die physikalischen Gesetze des Gesamtsystems eine bestimmende Rolle spielen und beim Entwurf berücksichtigt werden müssen. Hier sind insbesondere Realzeitanforderungen zu berücksichtigen, wobei dieser Realzeitaspekt auch bei nicht technischen Anwendungen eine wichtige Rolle spielen kann. Dies führt dazu, dass der allgemeine Entwurfszyklus von Informatiksystemen in allen Phasen spezifisch angepasst werden muss. So müssen bei der Spezifikation und der Modellierung Realzeit- und Ressourcenbeschränkung beschreibbar sein, was zu spezifischen Formalismen führt. Die abstrakten Modelle müssen im Zusammenspiel mit den umgebenden (z.T. ebenfalls modellierten oder real existierenden) Systemteilen validiert und analysiert werden. Der Syntheseprozess wird von der Vorgabe dominiert, die genannten Restriktionen zu respektieren. Da Eingebettete Systeme in der Regel sicherheitsrelevante Anteile beinhalten, teilweise die Sicherheit von Systemen erst gewährleisten, sind hier besonders stringente Verifikationstechniken anzuwenden, die zudem, nicht zuletzt wegen der zwingend erforderlichen Berücksichtigung von Realzeit besonders komplex sind.

Dieses Modul ist für Studierende konzipiert, die sich in den Aspekten des Bereichs „Eingebettete Systeme und Systemsoftware“ (ESS) vertiefen wollen, die sich mit der Interaktion mit physikalischen Systemen auseinandersetzen. Darüber hinaus sollen allgemeine Fragen der Realzeitverarbeitung behandelt werden. Die spezifische Ausrichtung auf Eingebettete- und Realzeitsysteme ermöglicht die Kombination mit allen anderen ESS-Aspekten, wie zum Beispiel Rechnernetzen, Betriebssystemen und Verteilten Systemen oder HW/SW-Codesign im Rahmen des Vertiefungsgebiets. Dieses Modul baut auf den in den Modulen „Konzepte und Methoden der Systemsoftware“ und „Technische Informatik“ vorgestellten Grundlagen auf. Die allgemeinen Prinzipien werden nun auf realzeitfähige Systemsoftware und anwendungsspezifische Programmiermodelle übertragen und durch Fallstudien verdeutlicht. Die besondere Berücksichtigung der durch die physikalischen Gesetze der umgebenden nicht informatischen Systemteile ergeben dabei spezifische Fragestellungen.

Inhaltliche Gliederung des Moduls

Das Modul umfasst die Gebiete Eingebettete Systeme, Echtzeitbetriebssysteme (Real Time Operating Systems), Distributed and Parallel Embedded Systems und Intelligenz in Eingebetteten Systemen. Die Veranstaltung über Eingebettete Systeme liefert einen Überblick über die Aufgabenstellung und grundsätzliche Lösungsansätze. Dabei wird fallbasiert vorgegangen. Eine zentrale Bedeutung im Bereich der Eingebetteten Systeme kommt dem Realzeitbetriebssystem dar, daher wird dieser Bereich ausführlich in aufeinander aufbauenden Veranstaltungen behandelt. Zunächst wird aus den Konzepten allgemeiner Betriebssysteme die Besonderheiten der Realzeitbetriebssysteme abgeleitet und die grundsätzlichen Konzepte dargestellt. Der Schwerpunkt liegt dabei auf den Gebieten Realzeit-Scheduling und deterministische Ressourcenverwaltung. Anschließend werden diese Konzepte verfeinert und

mathematisch präzise unterfüttert. Insbesondere werden für die verschiedenen Schedulingverfahren Analysetechniken eingeführt, die eine präzise Vorhersagbarkeit sicherstellen. Die spezifischen Aspekte, die bei der zunehmenden Realisierung Eingebetteter Systeme in Form verteilter Systeme zu berücksichtigen sind, werden in Veranstaltungen über Distributed and Parallel Embedded Systems behandelt. Hier geht es auch darum, realzeitfähige Kommunikationsprotokolle und die verteilte Implementierung streng deterministischer Algorithmen zu diskutieren. Die Veranstaltung über Intelligenz in eingebetteten Systemen spannt den Bogen zu dem erkennbaren Trend zu mehr autonomen, selbstorganisierenden Systemen bis hin zu Evolving bzw. Organic Systems. Den wichtigen Gesichtspunkt der Entwurfsmethoden für Eingebettete Systeme deckt eine weitere Veranstaltung ab.

Inhaltliche Verwendbarkeit

Typische Anwendungsfelder sind in der Anwendungsentwicklung technischer Systeme sowie beim Entwurf- und der Umsetzung von Spezialsystemen zu finden. Die vorgestellten Methoden zur Spezifikation, Modellierung, Analyse, Synthese und Verifikation werden in allen Anwendungsbereichen eingebetteter Systeme, d.h. im gesamten Bereich technischer Systeme benötigt. Realzeitanwendungen finden aber auch im nichttechnischen Umfeld Verwendung, z.B. in der Wetterprognose oder in der strategischen Planung von Finanzdienstleistungen. Über diesen Anwendungsbezug hinaus eröffnet die Beschäftigung mit eingebetteten und Realzeitsystemen aber auch einen nicht zu vernachlässigenden Erkenntnisgewinn, da man gezwungen ist, sich von der Fiktion des Idealismus im Platonschen Sinn zu lösen und sich mit physikalischen Randbedingungen, insbesondere einer von der Umgebung prädestinierten Zeitablauf auseinanderzusetzen.

Voraussetzungen und Vorkenntnisse

Als Voraussetzung für das Modul Eingebettete Systeme und Echtzeitsysteme dienen die Veranstaltungen über Eingebettete Systeme oder über HW/SW-Codesign aus dem Bachelormodul Eingebettete Systeme und Systemsoftware. Als weitere Vorkenntnisse werden die Inhalte der Module Technische Informatik und KMS vorausgesetzt. Ferner sind grundlegende Kenntnisse von Modellierungsprinzipien aus dem Modul Modellierung und der Programmiersprachen aus SWE zwingend erforderlich. Darüber hinaus wird die Bereitschaft erwartet, sich in systemnahe Programmiersprachen einzuarbeiten.

Prüfbare Standards / Lernziele der Veranstaltung

Die Studierenden sollen das Verständnis der spezifischen Eigenschaften von Eingebetteten Systemen erlangen sowie die elementaren Konzepte zum Entwurf derartiger Systeme kennen lernen. Die Studierenden sollen mögliche Gefahren im Falle des fehlerhaften Entwurfs eingebetteter Systeme erkennen und das Instrumentarium zur Vermeidung solcher Fehler beherrschen können. Sie sollen in der Lage sein, die spezifischen Restriktionen, die sich durch die physikalischen Gesetze des umgebenden Systems ergeben, einzuschätzen und lernen diese gezielt in den Entwurfsprozess einzubeziehen. Schließlich sollen die Kernmethoden für die Sicherstellung eines präzise vorhersagbaren Systemverhaltens verstanden und an konkreten Beispielen angewendet werden.

Vermittlung von Faktenwissen – Inhaltskompetenz

- Zusammenhang zwischen informatischen und physikalischen Systemkomponenten
- Implementierungsvarianten für eingebettete und Realzeitsysteme
- Techniken der Realzeitverwaltung
- Techniken zur Validierung und Verifikation

- Techniken für den Entwurf von eingebetteten und Realzeitsystemen

Vermittlung von methodischem Wissen – Methodenkompetenz

- Methoden zur vorhersagbaren Einplanung von Betriebsmitteln
- Methoden zur logischen Interaktion mit physikalischen Systemen
- Methoden zur Verifikation zeitbehafteter Systeme
- Methoden für den Entwurf von Systemen mit inhärenter Intelligenz

Vermittlung von Transferkompetenz

Übertragung der globalen Strategien auf vorgegebene Einzelsituationen, zum Beispiel im Rahmen von Übungsaufgaben

Vermittlung von normativ-bewertender Kompetenz

- Strategien zur Behandlung unterschiedlicher Strategien ausarbeiten
- Den praktischen Wert der Konzepte und Methoden der eingebetteten Systeme erkennen

Schlüsselqualifikationen

Durch den Übungsbetrieb in kleinen Gruppen wird die Kooperations- und Teamfähigkeit gefördert.

Modulzugehörigkeit

Wahlmodul im Gebiet „Eingebettete Systeme und Systemsoftware“

Modus

- Leistungspunkte pro Modul (Workload) : 8
- Leistungspunkte der Veranstaltung: jeweils 4
- SWS (2V+1Ü, 2V+1Ü)
- Häufigkeit des Angebotes: 2-4 Veranstaltungen pro Jahr im WS und SS
- Dauer (2 Semester)

Methodische Umsetzung

Übungen in kleinen Gruppen fördern die praktische Anwendung der vorgestellten Methoden auf ausgewählte Beispiele. Insbesondere müssen Parameter und Strategien an die konkrete Situation angepasst werden. Die Aufgaben werden in Gruppen von 3 Studierenden bearbeitet, wodurch die Teamfähigkeit gefördert wird. Der Aufbau der Aufgabenzettel bildet den Aufbau der Systemsoftware nach, beginnend von der Hardware über Prozesse bis zu Betriebsmittelverwaltung und Scheduling.

Organisationsformen / Medieneinsatz / Literaturangaben

- Folien-orientierte Vorlesung, Tafelanschrieb bei Beispielen, zusätzlichen Erläuterungen und zu potenzierenden Sachverhalten
- Wöchentliche Übungen in kleinen Gruppen. Dabei werden Präsenzaufgaben sowie die Musterlösungen zu den Hausaufgaben vorgerechnet
- Erwartete Aktivitäten der Studierenden: Mitarbeit bei Präsenzübungen, Hausaufgaben, selbständiges Studium von sekundärer Literatur
- Folienskript ist vorhanden auf der Homepage der Vorlesung

Prüfungsmodalitäten

In der Regel Einzelprüfungen pro Veranstaltung und anschließende Berechnung des Mittelwertes der beiden Einzelleistungen. Die Prüfungen zu den einzelnen Veranstaltungen im Modul werden je nach Teilnehmerzahl als Klausur oder als Fachgespräch durchgeführt. Abweichungen von diesem Schema sowie zusätzliche Forderungen werden bei der Ankündigung der entsprechenden Vorlesungen angegeben.

Modulverantwortliche(r)

Rammig

III.4 Gebiet Mensch-Maschine-Wechselwirkung

III.4.1 Grafische Datenverarbeitung

Rolle im Studiengang

Das Modul „Grafische Datenverarbeitung“ widmet sich der Erzeugung computer-generierter Bilder durch Szenebeschreibungen, simulierter, gemessener oder empirischer Daten, sowie der Aufnahme, Analyse, Interaktion, und dem Austausch von Bilddaten. Es gehört zu den Modulen im Teilbereich Mensch-Maschine-Wechselwirkung (MMWW).

Inhaltliche Gliederung des Moduls:

Der Modul setzt sich zusammen aus

- einer Grundveranstaltung Computergrafik II mit folgenden Inhalten:
 - Raytracing
 - Radiosity
 - Volume Rendering
 - Advanced Modelling
 - Texture Mapping
 - Image-Based Rendering
 - Non-photorealistic Rendering
 - Animation
- einer Reihe von Ergänzungsveranstaltungen, von denen eine gewählt werden muss
 - Digitale Bildverarbeitung
 - Computergenerierte Visualisierung
 - AR/VR (Augmented Reality/Virtuelle Realität) (geplant)
 - Seminar Ausgewählte Themen aus der Computergrafik
 - Seminar Ausgewählte Themen aus der Digitalen Bildverarbeitung
 - Seminar Ausgewählte Themen aus der Visualisierung

Die Ergänzungsveranstaltungen sind inhaltlich z. B. wie folgt gegliedert:

- Computergenerierte Visualisierung mit den Teilgebieten
 - Grundlagen, Definitionen
 - Daten und Datenmodelle
 - Betrachter und Aufgaben (Tasks)
 - Mapping (Abbildung von Daten auf Bilder)
 - Repräsentation
 - Interaktionsfluss
- Digitale Bildverarbeitung mit den Teilgebieten

- Charakterisierung digitaler Bilder (Rasterung, Quantisierung etc.)
 - Punktoperationen und Filteroperationen im Ortsbereich
 - Transformationen
 - Operationen im Frequenzbereich
 - Bildkompression
 - Bildsegmentierung
 - Bildrestauration.
- Die anderen Veranstaltungen werden inhaltlich nach Bedarf festgelegt.

Inhaltliche Verwendbarkeit

Die Methoden der photorealistischen Darstellung sind ein sehr aktueller und dynamischer Bereich der Informatik. Die Veranstaltung Computergrafik II birgt Kenntnisse zum state-of-the-art des photorealistischen Renderings und legt die notwendigen Grundlagen um zukünftige Entwicklungen im Bereich der Computergrafik zu verstehen. Da Datenmengen beständig zunehmen (z.B. medizinische Daten, Daten aus Weltraummissionen, statistische Daten, wissenschaftliche Berechnungen, etc.) und in vielen Fällen von Menschen (z.B. Chirurgen, Geologen, Umweltpolitiker, Sozialwissenschaftler, etc.) schnell und korrekt interpretiert werden sollen, sind systematische Strategien zur Umwandlung von Daten in ausdrucksvolle und wirksame Bilder (oder Bildsammlungen) erforderlich. Damit beschäftigt sich die Veranstaltung computergenerierte Visualisierung. Um die so entstandenen Bilddaten in Qualität und Quantität zu charakterisieren, Transformationen und Operationen zur Bildverbesserung, Bildmanipulation und zum Bildtransfer einzusetzen, bietet die Veranstaltung Digitale Bildverarbeitung die notwendigen Grundlagen, die entsprechenden Algorithmen zu verstehen.

Voraussetzungen und Vorkenntnisse

Teilnahme an Computergrafik I.

Lernziele der Veranstaltung

Studierende lernen...

Vermittlung von Faktenwissen

... über:

- siehe Inhaltsverzeichnis

Vermittlung von methodischem Wissen

- die methodischen Grundlagen der Algorithmen
- effiziente Algorithmen vs. photorealistische Algorithmen
- praktische Anwendung der Methoden am Computer
- strategisches Vorgehen in der Umwandlung von Daten in Bilder unter dem Gesichtspunkt der Interpretation durch den Menschen
- Transformationen in unterschiedliche Bildräume
- Kompressionsalgorithmen

- Praktische Durchführung der Algorithmen am Computer: ein wesentlicher Schritt um die Problematik des Wechsels von Theorie in die Praxis zu begreifen.

Vermittlung von Transferkompetenz

Kenntnisse der Computergrafik und Bildverarbeitung ermöglichen die Erstellung wirksamerer Visualisierungen für Anwendungsbereiche wie Medizin, Biologie, Chemie, und viele mehr. Bildsegmentierung ist ein Vorverarbeitungsschritt für die Robotik.

Vermittlung von normativ-bewertender Kompetenz

- Bewertung der Effizienz von Computergrafikalgorithmen
- Bewertung der Qualität einer Grafikkarte
- Bewertung der Qualität eines Bildes für eine bestimmte Zielgruppe von Personen und ein bestimmtes Visualisierungsziel
- Bewertung des Qualitätsverlustes bei Kompression von Bildern

Schlüsselqualifikationen

- Fähigkeit zur Nutzung moderner IuK-Technologien
- fachbezogenen Fremdsprachenkompetenzen durch englische Begleitliteratur
- Kooperations- und Teamfähigkeit durch Gruppenprojekte

Modulzugehörigkeit:

Wahlmodul im Gebiet Mensch-Maschine-Wechselwirkung (MMWW)

Modus

- Leistungspunkte: 4 + 4 ECTS (2 Katalogveranstaltungen)
- SWS: 2+1, 2+1 (oder 2+1, 2 im Falle eines Seminars)
- Häufigkeit des Angebotes: 2-3 Katalogveranstaltungen pro Jahr im WS und SS

Methodische Umsetzung

Die theoretischen Konzepte werden in Präsenzübungen in Kleingruppen vertieft. Die Methoden werden in praktischen Übungen erprobt.

Organisationsformen / Medieneinsatz / Literaturangaben

- Bei Veranstaltungen eine Doppelstunde Vorlesung pro Woche, und eine Doppelstunde Präsenzübung jede zweite Woche, oder Lösen von Programmieraufgaben in eigener Zeit in ähnlichem Ausmaß
- Eingesetzte Software: derzeit OpenGL, für Bildverarbeitung IDL (von Research Systems, Inc.)
- Bei Seminaren entweder semesterbegleitendes oder Blockseminar, je nach Ankündigung
- Hinweise auf erwartete Aktivitäten der Studierenden: Bereitschaft, die mathematischen Kenntnisse der Vergangenheit wieder zu aktivieren; selbständiges Programmieren; Mitarbeit bei den Präsenzübungen.
- Eingesetzte Materialien: Powerpoint Folien zum Herunterladen und Übungsblätter
- Literaturhinweise zur Veranstaltung Computergrafik: Angel, Interactive Computer Graphics, Addison-Wesley; oder Watt, Three Dimensional Computer Graphics, Addison-Wesley; oder Foley et al., Computer Graphics, Addison Wesley Verlag.

- Literaturhinweise zur Veranstaltung Digitale Bildverarbeitung: Gonzalez /Woods, Digital Image Processing, Addison Wesley Verlag.
- Literaturhinweise zur Veranstaltung Computergenerierte Visualisierung: Webbasiertes Veranstaltungsskript

Prüfungsmodalitäten

- Klausur
- Eigenständiges Programmieren (von Teilen in) der Rendering Pipeline oder den Ergänzungsthemen angepasste Aufgaben
- Angaben zur Notenermittlung: Die Gewichtung von Klausur und Programmieraufgaben/-projekten wird Anfang des Semesters bekannt gegeben.

Modulverantwortliche(r)

Domik

III.4.2 Informatik und Gesellschaft

Rolle im Studiengang

Informatiker entwickeln auf Zeichen basierende Produkte (Programme, Spezifikationen, Dokumentationen etc.). Im Gegensatz zu anderen Ingenieurprodukten, die aus Materialien wie Stahl, Kunststoff oder Glas gefertigt werden, bildet Software soziale Wirklichkeit in vielfältiger Form ab. Durch den Einsatz ändert sich diese Wirklichkeit. Das führt zu vielfältigen Wechselwirkungen zwischen Informatiksystemen und ihrem Einsatzumfeld hinsichtlich Verständnis, Nutzungspotential und Einsatzrisiken. Ausgehend von diesen Besonderheiten der Informatik werden in der Veranstaltung maschinelle Datenverarbeitung (Produkt) und menschliche Informationsverarbeitung (Prozess) einander gegenüber gestellt und die daraus entstehenden Konsequenzen für die Gestaltung von Informatiksystemen auf allen Ebenen der Entwicklung und des Gebrauchs behandelt. Vertiefende Themenbereiche sind dabei:

- Kulturgeschichte der Datenverarbeitung
- Gestaltung als Anpassung
- Informatik und Militär
- Sozial orientierte Systemgestaltung
- Multimedia und Gesellschaft

Das Modul gehört zu den Modulen im Teilbereich Mensch-Maschine-Wechselwirkung (MMWW)

Inhaltliche Gliederung des Moduls:

Der Modul setzt sich zusammen aus

- einer Grundveranstaltung Informatik und Gesellschaft mit folgenden Inhalten:
 - Besonderheiten von Software als Ingenieurprodukt
 - Mensch-Maschine-Wechselwirkungen
 - Biologische Informationsverarbeitung
 - Fehler in technischen und natürlichen Systemen
 - Artefakte als externes Gedächtnis
 - Nicht-selbstbewusste und selbstbewusste Gestaltungsprozesse
 - Produkt-Prozess-Komplementarität
 - Informatik und Militär
 - Atomkrieg aus Versehen
 - Software-Entwicklung als Lernprozess
 - Verantwortung des Informatikers
- einer Reihe von Ergänzungsveranstaltungen, von denen eine gewählt werden muss
 - Konzepte digitaler Medien
 - Seminar Zwischen Science und Fiction
 - Seminar Kulturgeschichte der Datenverarbeitung

- Seminar Urheberrecht und digitale Medien

Die Ergänzungsveranstaltungen sind inhaltlich z. B. wie folgt gegliedert:

- Konzepte digitaler Medien
 - Instrument, Maschine, Automat, interaktives System
 - Hypertext
 - Groupware
 - Virtuelle Gemeinschaften
 - Virtuelle Realität, Erweiterte Realität
 - Ambient Computing, Mediatronik
- Die Seminarthemen werden inhaltlich nach Bedarf festgelegt.

Inhaltliche Verwendbarkeit

Die Studierenden erwerben grundlegende Kenntnisse über die Möglichkeiten und Grenzen der Gestaltung von Informatiksystemen und der Formalisierung. Diese Einsichten sind sowohl zur Bewertung technischer Potenziale erforderlich als auch für leitende Tätigkeiten bei der Abwicklung von Softwareprojekten. Das Studium der Wechselwirkungen schafft zudem ein vertieftes Verständnis für Probleme und Potenziale der IT in verschiedenen Anwendungskontexten. Historische Betrachtungen zur Geschichte der Datenverarbeitung ordnen aktuelle Konzepte der Informatik in einen größeren kulturgeschichtlichen Zusammenhang ein.

Voraussetzungen und Vorkenntnisse

Keine.

Lernziele der Veranstaltung

Studierende lernen...

Vermittlung von Faktenwissen

... über:

- Geschichte der Datenverarbeitung
- Mensch-Maschine-Wechselwirkungen
- Biologische Informationsverarbeitung
- Kulturelle und soziale Gestaltungsprozesse
- Informatik und Militär
- Prozessorientierter Softwaregestaltung

Vermittlung von Methodenkompetenz

- Erwägen unterschiedlicher Konzepte der technischen und biologischen Informationsverarbeitung
- Erschließung interdisziplinärer Zugänge zum Themenfeld Mensch-Maschine
- Behandlung ethischer Fragestellungen
- Anwenden der Produkt-Prozess-Komplementarität auf unterschiedliche Fragestellungen

Vermittlung von Transferkompetenz

- Beurteilung formaler und informeller Verfahren
- Erkennen und Auflösen von Designkonflikten
- Gestaltung fehlerfreundlicher Entwicklungsumgebungen
- Projektmanagementkompetenzen für die Prozessgestaltung
- Erschließung interdisziplinärer Literatur

Vermittlung von normativ-bewertender Kompetenz

- Bewertung der Grenzen der Formalisierung
- Bewertung von einschlägigen Gesetzestexten

Schlüsselqualifikationen

- Grundlegende Gestaltungs- und Präsentationskompetenzen
- Aneignung fachfremder Konzepte
- Kooperations- und Teamfähigkeit durch Gruppenarbeit

Modulzugehörigkeit:

Wahlmodul im Gebiet Mensch-Maschine-Wechselwirkung (MMWW)

Modus

- Leistungspunkte: 4 + 4 ECTS (2 Katalogveranstaltungen)
- SWS: 2+1, 2+1 (oder 2+1, 2 im Falle eines Seminars)
- Häufigkeit des Angebotes: 2-3 Katalogveranstaltungen pro Jahr im WS und SS

Methodische Umsetzung

Themen werden in Gruppenarbeit umgesetzt und ausgearbeitet. Die Präsentation erfolgt in Form der Gestaltung multimedialer Wissensräume wodurch Präsentationskompetenzen über einen längeren Zeitraum anhand einer zusammenhängenden Fragestellung angeeignet werden. Ein wichtiger Aspekt ist dabei die Recherche und Erschließung von Literatur aus anderen Disziplinen.

Organisationsformen / Medieneinsatz / Literaturangaben

- Bei Veranstaltungen eine Doppelstunde Vorlesung pro Woche, und eine Doppelstunde Präsenzübung jede zweite Woche,
- Erarbeiten von Themen durch Aufbereitung interdisziplinärer Literatur in eigener Zeit in ähnlichem Ausmaß
- Bei Seminaren (entweder semesterbegleitend oder als Blockseminar) Ausarbeitung eines speziellen Themenbereiches und Präsentation in der Veranstaltung
- Hinweise auf erwartete Aktivitäten der Studierenden: Bereitschaft, über den traditionellen Tellerrand der Informatik hinauszuschauen und sich Konzepte und Ansätze aus anderen Disziplinen anzueignen; Mitarbeit bei den Präsenzübungen.
- Eingesetzte Materialien: Powerpoint Folien zum Herunterladen und Übungsblätter, Gesetzestexte und Fremdmaterialien, Audio-Annotationen zur Vorlesung.

Prüfungsmodalitäten

- Klausur und mündliche Prüfungen
- Angaben zur Notenermittlung: Die Gewichtung von Klausur und Übungsleistungen wird Anfang des Semesters bekannt gegeben.

Modulverantwortliche(r)

Keil

III.4.3 Konzepte digitaler Medien

Rolle im Studiengang

Die klassischen Medientheorien sind vorrangig Rezeptionsanalysen von Massenmedien unter der besonderen Berücksichtigung von Film und Fernsehen. Mathematische Formeln, technische Zeichnungen oder Verwaltungsformulare werden in der Medientheorie nicht betrachtet. Durch den Computer werden jedoch diese Grenzen aufgelöst. Digitale Medien verbinden potenziell alle bislang gekannten Medienformen, wenn auch nicht in der gleichen Qualität und mit den gleichen Produktions- und Rezeptionsbedingungen.

Über den Begriff des Zeichens und seiner Verarbeitung mit Hilfe von (digitalen) Automaten erhält man einen erweiterten Medienbegriff, der es gestattet, die Vielfalt digitaler Medien unter einem gemeinsamen technischen Bezugspunkt zu thematisieren. Analog zu den Konzepten von Programmiersprachen lassen sich unterschiedliche Ausprägungen digitaler Medien vergleichen und die jeweiligen medialen Mehrwerte bestimmen. Dies ist für alle Anwendungsbereiche, die heutzutage mit dem e-Präfix versehen sind (e-Learning, e-Government, e-Business, etc) von entscheidender Bedeutung.

Das grundlegende Verständnis des Computers als digitalem Medium ist für die zukünftige Entwicklung der Informationstechnologien von entscheidender Bedeutung. Es gestattet zugleich die Positionierung der Informatik und der informatikrelevanten Forschungs- und Entwicklungsanteile in neu sich bildenden interdisziplinären Studiengängen und Forschungsthemen. Es stellt zugleich anschlussfähiges Wissen für eine interdisziplinäre Arbeit insbesondere mit den Medienwissenschaften, der Pädagogik und der Psychologie zur Verfügung und schafft zugleich ein vertieftes Verständnis der eigenen technischen Grundlagen.

Inhaltliche Gliederung des Moduls:

Der Modul umfasst sowohl Veranstaltungen, die sich mit der Gestaltung interaktiver Systeme und kooperativer Medien befassen als auch mit grundlegenden Gestaltungs- und Medienkonzepten, mit denen es möglich ist, kognitive und mediale Mehrwerte ebenso wie Defizite zu thematisieren. Die gleichnamige Veranstaltung „Konzepte digitaler Medien“ legt dazu die theoretischen und begrifflichen Grundlagen. Die weiteren Veranstaltungen vertiefen und differenzieren diese Konzepte in Bezug auf unterschiedliche Aufgabenbereiche. Zum Modul gehören die Veranstaltungen:

- Konzepte digitaler Medien (Pflicht)
- Software-Ergonomie
- Architekturen von CSCW-Systemen
- Gestaltung interaktiver Systeme

Inhaltliche Verwendbarkeit

Ein Grundverständnis der Wirkungsweise digitaler Medien in Bezug auf kognitive und kulturell-soziale Prozesse ist für den produktiven Einsatz des Computers in einer vernetzten Arbeits- und Lebenswelt unverzichtbar. Insbesondere die Fähigkeit technische Problemstellungen und Konzepte von nicht technischen abzugrenzen ist eine wichtige Voraussetzung für die Analyse von Anforderungen und die Entwicklung transparenter Anwendungssysteme. Vertiefte Kenntnisse über spezielle Architekturen ebenso wie über die

grundlegende Vergleichbarkeit funktionaler, interaktiver und kooperativer Systemkonzepte sind für Informatiker in unterschiedlichen Anwendungskontexten erforderlich.

Voraussetzungen und Vorkenntnisse

Vorausgesetzt werden Grundkenntnisse aus den ersten vier Semestern.

Lernziele der Veranstaltung

Die Studierenden sollen lernen, technische und nicht technische Problemstellungen zu differenzieren und adäquat aufeinander zu beziehen. Des Weiteren sollen sie in der Lage sein, mit Hilfe grundlegender anwendungsbezogener aber nicht anwendungsspezifischer Konzepte Anforderungen aus einem medienbezogenem Einsatzumfeld zu erheben, in Frage kommende Systemarchitekturen zu bewerten und zu vergleichen sowie neue Innovationspotenziale im Medienbereich abschätzen zu können. Die Vermittlung und Anwendung kognitionswissenschaftlicher Grundlagen soll sie in die Lage versetzen, technische und nicht-technische Konzepte konstruktiv miteinander zu verbinden.

Modulzugehörigkeit

Mensch-Maschine-Wechselwirkung.

Modus

- Leistungspunkte : 4 + 4
- SWS 2 + 1, 2 + 1
- Häufigkeit des Angebotes: Basisveranstaltung jedes WS, die weiterführenden Veranstaltungen jedes SS.

Prüfungsmodalitäten

Je nach Teilnehmerzahl mündliche Prüfung oder Klausur

Modulverantwortliche(r)

Keil

III.4.4 Computer gestützte kooperative Zusammenarbeit und Lernen

Rolle im Studiengang

Kooperationsunterstützende Systeme spielen eine größer werdende Rolle in weiten Bereichen menschlicher Zusammenarbeit und des menschlichen Lernens. Entsprechend umfasst das Forschungsfeld CSCW (Computer Supported Cooperative Work) bzw. CSCL (Computer Supported Cooperative Learning) sowohl Werkzeuge und Systeme, aber auch Theorien und Ansätze der kooperativen Mediennutzung. Die Veranstaltung versetzt Informatiker in die Lage den State-of-the-Art des Forschungsfeldes CSCW/L bewerten zu können, sowie die Grundlagen der Klassifikation, architektonischen Entwicklungslinien und verschiedenen Formen der Unterstützung menschlicher Zusammenarbeit einordnen und gegeneinander abgrenzen zu können. Das Modul gehört zu den Modulen im Teilbereich Mensch-Maschine-Wechselwirkung (MMWW)

Inhaltliche Gliederung des Moduls:

Der Modul setzt sich zusammen aus

- Einführungsveranstaltung CSCW mit folgenden Inhalten:
 - Grundlagen der Begrifflichkeiten: CSCW/CSCL/Groupware
 - Medien zwischen Produktion und Kommunikation
 - Von der Interaktion zur Kooperation: Überblick zu den Forschungsfeldern CSCW/CSCL/Groupware
 - Grundlagen der kooperativen Mediennutzung, Medienfunktionen
 - Nicht sequentielles Schreiben: Hypertext
 - Gegenseitige Wahrnehmung, Media Spaces, Awareness
 - Virtuelle Gemeinschaften: MUD/MOO
 - Vorstellung verschiedener Systeme: BSCW, Notes, Netmeeting, Groove, etc.
 - Virtuelle Wissensräume/CSCL
 - Technisch-architektonische Grundlagen kooperationsunterstützender Systeme
 - Grundlegende Architekturkonzepte replizierter und zentralistischer Systeme
 - Mobile Computing, Dynamische Konfiguration
- einer Reihe von Ergänzungsveranstaltungen, von denen eine gewählt werden muss
 - Ergänzungsveranstaltungen aus dem Modul „Mensch-Computer-Interaktion“
 - Konzepte digitaler Medien
 - Architekturen CSCW
 - Seminar: Mobile Ad-Hoc Vernetzung kooperativer Wissensräume
 - Seminar: Virtuelle Wissensräume - CSCL

Die Ergänzungsveranstaltungen sind inhaltlich z. B. wie folgt gegliedert:

- Konzepte digitaler Medien (Reinhard Keil-Slawik)
 - Instrument, Maschine, Automat, interaktives System
 - Hypertext
 - Groupware
 - Virtuelle Gemeinschaften
 - Virtuelle Realität, Erweiterte Realität
 - Ambient Computing, Mediatronik
- Siehe Ergänzungsveranstaltungen „Mensch-Computer-Interaktion“
- Architekturen kooperationsunterstützender Systeme - CSCW/CSCL
 - Von der Interaktion zur Kooperation: Überblick zu den Forschungsfeldern CSCW/CSCL/Groupware
 - Historische Entwicklung verschiedener CSCW-Architekturkonzepte

- Vom Screen-Sharing zu gemeinsamen Informationsräumen
- Architektonische Entwurfsmuster kooperationsunterstützender Systeme
 - Grundlegende Protokolle und Standards
 - Replizierte Architekturen
 - Zentralistische Architekturen – kooperatives MVC-Konzept
 - Architekturkonzepte MUDs und MOOs
 - Inter-Applikationskommunikation – verschiedene Standards
- Für den Bereich kooperationsunterstützender Systeme relevante Architekturkonzepte des WWW
 - Semantic Web
 - Webservices
- Technologische Grundkonzepte und Metaphern kooperationsunterstützender Systeme
 - Virtuelle Räume
 - Gruppen und Zugriffsrechte
 - Floor-Control
 - Collaboration Unaware/Aware - Collaboration Transparent
 - Nebenläufigkeitskontrolle
 - Versionierung
 - Persistenzschicht
- Frameworks kooperationsunterstützender Systeme
 - Zentralisierte und Replizierte Architekturansätze
 - Objektorientierte Frameworks
 - Frameworks des Bereiches CSCL(Computer gestütztes kooperatives Lernen)
 - Exemplarische Vorstellung verschiedener innovativer Architekturkonzepte, z.B. Fraunhofer DyCE, Universität Paderborn sTeam
- Mobilitätsaspekte kooperationsunterstützender Systeme
 - Protokolle und Standards (Peer-to-Peer, IPV6)
 - Protokolle der Spontaner Vernetzung
 - Frameworks für Spontane Vernetzung
- Die Seminarthemen werden inhaltlich nach Bedarf festgelegt.

Inhaltliche Verwendbarkeit

Die vermittelten Kenntnisse aus den Bereichen der computergestützten Gruppenarbeit bilden für vielerlei Anwendungsszenarien wichtige Grundlagen. Szenarien umfassen beispielsweise die bürowirtschaftlich motivierten Bereiche der Groupware und des Workflows, und die Bereitstellung netzgestützter Gruppenarbeitsumgebungen. Hinzu kommen Anwendungsfelder der Unterstützung von Forschergruppen in der wissenschaftlichen Arbeit und das weite Feld des E-Learnings.

Voraussetzungen und Vorkenntnisse

Keine.

Lernziele der Veranstaltung

Studierende lernen...

Vermittlung von Faktenwissen

... über:

- verschiedene Groupware Systeme
- architektonische Grundlagen dieser Systeme (z.B. Gruppenstrukturen, Benutzerrechte)
- Musterarchitekturen synchroner und asynchroner Systeme
- Grundlagen der medialen Unterstützung menschlicher Zusammenarbeit

Vermittlung von Gestaltungskompetenz

- Entwickeln von Groupware-Applikationen
- Architekturdesign synchroner und asynchroner Applikationen
- Erstellen verschiedener Modelle von Nutzerrechten und Gruppenstrukturen

Vermittlung von Transferkompetenz

- Gegenüberstellung individueller und kooperativer Formen der Zusammenarbeit
- Differenziertes Verständnis verschiedener Formen und Stufen der Unterstützung menschlicher Zusammenarbeit.
- Erkennen und Auflösen von Designkonflikten in der Gestaltung kooperationsunterstützender Systeme.

Vermittlung von normativ-bewertender Kompetenz

- Bewertung und Evaluation verschiedener Formen der Unterstützung menschlicher Zusammenarbeit
- Bewertung und Evaluation verschiedener Systeme
- Bewertung der Gebrauchstauglichkeit kooperationsunterstützender Systeme

Schlüsselqualifikationen

- Grundlegende Bewertungskompetenzen verschiedener Ansätze und Systeme
- Gestaltungskompetenz verschiedener Architekturkonzepte
- Aneignung fachfremder Konzepte
- Kooperations- und Teamfähigkeit durch Gruppenarbeit

Modulzugehörigkeit:

Wahlmodul im Gebiet Mensch-Maschine-Wechselwirkung (MMWW)

Modus

- Leistungspunkte: 4 + 4 ECTS (2 Katalogveranstaltungen)
- SWS: 2+1, 2+1 (oder 2+1, 2 im Falle eines Seminars)
- Häufigkeit des Angebotes: 2 Katalogveranstaltungen pro Jahr im WS und SS

Methodische Umsetzung

Die Gestaltungskonzepte werden in Kleingruppen angewandt und vertieft. Präsentationskompetenzen werden in Präsenzübungen angeeignet und umgesetzt.

Organisationsformen / Medieneinsatz / Literaturangaben

- Bei Veranstaltungen eine Doppelstunde Vorlesung und eine Präsenzübung jede Woche, oder Lösen von Gestaltungsaufgaben in eigener Zeit in ähnlichem Ausmaß
- Eingesetzte Software: Präsentationssysteme und verschiedene Entwicklungssysteme, Vorstellung verschiedener kooperationsunterstützender Systeme,

- Nutzung des kooperationsunterstützenden Systems sTeam zur Arbeit in Kleingruppen, Bereitstellung von Folien und Lehrmaterialien.
- Bei Seminaren (entweder semesterbegleitend oder als Blockseminar) Ausarbeitung eines speziellen Themenbereiches und Präsentation in der Veranstaltung
- Hinweise auf erwartete Aktivitäten der Studierenden: Mitarbeit bei den Präsenzübungen, Aktiver Wille zur Aufarbeitung und Gestaltung eigenständiger Lösungsansätze und Architekturkonzepte, Wille zur Analyse und Bewertung verschiedener Systeme.
- Eingesetzte Materialien: Powerpoint Folien, kooperationsunterstützendes System sTeam.
- Literaturhinweise zur Veranstaltung Einführung CSCW: Borghoff, U.M., Schlichter, J.H.: Rechnergestützte Gruppenarbeit – Eine Einführung in verteilte Anwendungen. Berlin: Springer 1995. Schwabe, G., Streitz, N., Unland, R.: CSCW Kompendium – Lehr- und Handbuch zum computergestützten kooperativen Arbeiten, Springer 2001, Teufel, S., Sauter, C., Mühlherr, T., Bauknecht, K.: Computerunterstützung für die Gruppenarbeit. Bonn: Addison-Wesley 1995. Greif, I.: Computer Supported Cooperative Work: A Book of Readings. San Mateo: Morgan Kaufmann Publishers, 1988, Hofte, G.H.: Working Apart Together – Foundations for Component Groupware, Telematica Instituut Fundamental Research Series, Vol. 1, Enschede, the Netherlands, 1998.

Prüfungsmodalitäten

- mündliche Prüfungen
- Angaben zur Notenermittlung: Die Gewichtung von mündlicher Prüfung und Übungsleistungen wird Anfang des Semesters bekannt gegeben.

Modulverantwortliche(r)

NN

III.4.5 Entwicklung von Benutzungsschnittstellen

Rolle im Studiengang

Die Verbreitung von Software und Softwarebenutzung nimmt ständig zu. Dementsprechend ist die Entwicklung von Benutzungsschnittstellen eine wesentliche Aufgabe von Softwarehäusern. Es handelt sich dabei um ein vielschichtiges Problem, welches sich mit so verschiedenen Aspekten wie Softwareentwicklung, logischer und graphischer Gestaltung, arbeitsorganisatorischer Einordnung, wahrnehmungspsychologischen Fragestellungen u.ä. befaßt. Das Modul „Entwicklung von Benutzungsschnittstellen“ gehört zu den Modulen im Teilbereich Mensch-Maschine-Wechselwirkung (MMWW). Dieses Modul vermittelt wesentliche Konzepte und Methoden für diese Aufgabe, z.B. Modellierungskonzepte und –techniken, Benutzungsparadigmen und Gestaltungsrichtlinien. Die Ergänzungsveranstaltungen bieten eine Vertiefung in Teilbereiche wie Programmier- und Werkzeuge, Usability Engineering, webbasierte Benutzungsschnittstellen, oder mehr.

Inhaltliche Gliederung des Moduls:

Der Modul setzt sich zusammen aus

- einer Grundveranstaltung „Modellierung von Benutzungsschnittstellen“ mit folgenden Inhalten:
 - Grundlagen von Modellierung
 - Der modellbasierte Entwicklungsprozeß
 - Aufgabenmodellierung
 - Benutzerinteraktionsmodellierung
 - Kontrollmodellierung
- einer Reihe von Ergänzungsveranstaltungen, von denen eine gewählt werden muß:
 - Gestaltung interaktiver Systeme
 - Praxis des Usability Engineering
 - Programming Interactive Systems (in Engl.)
 - Web-Modellierung (geplant)
 - Programming Interactive Web Sites (geplant, in Engl.)
 - Seminar „Aktuelle Themen zu Benutzungsschnittstellen“

Inhaltliche Verwendbarkeit

Die erworbenen Kenntnisse und Fähigkeiten sind notwendig, um die Entwicklung von Benutzungsschnittstellen adäquat zu strukturieren. Hierbei sind vor allem die in der grundlegenden Veranstaltung dargestellten Modellierungskonzepte hilfreich. Sie stellen wesentliche Abstraktionskonzepte bereit, um den komplexen Entwicklungsvorgang in systematisch zu durchlaufende Schritte zu gliedern. Die Ergänzungsveranstaltungen bieten die Möglichkeit, das Erlernte in verschiedene Richtungen zu vertiefen: Etwa in Richtung der auf die Modellierung folgenden Implementationsschritte, oder auf die Anwendung der Konzepte und Techniken auf webbasierte Benutzungsschnittstellen oder das Vertiefen von Techniken des Usability Engineering. Die Ergänzungsveranstaltungen werden nach dem Fortschritt des Gebietes jeweils aktuell erweitert bzw. ergänzt.

Voraussetzungen und Vorkenntnisse

Für das gesamte Modul sind Programmierkenntnisse (zur Zeit in der Programmiersprache Java) hilfreich für das Verständnis, werden aber nicht explizit vorausgesetzt. Für die programmierorientierten Ergänzungsveranstaltungen (Programming Interactive Systems und

Programming for the Web) ist allerdings die erfolgreiche Teilnahme am Softwaretechnikpraktikum nachzuweisen. Die Veranstaltung „Praxis des Usability Engineering“ setzt Grundkenntnisse von Techniken des Usability Engineering voraus. Diese können etwa in der BSc-Veranstaltung „Usability Engineering“ erworben werden.

Lernziele der Veranstaltung

Studierende lernen...

Vermittlung von Faktenwissen

... über:

- Bedeutung verschiedener Modellierungstechniken im Entwicklungsprozeß
- Struktur eines modellbasierten Entwicklungsprozeß
- Aufgabenmodellierung
- Benutzerinteraktionsmodellierung
- Kontrollmodellierung
- Gestaltungsrichtlinien aus psychologischer Sicht
- wahrnehmungspsychologische Grundtatsachen und Gesetze
- Normen und Styleguides
- Konzepte und Techniken der Entwicklung webbasierter Benutzungsschnittstellen (Ergänzungsveranstaltung)
- Konzepte und Techniken der Programmierung von Benutzungsschnittstellen (Ergänzungsveranstaltung)

Vermittlung von methodischem Wissen

- den Prozess der Entwicklung von Benutzungsschnittstellen adäquat zu gliedern
- die Gestaltung verschiedener Aspekte von Benutzungsschnittstellen anhand geeigneter Modelle ohne eigentliche Programmieraktivitäten durchzuführen
- die Beurteilung und Erstellung von Benutzungsschnittstellen anhand gestalterischer Grundprinzipien
- Styleguides bei der Entwicklung von Benutzungsschnittstellen einzusetzen
- Benutzungsschnittstellen anhand von Normen zu überprüfen
- die Benutzbarkeit webbasierter Benutzungsschnittstellen einzuschätzen (Ergänzungsveranstaltung)
- Konzepte der klassischen Benutzungsschnittstellenentwicklung auf webbasierte Aufgabenstellungen anzuwenden (Ergänzungsveranstaltung)
- Konzepte, Techniken und Werkzeuge der Programmierung von Benutzungsschnittstellen anzuwenden (Ergänzungsveranstaltung)
- Usability-Tests in realistischem Umfang zu planen und durchzuführen (Ergänzungsveranstaltung)

Vermittlung von Transferkompetenz

Die im Modul als zentraler Inhalt vermittelten Modellierungskonzepte sind auch in anderen Bereichen der Informatik (jede Form von Verhaltensbeschreibungen) und weitgehend auch bei fortschreitender Entwicklung von Benutzungsschnittstellen einsetzbar, da etwa von Grafiktechniken oder Hardwareeigenschaften abstrahiert wird. Die in Ergänzungsveranstaltungen behandelte Erweiterung auf webbasierte Systeme schafft die Voraussetzung für das Erlernen jeweils neuer, aktueller Techniken, wird aber auch ständig an neue Entwicklungen angepaßt werden müssen. Da der menschliche Benutzer einem relativ geringen Wandel unterliegt, bleiben die Usability-

orientierten Veranstaltungen in hohem Maße aktuell und sind und bleiben für alle Problembereiche, die zwischen Mensch und Maschine existieren, relevant.

Vermittlung von normativ-bewertender Kompetenz

- ... die Einhaltung von Normen, Styleguides und gestalterischen Gesetzen in klassischen Benutzungsschnittstellen und bei Webseiten zu überprüfen
- ... die Werkzeuge zur Entwicklung von Benutzungsschnittstellen in ihrer grundsätzlichen Leistungsfähigkeit einzuschätzen und zu überprüfen
- ... die Schwierigkeit der Entwicklung einer Benutzungsschnittstelle in Relation zur gegebenen Entwicklungs- und Laufzeitumgebung einzuschätzen (Ergänzungsveranstaltung)
- ... die Machbarkeit interaktiver Systeme im Web einzuschätzen (Ergänzungsveranstaltung)
- ... die Qualität und Aussagekraft von Usability-Tests einzuschätzen (Ergänzungsveranstaltung)

Schlüsselqualifikationen

- Fähigkeit zur Nutzung moderner Benutzungsschnittstellentechnologien
- fachbezogenen Fremdsprachenkompetenzen durch englische Begleitliteratur
- Kooperations- und Teamfähigkeit durch Gruppenübungen

Modulzugehörigkeit:

Wahlmodul im Gebiet Mensch-Maschine-Wechselwirkung (MMWW)

Modus

- Leistungspunkte: 4 + 4 ECTS (2 Katalogveranstaltungen)
- SWS: 2+1, 2+1 (oder 2 im Falle des Seminars)
- Häufigkeit des Angebotes: 2-3 Katalogveranstaltungen pro Jahr im WS und SS

Methodische Umsetzung

Die theoretischen Konzepte werden in Präsenzübungen in Kleingruppen vertieft. Die Methoden werden in praktischen Übungen erprobt, je nach gewählter Ergänzungsveranstaltung an Entwicklungsaufgaben für Benutzungsschnittstellen, Webschnittstellen oder Usability-Tests.

Organisationsformen / Medieneinsatz / Literaturangaben

- Bei Veranstaltungen eine Doppelstunde Vorlesung pro Woche, und eine Stunde Präsenzübung jede Woche (ggf. eine Doppelstunde Präsenzübung jede zweite Woche), und/oder Lösen von Programmieraufgaben, bzw. Entwicklung von Modellierungsdokumenten oder Usability-Tests in eigener Zeit in ähnlichem Ausmaß
- Das Seminar ist typischerweise ein Blockseminar, ggf. aber auch semesterbegleitend, je nach Ankündigung
- Hinweise auf erwartete Aktivitäten der Studierenden: selbständiges Programmieren; Mitarbeit bei den Präsenzübungen; ggf. Umgang mit Versuchspersonen bei Benutzertests.
- Eingesetzte Materialien: Übungsblätter

Prüfungsmodalitäten

- Klausur

- Lösen praktischer Probleme durch Programmieraufgaben und/oder eigenständige Projekte
- Angaben zur Notenermittlung: Die Gewichtung von Klausur und Programmieraufgaben/-projekten wird zu Anfang des Semesters bekannt gegeben.

Modulverantwortliche(r)

Szwillus

III.4.6 Mensch-Maschine-Wechselwirkung

Rolle im Studiengang

Das Modul dient dazu, Studierenden, die sich für die Inhalte des Gebietes Mensch-Maschine-Wechselwirkung interessieren, einen tieferen und teilweise breiteren Zugang zu den Inhalten zu ermöglichen. Das Modul ist nicht inhaltlich zusammenhängend, vielmehr besteht hier eine breite Wahlmöglichkeit für die Studierenden, sich in die verschiedenen Richtungen je nach Wunsch tiefer oder parallel in mehreren Richtungen breiter einzuarbeiten.

Inhaltliche Gliederung des Moduls:

Der Modul setzt sich zusammen aus

- einer Reihe von Ergänzungsveranstaltungen, von denen zwei gewählt werden müssen:
 - Architekturen CSCW
 - Augmented Reality/Virtual Reality (geplant)
 - Computer und Behinderte - Zugang zur Informationstechnologie für Menschen mit speziellen Bedürfnissen
 - Computergenerierte Visualisierung
 - Digitale Bildverarbeitung
 - Gestaltung interaktiver Systeme
 - Konzepte digitaler Medien
 - Praxis des Usability Engineering
 - Programming Interactive Systems (in Engl.)
 - Programming Interactive Web Sites (geplant, in Engl.)
 - Software-Ergonomie
 - Web-Modellierung (geplant)

Die einzelnen Veranstaltungen gliedern sich wie folgt:

1. Architekturen CSCW
 - Von der Interaktion zur Kooperation: Überblick zu den Forschungsfeldern CSCW/CSCL/Groupware
 - Historische Entwicklung verschiedener CSCW-Architekturkonzepte
 - Architektonische Entwurfsmuster kooperationsunterstützender Systeme
 - Für den Bereich kooperationsunterstützender Systeme relevante Architekturkonzepte des WWW
 - Technologische Grundkonzepte und Metaphern kooperationsunterstützender Systeme
 - Frameworks kooperationsunterstützender Systeme
 - Mobilitätsaspekte kooperationsunterstützender Systeme
2. Augmented Reality/Virtual Reality (geplant)
3. Computer und Behinderte - Zugang zur Informationstechnologie für Menschen mit speziellen Bedürfnissen
4. Computergenerierte Visualisierung
 - Definition, Geschichte, Ziele
 - Die Ausgangsdaten
 - Der Betrachter und die Visualisierungsziele
 - "Mapping" von Daten auf Bilder
 - Visuelle Repräsentationen
 - Systeme und Werkzeuge
5. Digitale Bildverarbeitung (in engl.)
 - Visual Perception, Sampling and Quantization
 - Fourier Transformation

- Fast Fourier Transformation, Sampling, other Transformations
 - Point Processes
 - Neighborhood Processes
 - Filtering in Frequency Domain, Image Restoration
 - Image Compression
6. Gestaltung interaktiver Systeme
- Design Prinzipien
 - User-Centered Design
 - Xerox Star
 - Direkte Manipulation
 - Rooms System
 - Fisheye Views
 - See-Through Tools, Click-Through Tools, Magic Lenses
 - Information Visualizer, 3D Rooms
7. Konzepte digitaler Medien
- Instrument, Maschine, Automat, interaktives System
 - Hypertext
 - Groupware
 - Virtuelle Gemeinschaften
 - Virtuelle Realität, Erweiterte Realität
 - Ambient Computing, Mediatronik
8. Praxis des Usability Engineering
- Diese Veranstaltung ist projektorientiert – es werden über das Semester hinweg Entwurfsprojekte aus dem Bereich klassischer Benutzungsschnittstellen und/oder Webauftritte durchgeführt. Dabei werden die Studierenden vor allem als Entwickler von Benutzertests praktisch gefordert.
9. Programming Interactive Systems (in Engl.)
- Einleitung
 - Das Implementationsproblem
 - Grundlagen der Computergraphik
 - Ereignisorientierte Programmierung
 - Implementierung elementarer Interaktion
 - Benutzungsschnittstellen aus Widgets
 - Implementation von Websites
10. Programming Interactive Web Sites (geplant, in Engl.)
- Einleitung
 - Grundprinzipien interaktiver Websites
 - Client-seitige Technologie
 - Server-seitige Technologie
11. Software-Ergonomie
- Arbeitsschutzgesetze und Verordnungen
 - Internationale Normen und Standards der Software-Ergonomie
 - Theoretische Grundlagen der Gestaltung
 - a) Wahrnehmung
 - b) Gedächtnis
 - c) Ikonizität und Textualität
 - Leitprinzip „Reduzierung erzwungener Sequenzialität“
 - Präsentationskriterien
 - Interaktionskriterien
 - Einbettungskriterien (Konventionen)
 - Spezifische Aspekte des Web-Design

12. Web-Modellierung (geplant)

- Grundprinzipien modellbasierter Ansätze
- WebML
- OOHDM
- Aufgabenmodellbasierte Webmodellierung
- Implementationsfragen

Voraussetzungen und Vorkenntnisse

Für die programmierorientierten Veranstaltungen (Programming Interactive Systems und Programming for the Web) ist die erfolgreiche Teilnahme am Softwaretechnikpraktikum nachzuweisen. Die Veranstaltung „Praxis des Usability Engineering“ setzt die Veranstaltung „Usability Engineering“ oder alternativ die „Modellierung von Benutzungsschnittstellen“ voraus. Letztere ist Voraussetzung für die Veranstaltung „Web Modellierung“. Die Veranstaltungen im Computergrafik-Bereich setzen die Vorlesung „Computergrafik I“ voraus.

Lernziele der Veranstaltung

In den Veranstaltungen dieses Moduls können die Studierenden sich in den verschiedenen Teilbereichen der Mensch-Maschine-Wechselwirkung vertiefen. Bezogen auf die Computergrafik werden aufbauend auf den vorher vermittelten Grundkonzepten die Vertiefungen Visualisierungen von Daten und Bildverarbeitung angeboten. Im Bereich der Entwicklung von Benutzungsschnittstellen werden die bereits vermittelten Modellkonzepte vertieft und um Implementationskonzepte und Techniken aus dem Bereich der Webprogrammierung ergänzt. Im Usability Engineering werden praktische Fähigkeiten im Entwickeln von Benutzertests, einschließlich dem Umgang mit Versuchspersonen und dem praktischen Versuchsaufbau vermittelt. Die Anwendung der Techniken für Web Usability ist eine alternative Vertiefungsmöglichkeit. Im Gestaltungsbereich sollen die Studierenden an relevanten Beispielen und in eigenständig erarbeiteten Entwicklungsprojekten die Möglichkeiten kennenlernen und erproben, eine benutzer- und aufgabengerechte Gestaltung von Bedienoberflächen, sei es von klassischer Software oder von Websites, auch orientiert an allgemeinen Gestaltungsregeln, Normen und Vorschriften, durchzuführen. Die notwendigen Kenntnisse und Fähigkeiten werden u.a. in der „Gestaltung interaktiver Systeme“ und in der „Softwareergonomie“ vermittelt.

Schlüsselqualifikationen

- Fähigkeit zur Nutzung moderner Technologien in der Mensch-Computer-Interaktion
- fachbezogenen Fremdsprachenkompetenzen durch englische Begleitliteratur
- Kooperations- und Teamfähigkeit durch Gruppenübungen

Modulzugehörigkeit:

Wahlmodul im Gebiet Mensch-Maschine-Wechselwirkung (MMWW)

Modus

- Leistungspunkte: 4 + 4 ECTS (2 Katalogveranstaltungen)
- SWS: 2+1, 2+1 (oder 2 im Falle des Seminars)
- Häufigkeit des Angebotes: 2-3 Katalogveranstaltungen pro Jahr im WS und SS

Methodische Umsetzung

Die theoretischen Konzepte werden in Präsenzübungen in Kleingruppen vertieft. Die Methoden werden in praktischen Übungen erprobt, je nach gewählter Ergänzungsveranstaltung an Entwicklungsaufgaben mit entsprechenden Inhalten.

Organisationsformen / Medieneinsatz / Literaturangaben

- Bei Veranstaltungen eine Doppelstunde Vorlesung pro Woche, und eine Stunde Präsenzübung jede Woche (ggf. eine Doppelstunde Präsenzübung jede zweite Woche), und/oder Lösen von Programmieraufgaben, bzw. Entwicklung von Modellierungsdokumenten oder Usability-Tests in eigener Zeit in ähnlichem Ausmaß
- Das Seminar ist typischerweise ein Blockseminar, ggf. aber auch semesterbegleitend, je nach Ankündigung
- Hinweise auf erwartete Aktivitäten der Studierenden: selbständiges Programmieren; Mitarbeit bei den Präsenzübungen; ggf. Umgang mit Versuchspersonen bei Benutzertests.
- Eingesetzte Materialien: Übungsblätter

Prüfungsmodalitäten

- Klausur
- Lösen praktischer Probleme durch Programmieraufgaben und/oder eigenständige Projekte
- Angaben zur Notenermittlung: Die Gewichtung von Klausur und Programmieraufgaben/-projekten wird zu Anfang des Semesters bekannt gegeben.

Modulverantwortliche(r)

Keil

III.4.7 Mensch-Computer-Interaktion

Rolle im Studiengang

Die Produktivität von Informatiksystemen hängt maßgeblich auch von ihrer Gebrauchstauglichkeit ab. Doch auch gesundheitliche Beeinträchtigungen durch den Einsatz von Software sind mittlerweile bekannt und international als Gefährdung anerkannt. Aufgrund einschlägiger Gesetze stellt der Gesundheitsschutz bei der Bildschirmarbeit entsprechende Anforderungen an die Gestaltung von Benutzungsoberflächen. Produktivität und Gesundheitsschutz sind somit zwei wichtige sich ergänzende Faktoren, die auch in der Ausbildung von Informatikern berücksichtigt werden müssen. Hinzu kommt die mittlerweile durch internationale Standards und Normen definierte Barrierefreiheit, die dafür sorgen soll, dass digitale Informationsangebote allen Bürgern gleichermaßen zugänglich sind.

Entscheidend sind dabei konstruktive Gestaltungsansätze, Entwicklungswerkzeuge und Entwicklungsmethoden auf der Grundlage der kognitiven Psychologie mit analytischen Konzepten zu verbinden. Informatiker sollen in die Lage versetzt werden, die wichtigsten Anforderungen zu erkennen und in entsprechende technische Lösungen umzusetzen. Neben der Kenntnis einschlägiger Gesetze und Normen zählen dazu theoretische, gestalterische und methodische Grundlagen und entsprechende Spezifikationstechniken. Das Modul gehört zu den Modulen im Teilbereich Mensch-Maschine-Wechselwirkung (MMWW)

Inhaltliche Gliederung des Moduls:

Der Modul umfasst Veranstaltungen, die sich mit den verschiedenen Facetten der Analyse, Herstellung und Gestaltung interaktiver Systeme und kooperativer Medien befassen. Rechtliche Rahmenbedingungen und Verordnungen werden dabei ebenso berücksichtigt wie kognitionspsychologische Grundlagen. Dementsprechend besteht der Modul aus folgenden Veranstaltungen: :

- Einführung in die Mensch Computer Interaktion
- Software-Ergonomie
- Gestaltung interaktiver Systeme
- Praxis der Usability

Inhaltliche Verwendbarkeit

Die in diesem Modul erworbenen Kenntnisse vermitteln eine breite Grundlage für die Entwicklung interaktiver Systeme. Solche Kenntnisse sind heute unverzichtbar, da die effektive und verlässliche Interaktion mit dem Rechner eine entscheidende Komponente in der Nutzung moderner Datenverarbeitungssysteme darstellt. Zugleich gibt es in diesem Bereich vielfältige Anknüpfungsbereiche zu speziellen Anwendungsfeldern wie e-Learning, Web-basiertes Arbeiten, mobile Geräte, Visualisierung, Virtuelle Realität, Informationsdesign, etc. und zu anderen wissenschaftlichen Disziplinen wie z.B. der Psychologie, der Pädagogik und den Medienwissenschaften.

Voraussetzungen und Vorkenntnisse

Es werden die wesentlichen Konzepte aus den ersten vier Semestern der Informatik vorausgesetzt. Insbesondere gehören dazu Modellierung und entsprechende Spezifikationssprachen sowie Softwarewerkzeuge zur Entwicklung interaktiver Systeme.

Kenntnisse über grafische Datenverarbeitung und Wissenschaftliche Visualisierung stellen eine hilfreiche Ergänzung dar, sind aber zum Verständnis nicht unbedingt erforderlich.

Lernziele der Veranstaltung

Die Veranstaltung soll die Studierenden in die Lage versetzen, interaktive Systeme auf allen Niveaus und in allen Formen analysieren, bewerten und konstruktiv verbessern zu können. Außerdem sollen Sie mit den gesetzlich formulierten Anforderungen und den einschlägigen internationalen Standards und Normen vertraut sein. In einzelnen Bereichen geht es zudem darum Speziallösungen und Mustergültige Ansätze für die Umsetzung hochwertiger Benutzungsoberflächen kennen zu lernen und dabei auch Innovationspotenziale abschätzen zu können.

Modulzugehörigkeit:

Wahlmodul im Gebiet Mensch Maschine Wechselwirkung.

Modus

Leistungspunkte : 4+4

SWS 2+1, 2+1

Häufigkeit des Angebotes: Eine Basisveranstaltung jedes WS, die weiterführenden Veranstaltungen jedes SS.

Prüfungsmodalitäten

Je nach Teilnehmerzahl mündliche Prüfung oder Klausur

Modulverantwortliche(r)

Keil