

**INFO
zu den
Projektgruppen
der Fakultät für Informatik
mit
Beginn im SoSe 2011**

Stephan Windmüller
Projektgruppen-Beauftragter
OH 14 / Raum 129
Tel.: (0231) 755-7731
E-Mail: pgadmin@ls4.cs.uni-dortmund.de

Postadresse:
Technische Universität Dortmund
Lehrstuhl Informatik V
44221 Dortmund

Inhalt

- Vorbemerkungen
- Projektgruppen SoSe 2011 und Termine der Einzelpräsentationen
- Zum Auswahlverfahren
- Anmeldung zu den Projektgruppen
- Zur Stellung der Projektgruppen in der Diplomprüfungsordnung
- Übersicht zu den Voraussetzungen der einzelnen Projektgruppen
- Kurzbeschreibung der Projektgruppen

Vorbemerkungen

Projektgruppen gibt es an den Hochschulen der Bundesrepublik seit ca. 1970. Sie wurden im wesentlichen aufgrund der Forderungen von Studenten der 68-er und Nach-68-er Generation in Studienpläne und Prüfungsordnungen aufgenommen. Zentrale Forderungen waren:

- selbstbestimmtes Lernen
- interdisziplinäre Ausbildung
- berufsbezogene Studiengänge

wobei gesellschaftsbezogene Aspekte des Studiums besonders betont werden sollten. Wenn in der Fakultät für Informatik auch lediglich der „Berufspraxisbezug“ der Ausbildung die wesentliche Begründung für die Einführung der Projektgruppen bildete, erscheinen vor obigem Hintergrund die folgenden Überlegungen wichtig. Lernende und Lehrende kommen bei der Projektarbeit in eine völlig neue, ungewohnte und sicherlich teilweise schwierig zu akzeptierende Situation. Für die Studenten heißt das zunächst: weg von der Konsumentenhaltung aus Vorlesungen, weg vom Bearbeiten vorgegebener Aufgaben aus Übungen, Praktika und Seminaren – stattdessen *selbständiges Strukturieren der Arbeit, selbständiges Erarbeiten von Problemen und eigenverantwortliches Lösen der Probleme*. Für die Lehrenden heißt das: weg von der Rolle als nie versagende Wissensquelle, die auch in der schwierigsten Situation immer noch eine richtige Lösung hervorbringt – stattdessen auch *Zugestehen von fachlicher Autorität an Studenten*. Und das wiederum heißt zu akzeptieren, dass die Lehrenden weder für alle fachlichen Probleme fertige Lösungen, noch bei jeder persönlichen Auseinandersetzung ideale Ratschläge zur Verhinderung von Streitigkeiten haben.

Zusammengefasst heißt das:

Projektarbeit lebt vom Engagement der Projektgruppenmitglieder (Studenten und Veranstalter) und stirbt bei einem Mangel davon. Noch stärker als bei anderen Lehrveranstaltungsformen liegt es in der Verantwortung jedes Teilnehmers (auch hier Studenten und Veranstalter), was er für sich und die Gruppe daraus macht.

Projektgruppen SoSe 2011 und Termine der Einzelpräsentationen

Nr.	Thema der Projektgruppe	Veranstalter	Einzelpräsentation
A	Scalable Virtual Data Infrastructure	Yahyapour, Röblitz, Feldhaus, Wieder, Chronz	25.11.2010 10 Uhr (GB V, Raum 223)
B	Mobile Musik-Welten – Spielerische Exploration von digitalen Musiksammlungen	Vatolkin, Preuß	19.11.2010 10–11 Uhr (OH14, Raum 202)
C	Internetbanking – Konzeption und Entwicklung eines sicheren Cloud-Basierten Internetbanking-Systems mit anschließender Sicherheitsanalyse auf Basis von Business Process Mining	Jürjens, Faßbender, Schmidt	25.11.2010 13–14 Uhr (GB V, Raum 420)
D	NetSensLog – Netzwerkbasierte Sensorfusion für einen Fahrzeugschwarm in der Intralogistik	Weichert, Timm, Kamagaew	24.11.2010 12–14 Uhr (OH14, Raum E04)
E	FiLeCC – Field-Level Computation Cloud – Verlagerung von übergeordneten Steuerungsfunktionen in die Feldebene von Materialflusssystemen mithilfe eines Cloud-basierten Ansatzes	Feldhorst, Mosblech	26.11.2010 10:30–12 Uhr (EF73)

Zum Auswahlverfahren

Bei der Anmeldung ist anzugeben, mit welcher Priorität Sie welche Projektgruppe wählen:

1=höchste Priorität

5=niedrigste Priorität

0=an dieser PG bin ich nicht interessiert

Es ist nicht erlaubt, die gleiche Priorität mehrfach zu vergeben. Jeder Veranstalter erhält die Anmeldungen, bei denen seine Projektgruppe mit erster Priorität angegeben ist und wählt sich seine Teilnehmer nach fachlichen Gesichtspunkten aus. Nicht besetzte Plätze in einer Projektgruppe werden durch überzählige Studenten aus anderen Projektgruppen entsprechend den Prioritäten besetzt. Jede Projektgruppe muss mindestens 8 und darf höchstens 12 Teilnehmer haben.

Die PG-Teilnehmer werden nach folgendem Verfahren ausgewählt und auf die PGs verteilt:

1. Die PG-Bewerber werden gemäß ihrer 1. Priorität den Veranstaltern zugeordnet.
2. Die PG-Veranstalter wählen davon insgesamt maximal 9 Teilnehmer nach fachlichen Gesichtspunkten aus.
3. Weitere Teilnehmer werden aus den Bewerbern mit 1. Priorität abhängig vom Zeitpunkt ihres Vordiploms (je weiter in der Vergangenheit, um so größere Priorität; für Bewerber, die bereits an einer PG erfolglos teilgenommen haben, gilt der Zeitpunkt der letzten Anmeldung) bestimmt; bei gleichem Zeitpunkt entscheidet das Los.
4. Ist die Gesamtzahl 12 in einzelnen PGs nicht erreicht, wird das Verfahren ab (2.) gemäß 2. Priorität, in einer weiteren Runde gemäß 3. Priorität etc. durchgeführt.
5. Nachrücklisten (max. 3 Nachrücker pro PG) werden erst nach Abschluss des Verteilverfahrens vom Veranstalter erstellt.

Anmeldung zu den Projektgruppen

Vordiplom ist Voraussetzung (Stichtag: 01.12.2010)

Die Anmeldung erfolgt bis

01.12.2010 (12.00 Uhr)

online unter

<https://projektgruppen.cs.uni-dortmund.de>

Weitere Informationen zu Projektgruppen können auch unter

<http://www4.cs.uni-dortmund.de/PGB/> abgerufen werden.

Zur Stellung der Projektgruppen in der Diplomprüfungsordnung

Die Teilnahme an einer Projektgruppe ist Pflicht für alle diejenigen Studenten, die im Hauptfach Informatik oder Ingenieur-Informatik studieren. Für die erfolgreiche Teilnahme gibt es einen Schein, der dem Antrag auf Zulassung zur letzten Diplomprüfung beigelegt werden muss. Kriterien zur Scheinvergabe sind in der Projektgruppenordnung näher erläutert. Die Projektgruppenordnung kann beim Projektgruppenbeauftragten eingesehen werden. Sie ist auch auf dem WWW-Server des LS4 unter

<http://www4.cs.uni-dortmund.de/PGB/>

abrufbar.

Hinweis zum Arbeitsaufwand in Projektgruppen:

Immer wieder gibt es Diskussionen über den Arbeitsaufwand, der für eine Projektgruppe aufgebracht werden muss. Für die Studierenden ist es wichtig zu wissen, dass die im Vorlesungsverzeichnis genannten 8 SWS pro Semester NICHTS bzw. nur wenig mit dem tatsächlichen Arbeitsaufwand zu tun haben, der für die PG-Arbeit notwendig ist. In gleicher Weise, wie z. B. bei einer Lehrveranstaltung mit 6 SWS (4 SWS Vorlesung, 2 SWS Übung) erhebliche Vor- und Nacharbeit bis zum Bestehen der Prüfung notwendig ist, ist auch bei PGs deutlich mehr Zeit aufzubringen als etwa 8 Stunden pro Woche. Man rechnet für durchschnittlich begabte Studierende mit dem Faktor 2,5 über die gesamten 2 Semester, auch hier wie bei Prüfungen inkl. der vorlesungsfreien Zeit. Durchschnittlich begabte Studierende sollten mit einer tatsächlichen durchschnittlichen Arbeitsbelastung von etwa 15-20 Stunden pro Woche rechnen, weniger Begabte wie bei Prüfungen mit teilweise deutlich mehr, sehr Begabte mit deutlich weniger. Die Teilnahme an weiteren Lehrveranstaltungen ist entsprechend zu planen. Der Aufwand für die PG sollte etwa der Hälfte eines Vollzeitstudiums entsprechen.

Ansprechpartner:

Sollte es im Verlauf der Projektgruppenarbeit zu Schwierigkeiten, z. B. wegen zu hoher Anforderungen, zu großem Arbeitsaufwand oder ähnlichem kommen, steht der Projektgruppenbeauftragte der Fakultät für Informatik als Ansprechpartner zur Verfügung.

Übersicht zu den Voraussetzungen der einzelnen Projektgruppen

(V) Voraussetzung

(W) wünschenswert

(M) mindestens eine

(2) mindestens zwei

A: Scalable Virtual Data Infrastructure (Yahyapour, Röblitz, Feldhaus, Wieder, Chronz)

- Kenntnisse in mindestens einer Programmiersprache (V)
- Rechensysteme (M)
- Effiziente Algorithmen und Komplexitätstheorie (M)
- Service Computing (M)
- Webtechnologien 1 (M)
- Komponenten- und Service-Orientierte Softwareentwicklung (M)
- Verteilte Algorithmen 1 (M)
- Verteilte Algorithmen 2 (M)
- Rechnernetz Anwendungen (M)
- Begeisterung für wissenschaftliche Anwendungen wie z.B. aus der Astronomie, Hochenergiephysik, Klimaforschung oder Geoforschung (W)

B: Mobile Musik-Welten – Spielerische Exploration von digitalen Musiksammlungen (Vatolkin, Preuß)

- Programmierung in Java (V)
- Gute Deutsch- und Englischkenntnisse (V)
- Digitale Bildverarbeitung (M)
- Datenvisualisierung (M)
- Mensch-Maschine-Interaktion (M)
- Eingebettete Systeme (M)
- Einführung in die Computational Intelligence (M)
- Softwaretechnologie, -entwicklung (W)
- Programmierung mobiler Endgeräte, insbesondere graphischer Oberflächen (W)
- Grundlegende Kenntnisse der Statistik (W)
- \LaTeX -Kenntnisse zur Dokumentation (W)
- Kenntnis des Betriebssystems Unix (W)

C: Internetbanking – Konzeption und Entwicklung eines sicheren Cloud-Basierten Internetbanking-Systems mit anschließender Sicherheitsanalyse auf Basis von Business Process Mining (Jürjens, Faßbender, Schmidt)

- Softwaretechnik (V)
- Kenntnisse in objektorientierter Programmierung (V)
- Mensch-Maschine-Interaktion (M)
- Modellgestützte Analyse und Optimierung (M)
- Formale Methoden des Systementwurfs (M)
- Kenntnisse in UML (W)
- Kenntnisse in UMLsec (W)
- Kenntnisse in Cloud-Entwicklung (W)
- Kenntnisse in Sicherheitstesten von Software (W)

D: NetSensLog – Netzwerkbasierter Sensorfusion für einen Fahrzeugschwarm in der Intralogistik (Weichert, Timm, Kamagaew)

- Mensch-Maschine-Interaktion (Graphische Systeme), Digitale Bildverarbeitung oder Graphische Datenverarbeitung (V)
- Eingebettete Systeme, Rechnergestützter Entwurf von Mikroelektronik oder Rechnerarchitektur (V)
- Kenntnisse in einer Programmiersprache, z.B. Java, C++ oder C (V)

E: FiLeCC – Field-Level Computation Cloud – Verlagerung von übergeordneten Steuerungsfunktionen in die Feldebene von Materialflusssystemen mithilfe eines Cloud-basierten Ansatzes (Feldhorst, Mosblech)

- Kenntnisse in OOP mit Java (V)
- Methodische Grundlagen des Software Engineering (M)
- Modellierung und Analyse eingebetteter und verteilter Systeme (M)
- Rechnernetzwerken (M)
- Webtechnologien (W)
- Service Computing (M)

Projektgruppenantrag

- 1 **Thema** Scalable Virtual Data Infrastructure
- 2 **Zeitraum** SoSe 2011 und WiSe 2011/12
- 3 **Umfang** 8 SWS pro Semester
- 4 **Veranstalter** ITMC / Lehrstuhl Service Computing, Campus Süd, Gebäude V
 Prof. Dr.-Ing. Ramin Yahyapour, Raum 101, ☎ 2346, ramin.yahyapour@udo.edu
 Dr. Thomas Röblitz, Raum 340, ☎ 5329, thomas.roeblitz@udo.edu
 Florian Feldhaus, Raum 347, ☎ 5324, florian.feldhaus@udo.edu
 Philipp Wieder, Raum 123, ☎ 2767, philipp.wieder@udo.edu
 Peter Chronz, Raum 347, ☎ 5588, peter.chronz@udo.edu

5 Aufgabe

Motivation

Viele grundlegende Fragestellungen erfordern die Erforschung komplexer Zusammenhänge in verschiedenen Disziplinen, wie z.B. der Klimaforschung, Astronomie, Hochenergiephysik oder Bioinformatik. Dabei stehen die Wissenschaftler vor enormen technischen Herausforderungen, die sich aus der Vielfalt der anzuwendenden Methoden, deren benötigte Berechnungskapazität, den zu analysierenden Datenvolumina [1], deren verteilte Beobachtung und Speicherung sowie der Größe internationaler, zunehmend auch interdisziplinärer, Kooperationen ergeben [2,3].

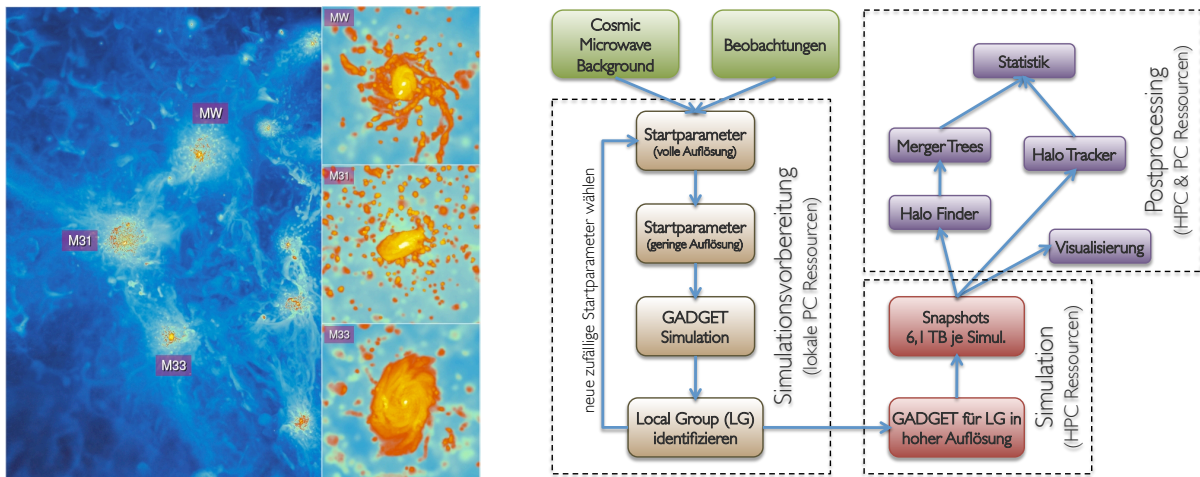


Abbildung 1: Simulation der Milchstraße. Links: Gasverteilung der *Local Group* (K. Riebe, Astrophysikalisches Institut Potsdam). Rechts: Wesentliche Arbeitsschritte für den Ablauf der Simulation.

Der linke Teil der Abbildung 1 zeigt einen Ausschnitt der Bildung von Galaxien, der durch Simulationen des CLUES-Projektes [4] berechnet wurde. Der rechte Teil illustriert die wichtigsten Arbeitsschritte, die auf weltweit verteilten Ressourcen ausgeführt werden. Die Verwaltung der Ein- und Ausgabedateien für jeden Arbeitsschritt erfordert **manuelle** Kopieraktionen durch die Wissenschaftler. Zunächst werden die Ausgangsdaten auf lokalen Ressourcen (PCs, Cluster) bearbeitet, um geeignete Startparameter für die Simulationen zu ermitteln (Simulationsvorbereitung). Im Anschluss werden die eigentlichen Simulationen auf Hochleistungsrechnern¹ durchgeführt. Für einen Simulationslauf werden dabei 6,1 Terabyte an Daten erzeugt, die in Dateispeichersystemen in den Rechenzentren abgelegt werden. Diese Daten können anschließend von verschiedenen Wissenschaftlern ausgewertet werden, um spezielle Fragen zu beantworten (Postprocessing). In

¹Im CLUES-Projekt werden insbesondere die Rechenzentren NIC (Jülich), LRZ (München) und BSC (Barcelona) genutzt.

der Auswertungsphase werden in der Regel nicht alle Informationen einer Datei sondern nur Ausschnitte [5] benötigt.

Da die Daten den Kern vieler Forschungsprojekte bilden, wurden in den letzten zehn Jahren grundlegende Technologien (Grid Computing) und Speziallösungen für die **verteilte Verwaltung und Analyse von Daten** entwickelt. Der Einsatz dieser Speziallösungen erfordert oft einen erheblichen Administrationsaufwand, einen hohen Einarbeitungsaufwand, die Adaptierung existierender Anwendungen und ist nur umständlich an neue Technologien (wie z.B. Cloud Computing) anpassbar.

Das Ziel der Projektgruppe ist, Wissenschaftlern eine leicht benutzbare, adaptierbare und skalierbare virtuelle Dateninfrastruktur zur Verfügung zu stellen.

Eine **Scalable Virtual Data Infrastructure** aggregiert existierende Daten-Ressourcen (z.B. Dateiserver, \$HOME-Verzeichnisse, Online-Datenspeicher) und stellt den Nutzern an allen angeschlossenen Rechnern (z.B. Cluster, Grid, PC, Server) eine einheitliche Sicht zur Verfügung – z.B. in Form einer Webschnittstelle, einer API oder eines Dateisystems. Der Dienst Dropbox [6] realisiert eine verteilte Dateninfrastruktur durch die Synchronisation von Dateikopien zwischen beteiligten Rechnern. Da wissenschaftliche Datenvolumina mehrere Terabyte bis Petabyte betragen können, kann der Dropbox-Ansatz in der existierenden Form nicht angewendet werden (begrenzte Netzwerktransfervolumina und Speicherkapazität).

Um einen ersten Schritt in Richtung einer **Scalable Virtual Data Infrastructure** zu gehen, soll die Projektgruppe vorhandene Basistechnologien im Bereich des Datenmanagement auf Anwendbarkeit prüfen (z.B. XtremFS [7], Amazon S3 [8] und S3QL [9]), für ausgewählte Teilbereiche Lösungen vorschlagen und prototypisch für die IT-Infrastruktur der TU Dortmund – bestehend aus Rechenressourcen und Datenspeichern – umsetzen. Abbildung 2 illustriert eine Umgebung mit verschiedenen Ressourcen und deren Nutzung in virtuellen Dateninfrastrukturen (VDI A-C). Dabei können virtuelle Dateninfrastrukturen auch aufeinander aufbauen (VDI A&C).

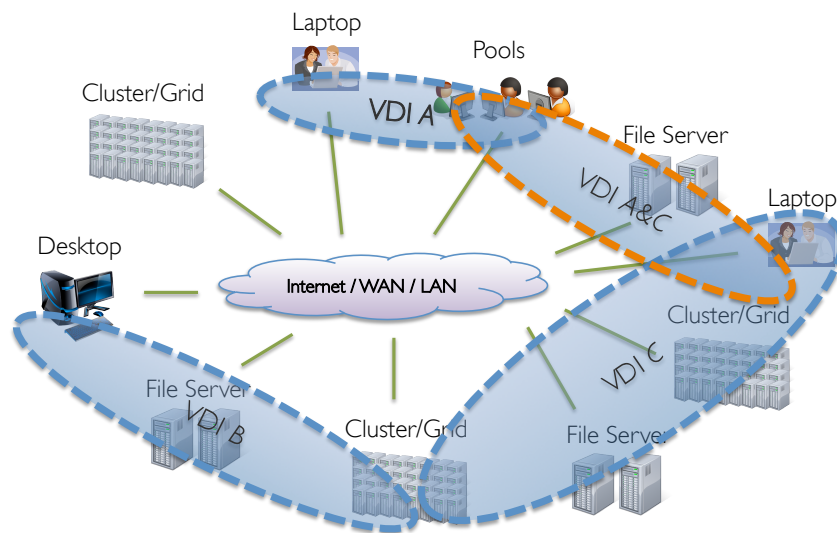


Abbildung 2: Verteilte Umgebung mit mehreren Rechen- und Speicherressourcen, die durch die Bildung teilweise überlappender virtueller Dateninfrastrukturen eine leicht zu benutzende Arbeitsumgebung bereitstellt.

Als konkretes Anwendungsbeispiel soll der Workflow aus dem CLUES-Projekt untersucht werden (vgl. Abb. 1). Dieser ist relativ einfach und stellt dennoch typische Anforderungen an die Dateiverwaltung. Durch weitere Kooperationen (siehe Rahmenbedingungen weiter unten) können leicht zusätzliche Anwendungsszenarien zur Verfügung gestellt werden. Die IT-Infrastruktur der TU Dortmund beinhaltet alle Arten von

Speicher- und Rechenressourcen, die in wissenschaftlichen Anwendungen benötigt werden. Daher sollen vorzugsweise diese für die prototypische Umsetzung genutzt werden. Die entwickelten Verfahren sollen aber so angelegt sein, dass sie auch für andere Ressourcen mit ähnlichen Eigenschaften verwendet werden können.

Aufgabenstellung

Die Entwicklung eines effizienten und zuverlässigen Dienstes in Form einer *Scalable Virtual Data Infrastructure* erfordert eine Kombination von wissenschaftlichen und softwaretechnischen Voraussetzungen. Dabei liegt Fokus der Projektgruppe auf den folgenden drei Arbeitsfeldern für die jeweils ein möglicher Lösungsansatz vorgestellt wird:

Basistechnologien für das transparente Management von Dateien Schnittstellen und Zugriffsprotokolle können durch einen Virtual Storage Layer (VSL) vereinheitlicht werden. Auf Basis des VSL können verschiedene Speicherorte in das Dateisystem der Benutzerin eingeblendet werden [10], um einen transparenten und leichten Zugriff auf verteilte Dateien zu ermöglichen. Werden Dateien geändert, so müssen diese Änderungen mit Hilfe eines geeigneten Protokolls an allen angeschlossenen Zugriffsorten sichtbar werden.

Verwaltung aller relevanten Metadaten Dateien umfassen den Inhalt und Metadaten (z.B. Besitzer, Zugriffsrechte, Datum der letzten Änderung, usw.). Für den Aufbau einer virtuellen Dateninfrastruktur ist die effiziente, verteilte und zuverlässige Verwaltung dieser Metadaten eine zentrale Komponente [11,12]. Neben den bekannten Kernmetadaten (Besitzer, Zugriffsrechte, usw.) müssen anwendungsspezifische Metadaten, die den Inhalt einer Datei beschreiben, verwaltet werden, um effizient und komfortabel nach bestimmten Informationen suchen zu können.

Persönliches Management Portal Die Verwaltung der persönlichen *Scalable Virtual Data Infrastructure* soll möglichst leicht von beliebigen Rechnern aufrufbar sein. Dazu bietet sich ein Webportal an [13], über das verschiedene Datenspeicher eingebunden werden können. Darüberhinaus können Arbeitsgruppen und deren Zugriffsrechte konfiguriert werden. Ein Webportal ermöglicht auch einen schnellen Überblick über die Auslastung der Ressourcen und kann für Mitteilung von wichtigen Ereignissen – Wartung, Ausfall von existierenden Ressourcen oder Verfügbarkeit von neuen – informieren.

Die grundlegenden Mechanismen für die Verwaltung von Dateien und Metadaten sollen dabei möglichst generisch gestaltet sein, so dass sie nicht nur im konkreten Umfeld der TU Dortmund sondern auch in anderen bzw. in erweiterten Anwendungsszenarien eingesetzt werden können.

Ein wesentliches Merkmal dieser Projektgruppe ist die Möglichkeit, die entwickelten Lösungen im realen Einsatz auf den Systemen der TU Dortmund zu erproben. Darüberhinaus bietet die PG einen Einblick in aktuelle Herausforderungen in der wissenschaftlichen Datenverwaltung.

Rahmenbedingungen

Am ITMC betreiben wir derzeit zwei Rechencluster (Grid und LiDOng), speichern Daten in einem parallelen Dateisystem (Lustre) und bereiten ein NetApp-Storage-System für den produktiven Einsatz vor. Da die Ergebnisse der Projektgruppe so weit wie möglich als Basis für produktive Lösungen im Lehr- und Forschungsbetrieb an der TU Dortmund eingesetzt werden sollten, soll die prototypische Implementierung auf dieser Basisinfrastruktur aufsetzen.

Neben der Kooperation mit dem Astrophysikalischen Institut Potsdam, bestehen Kooperationen mit dem GeoForschungsZentrum Potsdam, dem Alfred-Wegener-Institut für Polar- und Meeresforschung Bremerhaven sowie dem Lehrstuhl Experimentelle Physik Vb, TU Dortmund. Gegenstand aller dieser Kooperationen ist die Neu- und Weiterentwicklung von Infrastrukturen für nutzerfreundliche, effiziente und skalierbare Verwaltung von verteilten Dateien. Daher können bei Bedarf leicht zusätzliche Anwendungsszenarien zur Verfügung gestellt werden.

6 Teilnahmevoraussetzungen

Voraussetzung Kenntnisse in mindestens einer Programmiersprache

Mindestens eine Vorlesung Rechensysteme, Effiziente Algorithmen und Komplexitätstheorie, Service Computing, Webtechnologien 1, Komponenten- und Service-Orientierte Softwareentwicklung, Verteilte Algorithmen 1, Verteilte Algorithmen 2, Rechnernetz Anwendungen

Wünschenswert Begeisterung für wissenschaftliche Anwendungen wie z.B. aus der Astronomie, Hochenergiephysik, Klimaforschung oder Geoforschung

7 Minimalziel

Das Minimalziel ist erreicht, wenn gemäß Aufgabenstellung folgende Punkte realisiert wurden:

- Definition des Virtual Storage Layer inkl. Schnittstellen für Client, Metadaten dienst und Datenspeicher
- prototypische Implementierung des Virtual Storage Layer
- Konzeption und Implementierung eines Metadaten dienstes
- transparente Einbindung des VSL in ein Betriebssystem
- Implementierung eines web-basierten Portals für die Verwaltung einer *Scalable Virtual Data Infrastructure*

8 Referenzen

- [1] IDC. *The Diverse and Exploding Digital Universe*. White paper, März 2008, www.emc.com/collateral/analyst-reports/diverse-exploding-digital-universe.pdf.
- [2] T. Hey, S. Tansley, K. Tolle. *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Microsoft Research, 2009.
- [3] H. B. Newman et al. Data-intensive e-science frontier research. *Commun. ACM*, 46(11):68–77, 2003.
- [4] CLUES – Constrained Local UniversE Simulations Projekt. www.clues-project.org
- [5] T. Schütt, A. Merzky, A. Hutanu, F. Schintke. Remote Partial File Access Using Compact Pattern Descriptions. In *IEEE/ACM Intl. Symp. on Cluster Computing and the Grid - CCGrid2004*, pp. 1-8, April 2004.
- [6] Dropbox. www.dropbox.com
- [7] F. Hupfeld et al. The XtremFS architecture – a case for object-based file systems in Grids. *Concurrency and Computation: Practice and Experience*, 20(17), 2008.
- [8] Amazon Simple Storage Service (S3). aws.amazon.com/de/s3
- [9] S3QL – A file system for online storage. code.google.com/p/s3ql
- [10] FUSE – Filesystem in Userspace. fuse.sourceforge.net
- [11] G. Lausen. *Datenbanken: Grundlagen und XML-Technologien*. 1. Auflage, 2005
- [12] RDF Primer. www.w3.org/TR/rdf-primer
- [13] T. Alby. *Web 2.0. Konzepte, Anwendungen, Technologien*. Hanser Verlag, 2007.

9 Rechtlicher Hinweis

Die Ergebnisse der Projektarbeit und die dabei erstellte Software sollen der Fakultät für Informatik uneingeschränkt für Lehr- und Forschungszwecke zur freien Verfügung stehen. Darüber hinaus sind keine Einschränkungen der Verwertungsrechte an den Ergebnissen der Projektgruppe und keine Vertraulichkeitsvereinbarungen vorgesehen.

Projektgruppenantrag

1. PG-Thema *Mobile Musik-Welten:
Spielerische Exploration von digitalen Musiksammlungen*
2. PG-Zeitraum SoSe 11 und WiSe 11/12
3. PG-Umfang 8 SWS pro Semester
4. PG-Veranstalter Lehrstuhl für Algorithm Engineering (LS11), Computational Intelligence Group

Igor Vatolkin, OH 14, Raum 234, Tel. 7705, igor.vatolkin@tu-dortmund.de

Mike Preuß, OH 14, Raum 240, Tel. 7708, mike.preuss@tu-dortmund.de

beratend: **Dr. Wolfgang Theimer**, Research in Motion (RIM), Bochum

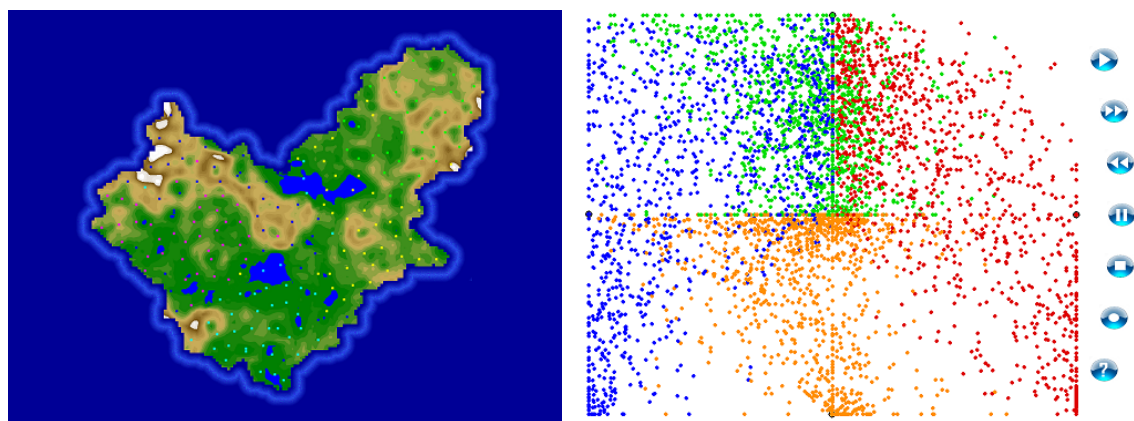
5. PG-Aufgabe

Motivation und Themengebiet

Leistungsfähigkeit und Funktionsumfang mobiler Endgeräte sowie die Verfügbarkeit des Internets haben in den letzten Jahren erheblich zugenommen. Ebenfalls hat sich der Musikkonsum stark verändert: Musik ist zumeist digital verfügbar, die persönlichen Sammlungen mit Audiodateien umfassen nicht selten hunderte von Alben. Das benötigt intelligente Verwaltungsmethoden und ermöglicht ganz andere spielerische Navigations- und Empfehlungskonzepte.

Zwei wesentliche Problemstellungen müssen dabei gelöst werden: zum einen die Handhabung von sehr großen digitalen Musiksammlungen - z.B. durch die Abbildung auf eine Landschaft mit selbst-organisierenden Karten [4, 3]. Zum anderen muss die Navigation durch diese Sammlungen für kleine Bildschirme ermöglicht werden, siehe [8, 6] für einige Ansätze.

In dieser Projektgruppe sollen mehrere Konzepte für Musiknavigation für mobile Geräte implementiert und untersucht werden. Eine geeignete Visualisierung von Kennzeichnungen bzw. Merkmalsdaten einzelner Stücke und ihres Zusammenhanges soll die Anwendung so gestalten, dass die Bedienung auch auf kleinen Bildschirmen informativ und intuitiv ist. Exploration und Sortierung eigener und fremder Musikdatenbanken wird durch die Integration in ein Spielkonzept motiviert, das die heute vorhandenen technischen Möglichkeiten mobiler Geräte voll ausschöpfen soll.



(a) MusicMiner-Visualisierung: mit einer topographischen Karte werden ähnliche Lieder nebeneinander dargestellt, das geschieht durch Transformation von Audiomeerkmalen mit selbst-organisierenden Karten (nach <http://musicminer.sourceforge.net/>, Stand 07.10.2010)

(b) Darstellung einer größeren Musiksammlung auf dem mobilen Gerät für [6]: hier jedes Lied 4 mal dargestellt und die Entfernungen zu den Referenzpunkten von vier Kategorien entsprechen der Zugehörigkeit zu diesen Kategorien

Figure 1: Beispiele für Visualisierungen vieler Musikstücke

Zu diesen gehören z.B. auch eine Bewegungssensorik und Ortungs-/Orientierungssysteme unter Ausnutzung von GPS. Allerdings sind die Rechenkapazitäten mobiler Geräte nach wie vor denen herkömmlicher Computer unterlegen. Daher können viele eigentlich wünschenswerte Ansätze wie z.B. Merkmalsextraktion von Musikstücken derzeit noch nicht in annehmbarer Zeit dezentral umgesetzt werden. Wie oder inwieweit solche Funktionalitäten in mobile Geräte eingebracht werden können ist auch Gegenstand aktueller Forschung.

Da die Einordnung von Musikstücken in eine Musikdatenbank durch die Wahl der unterschiedlichen (Meta-)Merkmale nach aktuellen Benutzervorlieben geschieht, muss eine Dimensionsreduktion durchgeführt werden, um die Datenbank als Raum wahrnehmbar zu machen, in dem man sich bewegen kann. Das vorgesehene Spiel findet nun in diesem niedrigdimensionalen (2D oder 3D) Raum statt, der jeweils eine persönliche Musikdatenbank repräsentiert. Es ist Ziel des Spieles, das zunächst für zwei Spieler vorgesehen ist, ein (z.B. zufällig) bestimmtes Musikstück beim jeweils anderen Spieler schneller aufzustöbern als dieser in der eigenen Musikdatenbank fündig wird. Das Spiel verbindet nicht nur die genannten Techniken, sondern schafft einen echten Mehrwert, denn diese Art von Spielen ist neuartig (die bisher vorhandenen Musikspiele erschöpfen sich in der Regel damit, den Takt oder die Melodie eines Liedes nachzuahmen) und ideal für mobile Endgeräte geeignet.

Zu dem Thema Musikdatenanalyse läuft aktuell am Lehrstuhl 11 ein Projekt mit dem Titel „Mehrkriterielle Optimierung der automatischen Musikklassifikation auf Basis von Merkmalen höherer Ordnung mit Methoden der Computational Intelligence“ (als Kooperation mit der Fakultät Statistik). Im Rahmen des Projektes werden z.B. evolutionäre Algorithmen zur Optimierung der Musikklassifikation eingesetzt [1], und im Laufe der letzten Jahre ist mit AMUSE ein Framework für unterschiedliche Musikanalyse-Aufgaben entstanden [7], das z.B. für die automatische Merkmalsextraktion aus Audiodaten und darauf basierende Klassifikation für die Projektgruppe eingesetzt werden kann. Außerdem soll die Gruppe vor allem durch den Kontakt mit Herrn Theimer vom Know-How der kooperierenden Firma Research in Motion (RIM, Bochum) profitieren.

Durch die Kontakte zu mehreren Lehrstühlen und Universitäten der regionalen Arbeitsgruppe für Musikdatenanalyse [2] bestehen weitere Möglichkeiten, sich interdisziplinär auszutauschen.

Aufgabe

In dieser Projektgruppe soll ein auf mobilen Geräten lauffähiges Programm entstehen, mit dem man im Rahmen eines Spieles Musikdatenbanken anhand der Merkmale der gespeicherten Musikstücke virtuell durchwandern kann. Dazu sind passende Visualisierungen sowohl der Merkmalsdaten eines Stückes (siehe Abb. 2) als auch der Daten vieler Stücke einzusetzen (Abb. 1), was zumeist eine Dimensionsreduktion erfordert.

Die Navigation in der virtuellen Welt soll durch die Bewegung des Gerätes gesteuert werden. Desweiteren ist die Realisierung virtueller Gruppen durch GPS-Ortung und die Anbindung an soziale Netzwerke wie Facebook vorgesehen. Eine interessante Erweiterung des Spielkonzeptes ist der Einspielermodus, bei dem ein virtueller, computergesteuerter Gegner die Funktion des zweiten Mitspielers übernimmt. Auf diesem Gebiet der Computational Intelligence in Games (CIG) konnte der Lehrstuhl 11 in den vergangenen Jahren einige Erfahrung sammeln (z.B. [5]).

Da die benötigten Techniken ein weites Spektrum in der Informatik abdecken und auch statistische Methoden wie das multi-dimensional Scaling (MDS) zum Einsatz kommen sollen, ist eine umfangreiche Einarbeitung notwendig. Diese wird zum Teil in der einleitenden Seminarphase erfolgen, aber

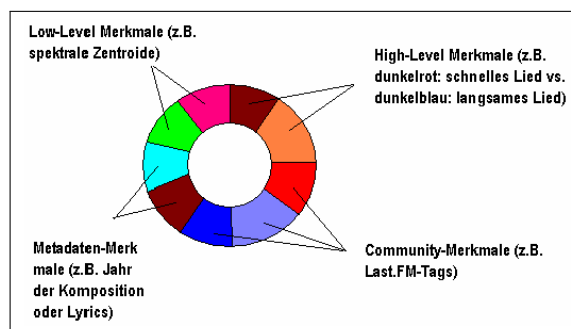


Figure 2: Beispiel für Visualisierung eines Liedes mit Merkmalskreis: die Ausprägungen der Merkmale werden durch Farbe dargestellt

auch während der PG-Laufzeit wird es Aufgabe der Studierenden bleiben, sich in neue Teilgebiete einzuarbeiten bzw. vorhandenes Wissen z.B. aus der digitalen Bildverarbeitung einzubringen. Dabei ist durchaus beabsichtigt, kreative Ansätze aus der PG in das vorhandene Konzept einzuarbeiten. Es erscheint aber nicht sinnvoll, von der ganzen PG die Einarbeitung in alle Techniken zu erwarten, so dass eine gewisse Spezialisierung der einzelnen Teilnehmer durchaus erwünscht ist.

Die Projektgruppe soll zunächst einzelne Komponenten implementieren bzw. aus vorhandenen Bibliotheken oder Programmpaketen adaptieren, die später benötigt werden, um ein lauffähiges Gesamtsystem zu erstellen. Dabei liegt der Schwerpunkt im ersten Semester hauptsächlich auf der Arbeit mit Musikdatenbanken und Merkmalsdaten, deren Weiterverarbeitung durch Clustering und Dimensionsreduktion, sowie der Bereitstellung der Navigationsfunktionalität aus der Bewegungssensorik des mobilen Geräts. Allerdings sollte zum Ende des Semesters ein lauffähiges Demonstrationsprogramm erstellt werden, um die Schwachstellen besser zu lokalisieren und den Aufwand für das zweite Halbjahr besser planen zu können.

Im zweiten Semester wird dann das eigentliche Spiel aus den vorhandenen Komponenten realisiert und erprobt. Hier ist darauf zu achten, dass es nicht nur spielbar sein soll, sondern auch Spaß machen muss. Dazu ist eine ausführliche Erprobung mitsamt inkrementeller Verbesserung notwendig, vermutlich auch eine gewisse Anpassung an das Verhalten von Spielern (z.B. wie in [9]). Um einen Mehrspielermodus überhaupt zu ermöglichen bzw. Spieler für den Zweispielermodus zu finden, soll gleichzeitig die Einbettung des Spieles in soziale Netzwerke wie Facebook implementiert werden. Als alternatives oder unterstützendes Konzept kann die Selbstortung der mobilen Geräte über GPS verwendet werden, um geeignete Mitspieler oder auch Gegenspieler zu finden (so könnten z.B. zwei räumlich getrennte Mannschaften im Turniermodus gegeneinander antreten).

Über diese rein inhaltlichen Ziele hinausgehend ist es natürlich auch Aufgabe der PG, sich einige 'soft skills' anzueignen, die für die Arbeit in Gruppen unabdingbar sind. So werden die Präsentation von Sachverhalten und Konzepten ebenso trainiert wie die Entscheidungsfindung und Arbeitsorganisation im Team, außerdem soll auch das Arbeiten mit wissenschaftlichen Ergebnissen dritter und, soweit möglich, die wissenschaftliche Verwertung eigener Ergebnisse geübt werden.

6. Teilnahmevoraussetzungen

Notwendig

- Mindestens eins der Module:
Digitale Bildverarbeitung, Datenvisualisierung, Mensch-Maschine-Interaktion, Eingebettete Systeme oder Einführung in die Computational Intelligence
- Programmierung in Java
- Gute Deutsch- und Englischkenntnisse

Wünschenswert

- Softwaretechnologie, -entwicklung
- Programmierung mobiler Endgeräte, insbesondere graphischer Oberflächen
- Grundlegende Kenntnisse der Statistik
- LaTeX-Kenntnisse zur Dokumentation
- Kenntnis des Betriebssystems Unix

7. Minimalziel

Die von der PG zu erstellende Software soll mindestens folgende Komponenten beinhalten:

- Merkmalsextraktion bzw. Verwaltung von Merkmalsdaten; hier soll soweit möglich auf existierende Software zurückgegriffen werden (z.B. AMUSE).
- Merkmalskreis oder vergleichbare Visualisierung der Merkmalsdaten eines Musikstückes.
- Mindestens eine Methode zur Dimensionsreduktion der Merkmalsdaten auf 2D/3D, z.B. Self-Organizing Maps (SOM) oder Multi-Dimensional Scaling (MDS).
- Umsetzung von Bewegungssensordaten in eine Navigation im virtuellen 2D/3D-Raum.
- Integration der genannten Komponenten in ein Spiel, bei dem mindestens 2 Spieler versuchen, in der Musikdatenbank des jeweils anderen Spielers mithilfe der Bewegung im projizierten 2D/3D-Raum ein vorgegebenes Musikstück möglichst schnell zu finden.
- Erstellung eines Zwischen- und Endberichts.

Außerdem ist es ausdrücklich Ziel der PG, die jeweils gewählten Ansätze hinsichtlich der Verwendbarkeit in möglichen Anschlussprojekten zu bewerten.

8. Literatur

- [1] B. Bischl, I. Vatulkin, and M. Preuss. Selecting small audio feature sets in music classification by means of asymmetric mutation. In *Proceedings of the 11th International Conference on Parallel Problem Solving From Nature (PPSN)*, pages 314–323, Krakow, 2010.
- [2] H. Blume, B. Bischl, M. Botteck, C. Igel, R. Martin, G. Rötter, G. Rudolph, W. Theimer, I. Vatulkin, and C. Weihs. Towards automated dynamic structuring of huge music archives on mobile devices. *Submitted to IEEE Signal Processing Magazine (Accepted in the 1st round of review)*, 2011.
- [3] P. Knees, M. Schedl, T. Pohle, and G. Widmer. Exploring music collections in virtual landscapes. *IEEE MultiMedia*, 14(3):46–54, 2007.
- [4] F. Mörchen, A. Ultsch, M. Thies, I. Löhken, M. Nöcker, C. Stamm, N. Efthymiou, and M. Kümmerer. Visualizing timbre distances of music as topographical maps. Technical Report 47, Data Bionics Research Group, Philipps-Universität Marburg, Germany, 2005.
- [5] M. Preuss, N. Beume, H. Danielsiek, T. Hein, B. Naujoks, N. Piatkowski, R. Stüer, A. Thom, and S. Wessing. Towards intelligent team composition and maneuvering in realtime strategy games. *IEEE Transactions on Computational Intelligence and AI in Games*, 2(2):82–98, 2010.
- [6] W. Theimer, I. Vatulkin, M. Botteck, and M. Buchmann. Content-based similarity search and visualization for personal music categories. In *Proceedings of the 6th International Workshop on Content-Based Multimedia Indexing (CBMI)*, pages 9–16, 2008.
- [7] I. Vatulkin, W. Theimer, and M. Botteck. Amuse (advanced music explorer) - a multitool framework for music data analysis. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, Utrecht, 2010.
- [8] F. Vignoli and S. Pauws. A music retrieval system based on user driven similarity and its evaluation. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR)*, pages 272–279, 2005.
- [9] G.N. Yannakakis. Preference Learning for Affective Modeling. In *Proceedings of the Int. Conf. on Affective Computing and Intelligent Interaction (ACII09)*, Amsterdam, The Netherlands, September 2009.

9. Rechtlicher Hinweis

Die Ergebnisse der Projektarbeit und die dabei erstellten Software sollen der Fakultät für Informatik uneingeschränkt für Lehr- und Forschungszwecke zur freien Verfügung stehen. Darüber hinaus sind keine Einschränkungen der Verwertungsrechte an den Ergebnissen der Projektgruppe und keine Vertraulichkeitsvereinbarungen vorgesehen.

Projektgruppenantrag

1. Thema

Konzeption und Entwicklung eines sicheren Cloud-Basierten Internetbanking-Systems mit anschließender Sicherheitsanalyse auf Basis von Business Process Mining

2. Zeitraum

SoSe 2011 und WiSe 2011/12

3. Umfang

8 SWS pro Semester

4. Veranstalter

- Prof. Dr. Jan Jürjens, Informatik XIV, Campus Süd, GB IV, Raum 302, Tel.: 7953, E-Mail: jan.jurjens@cs.tu-dortmund.de (1. Hauptbetreuer)
- Dipl.-Inform. Stephan Faßbender, Informatik XIV, Campus Süd, GB IV, Raum 326, Tel.: 2690, E-Mail: stephan.fassbender@cs.tu-dortmund.de
- Dr.-Ing. Holger Schmidt, Informatik XIV, Campus Süd, GB IV, Raum 304, Tel.: 2412, E-Mail: holger.schmidt@cs.tu-dortmund.de (2. Hauptbetreuer)
- externe Partner: Fraunhofer ISST Dortmund, Koordination durch Prof. Dr. Jan Jürjens

5. Aufgabe

Motivation und Hintergrund:

Im Software Engineering steht die Entwicklung hochqualitativer Softwareprodukte im Mittelpunkt. Dabei wird die Qualität eines Softwaresystems nicht nur durch die Implementierung bestimmt, sondern sie wird maßgeblich von den konzeptionellen Softwareentwicklungsphasen, d.h. der Anforderungsanalyse und dem Entwurf, beeinflusst. Heutige Softwaresysteme gewinnen zunehmend an Komplexität was unter anderem den gestiegenen Sicherheitsanforderungen geschuldet ist. Besonders das Qualitätsmerkmal *Sicherheit* im Sinne von *Security* gewinnt aufgrund neuer Technologien wie des Cloud Computings immer mehr an Bedeutung.

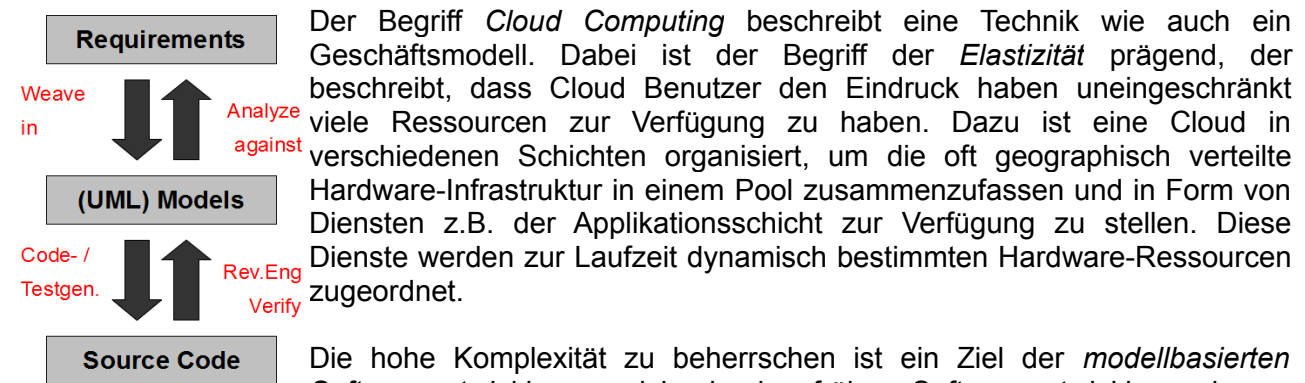


Abbildung 1:
Modellgetriebene
Softwareentwicklung

Die hohe Komplexität zu beherrschen ist ein Ziel der *modellbasierten Softwareentwicklung*, welche in den frühen Softwareentwicklungsphasen Modelle verwendet, die in den nachgelagerten Phasen schrittweise in eine Implementierung überführt werden. Der Prozess (siehe Abbildung 1) beginnt mit der Erhebung und der Analyse von Anforderungen, die anschließend in eine abstrakte Systemspezifikation überführt werden. Dann werden schrittweise weitere Modelle entwickelt bis ein Detaillierungsgrad erreicht ist, der die Implementierung des Softwaresystems ermöglicht.

Im modellbasierten Software Engineering werden Sicherheitsanforderungen wie z. B. Vertraulichkeit, Integrität, Authentizität, u. a. spezifiziert innerhalb von UML (Unified Modeling Language) Modellen oder im Quellcode (Java oder C) in Form von Annotationen. In der Arbeitsgruppe Software Engineering wird am Lehrstuhl XIV der Fakultät für Informatik die UML-Spracherweiterung *UMLsec* entwickelt. Sicherheitseigenschaften wie z. B. die Vertraulichkeit von

Information oder Datenintegrität können mit Hilfe von UMLsec in UML Modellen annotiert werden. Die Gültigkeit dieser Sicherheitseigenschaften kann durch ein vorliegendes Werkzeug verifiziert werden.

Aufgabenstellung:

Die Vermittlung der Lerninhalte über sichere Softwareentwicklung soll anhand eines praxisnahen Beispiels umgesetzt werden. Dazu werden die Studierenden im Zuge dieses Projekts ein Internetbanking-System als Cloud Computing-Dienst umsetzen. Anschließend erfolgt die Ausführung von bankspezifischen Geschäftsprozessen in der Cloud, wie Online-Überweisungen ausführen, Kontostandhistorie abfragen, etc., dessen Aktivitäten in Logdateien protokolliert werden. Aus diesen Logdateien erfolgt dann ein Reengineering der IST-Prozesse mit Hilfe von Business Process Mining, die mit den vorher definierten SOLL-Prozessen verglichen werden, um Anomalien und potentielle Sicherheitslücken zu identifizieren.

Es wird eine Beschreibung der Bank inklusive der wichtigsten Geschäftsprozesse vorgelegt. Anhand dieser Informationen werden die Studierenden eine Anforderungsanalyse durchführen. Nachfolgend ist eine Cloud-basierte Anwendung zu konzipieren und zu implementieren. Dazu gehört auch ein geeignetes Loggingsystem zu konzipieren und zu implementieren, z.B. mittels Jlog, um später Business Process Mining Ansätze verwenden zu können.

Die Entwicklung der Anwendung soll anhand des Multipfad-Vorgehensmodells durchgeführt werden (siehe Abbildung 2). Nachdem sich die Projektteilnehmer in einem Seminarteil in die Modellierung mit UMLsec, Entwicklung in Cloud Computing-Umgebungen und Business Process Mining Analysen eingearbeitet haben, beginnt die Projektgruppe mit der Voruntersuchung (V1). Hier soll eine Anforderungserhebung und -analyse anhand der vorhandenen Spezifikation und Prozesse der Bank durchgeführt sowie eine Spezifikation mit UMLsec entworfen werden. Im fachlichen Entwurf (V2) sollen die von der Anwendung unterstützten Prozesse und die grafische Benutzeroberfläche modelliert werden. Im Systementwurf (V3) sollen die Studierenden eine technische Perspektive einnehmen und die zuvor spezifizierten Prozesse in eine Cloud-basierte Architektur überführen. Die so erstellte Architektur wird dann in der Implementierungsphase und in der Konfigurationsphase umgesetzt, wobei auf maßgeschneiderte Komponenten (V4) und vorgefertigte Komponenten (V5) zurückgegriffen werden soll.

Die Stabilisierungsphase umfasst das Testen der Software anhand von zuvor erstellten Testfällen, Anwendungsbeispielen und Business Process Mining. Außerdem wird zuvor in einem Seminarteil Wissen über verschiedene Software-Angriffstypen (SQL-Injection, Cross-Side-Scripting, Buffer Overflows, usw.) vermittelt, das in konkrete Testfälle für die Software umgesetzt wird. Das eigentliche Testen der Software soll als Hackers Workshop durchgeführt werden.

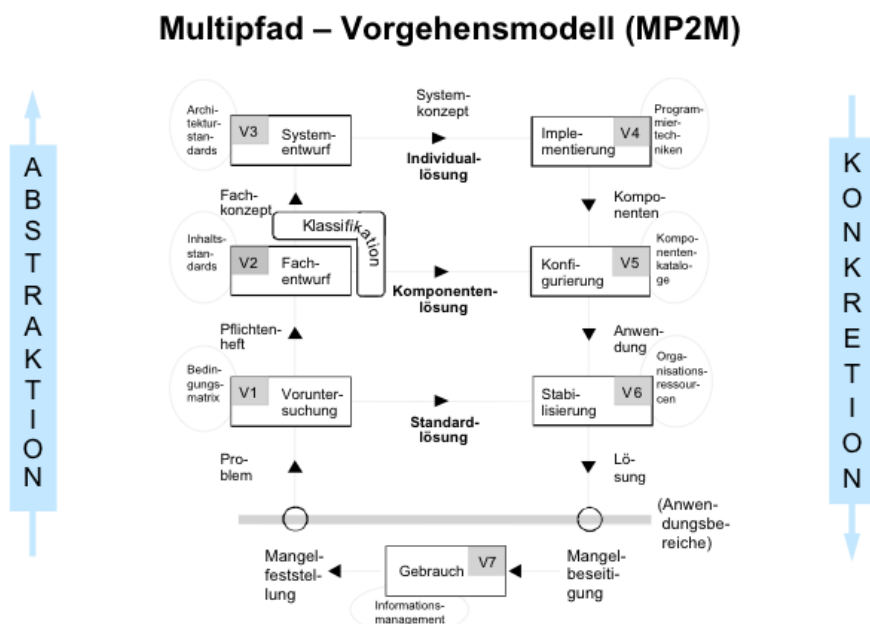


Abbildung 2: Multipfad-Vorgehensmodell

Im einzelnen soll die Projektgruppe

- eine vollständige Dokumentation in Form eines Pflichtenheftes und eines Fach- und Systemkonzepts erstellen,
- eine anhand von Testfällen für funktionsfähig erklärte prototypische Implementierung,
- einen Zwischenbericht zur Dokumentation der Arbeiten im ersten PG-Semester sowie einen Endbericht zur Dokumentation der PG-Ergebnisse erstellen

6. Teilnahmevoraussetzungen

- Softwaretechnik (V)
- Kenntnisse in objektorientierter Programmierung (V)
- Mensch-Maschine-Interaktion , Modellgestützte Analyse und Optimierung , Formale Methoden des Systementwurfs (M)
- Kenntnisse in UML (W)
- Kenntnisse in UMLsec (W)
- Kenntnisse in Cloud-Entwicklung (W)
- Kenntnisse in Sicherheitstesten von Software (W)

Legende: (M) Mindestens eine; (V) Voraussetzung; (W) wünschenswert

7. Minimalziele

- Konzeption einer Cloud-basierten Internetbanking-Anwendung
- Prototypische Realisierung der Konzepte
- Sicherheitsevaluation der implementierten Systeme
- Soll- und Ist-Vergleich der ausgeführten Prozesse mittels Business Process Mining
- Erstellung eines Endberichts
- Fachgespräch

8. Literatur

- Jürjens, J. (2005). *Secure systems development with UML*. Springer.
- Jürjens, J. (2005). *Sound methods and effective tools for model-based security engineering with UML*. In G.-C. Roman, W. G. Griswold, & B. Nuseibeh (Eds.), *Proceedings of the international conference on software engineering (ICSE)* (pp. 322-331). ACM Press.
- Sierra, K; Bates, B.; Schulten, L. & Buchholz, E. (2006). *Java von Kopf bis Fuß*. O'Reilly.
- Sommerville, I. (2007). *Software engineering*. 8th edition. Addison-Wesley.
- The Cloud Security Alliance (2010). *Security Guidance for Critical Areas of Focus in Cloud Computing*. <http://www.cloudsecurityalliance.org/>.
- Mather, T.; Kumaraswamy, S.; Latif, S. (2009). *Cloud Security and Privacy*. O'Reilly.
- Streitberger, W.; Ruppel, A. (2009): *Cloud Computing Sicherheit – Schutzziele. Taxonomie. Marktübersicht*, Fraunhofer Institut für Secure Information Technology (SIT).
- Armbrust, M.; Fox, A.; Griffith, R.; Joseph, A. D.; Katz, R. H.; Konwinski, A.; Lee, G.; Patterson, D. A.; Rabkin, A.; Stoica, I.; Zaharia, M. (2009). *Above the Clouds: A Berkeley View of Cloud Computing*, Technical report, UCB/EECS-2009-28, EECS Department University of California, Berkeley.
- Van der Aalst, W.M.P.; de Medeiros, A.K.A. (2005). *Process Mining and Security: Detecting Anomalous Process Executions and Checking Process Conformance*, Electronic Notes in Theoretical Computer Science.

9. Rechtlicher Hinweis

Die Ergebnisse der Projektarbeit und die dabei erstellte Software sollen der Fakultät für Informatik uneingeschränkt für Lehr- und Forschungszwecke zur freien Verfügung stehen. Darüber hinaus sind keine Einschränkungen der Verwertungsrechte an den Ergebnissen der Projektgruppe und keine Vertraulichkeitsvereinbarungen vorgesehen.

Projektgruppenantrag

1. PG-Thema: NetSensLog – Netzwerkbasierte Sensorfusion für einen Fahrzeugschwarm in der Intralogistik

2. PG-Zeitraum: SS 2011, WS 2011/2012

3. PG-Umfang: 8 SWS pro Semester

4. PG-Veranstalter:

Dr. Frank Weichert, Informatik VII (Graphische Systeme),
OH16, R.121, Tel.: 6122, E-Mail: frank.weichert@tu-dortmund.de

Constantin Timm, Informatik XII (Entwurfsautomatisierung Eingebetteter Systeme),
OH16, R.105, Tel.: 6144, E-Mail: constantin.timm@cs.uni-dortmund.de

Andreas Kamagaew, Fraunhofer-Institut für Materialfluss und Logistik,
Joseph-von-Fraunhofer-Str. 2-4, Tel.: 9743-127, E-Mail: andreas.kamagaew@iml.fraunhofer.de

5. PG-Aufgabe:

a) Motivation: In Logistik-/Distributionszentren und Lagereinrichtungen kommen zunehmend „Fahrerlose Transportsysteme“ (FTS) zum Einsatz [10]. Eine große Herausforderung innerhalb einer solchen logistischen Einheit ist es, dass gleichzeitig möglichst viele Transportsysteme aktiv sein können und diese Fahrzeuge (s. Abb. 1(a)) zudem möglichst autonom arbeiten, um den Zeitaufwand und die Kosten erheblich zu reduzieren. Das zugrundeliegende Konzept, bei dem ein sogenannter Schwarm von autonomen Transportfahrzeugen tätig ist (s. Abb. 1(b)), wird als „Zellulare Fördertechnik“ bezeichnet und ist Gegenstand aktueller Forschung [9].

Ein Hauptkriterium zum nachhaltigen Einsatz ist die Güte der vorhandenen Sensorik. Deshalb arbeiten die Lehrstühle Informatik VII und XII und das Fraunhofer-Institut für Materialfluss und Logistik (IML) gemeinsam an einem Projekt zur Optimierung und Verbesserung der vorhandenen Technik. Zurzeit werden Kollisionsvermeidung, Lokalisation und Navigation mit einem hybriden Sensorikkonzept, bestehend aus Laserscanner, Odometriedaten und einer Funkortung durchgeführt. Mithilfe der Funkortung und der Odometrie erfolgt eine grobe Lokalisation auf der Lagerfläche, mit dem Einsatz des Laserscanners sind präzise Andockmanöver an Übergabestationen, den sogenannten Kommissionierplätzen, möglich (s. Abb. 1(b)). Durch die Verfügbarkeit neuer Sensortechnologien, wie z.B. PMD-Kameras (Photonic Mixer Device - eine 3D-Kamera), auch Time-of-Flight-Kamera genannt [3], besteht die Möglichkeit, die sensorischen Einheiten der Fahrzeuge zu erweitern, um die Umgebung und damit auch die Lokalisation und Kollisionsvermeidung jetzt auch in drei Dimensionen durchzuführen.

Ausgehend von dieser Problemstellung ist die Zielsetzung dieser Projektgruppe der Entwurf und die Implementierung einer netzwerkbasierter Vision-System-Lösung für die neuartige PMD-Technologie zur 3D-Lokalisation und Kollisionsvermeidung und der Einsatz dieses Systems auf mehreren autonomen Fahrzeugen. Zur weiteren Steigerung der Messgüte ist zudem die Fusion von PMD-Messungen mit den Bilddaten einer 2D-CCD-Kamera zu realisieren. Die Fusion der Bilddaten der 2D- und 3D-Kameras wird zunächst auf einem Industrie-PC durchgeführt, der mittels Grafikkarte für eine echtzeitfähige Bildverarbeitung erweitert wurde. Anschließend sollen die fusionierten Sensorinformationen im Netzwerk des Fahrzeugschwarms in abstrahierter Form genutzt werden, um die Qualität der Sensormessungen und damit die Genauigkeit der Lokalisation zu verbessern und die Sicherheit der Kollisionsvermeidung zu erhöhen. Parallel dazu sollen die mit diesem Vision-System ausgestatteten Fahrzeuge durch ein geeignetes Kommunikationskonzept mit einem



(a) Reales Fahrzeug



(b) Exemplarisches Szenario

Abbildung 1: (a) Foto eines „Fahrerlosen Transportsystems“ und (b) virtuelle Darstellung eines exemplarischen Szenarios für ein Kommissioniersystem

synchronen deterministischen Zeitverhalten zu einem verteilten Sensornetzwerk verbunden werden. Hierfür kann eine am Fraunhofer IML entwickelte Middleware zur Bereitstellung von heterogenen Sensordaten verwendet werden, die um eine effiziente Fahrzeug-zu-Fahrzeug-Kommunikation (F2F-Kommunikation) zu erweitern ist. Eine Möglichkeit zur F2F-Kommunikation ist die Nutzung einer Peer-to-Peer-Middleware (P2P) [8]. Peer-to-Peer-Netzwerke erlauben den Aufbau von verteilten Anwendungen ohne Nutzung von bei Client-Server-Architekturen üblichen zentralen Servern und erlaubt es den einzelnen Teilnehmern (Schwarmfahrzeug) dieses Netzwerks (Peers) direkt miteinander zu kommunizieren. Der zentrale Unterschied zu den „Client/Server“-Architekturen ist daher, dass jedes Schwarmfahrzeug somit als sogenannter Servant agiert, d.h. er ist Server und Client in einem. Die so zur Verfügung stehenden Sensordaten werden in einer späteren Projektphase zur Entwicklung von Algorithmen zur Objekterkennung und Kollisionsvermeidung auf der Fahrzeughardware eingesetzt. Die Ergebnisse dieser Arbeitspakete sollen zu einer netzwerkbasierter Vision-System-Lösung kombiniert und zur Verifikation auch visualisiert werden.

b) Aufgabenbeschreibung: Aus Sicht der Informatik umfasst das Gesamtkonzept der Projektgruppe (s. Abb. 2) zwei wesentliche Aufgabengebiete, die in einem Optimierungs- und Validierungszyklus eingebunden sind. Ein Teilgebiet fokussiert sich dabei auf die digitale Bildverarbeitung und bildbasierte Mustererkennung, das zweite beinhaltet den Entwurf und die Implementierung von Konzepten Eingebetteter Systeme. Dabei werden die verschiedenen Aufgabengebiete während der gesamten Projektgruppe parallel bearbeitet. Für die digitale Bildverarbeitung sind folgende Verarbeitungsschritte angedacht:

- **Kamerakalibrierung:** Neben der klassischen photogrammetrischen Kalibrierung von RGB-Kameras zur Bestimmung von Kameraparametern ist ein zusätzlicher Kalibrierungsschritt zur Korrektur systematischer, PMD-typischer wie z.B. sogenannten Flying Pixeln (Objektgrenzen innerhalb eines Pixels) notwendig. Ansätze zur Korrektur sind u.a. die Verwendung von Lookup-Tabellen oder B-Spline basierten Korrekturfunktionen [4].
- **Vorverarbeitung der Messdaten:** Die Kompensierung von Artefakten in der Distanzmessung, beispielsweise Ausreißer bei räumlichen Diskontinuitäten (Flying Pixel), ist Thema dieses Arbeitspakets. Ansätze hierzu sind u.a. Non-Local-Means (NLM)-Filter oder die Aufskalierung der Entfernungsdaten [6].
- **Sensordatenfusion:** Teilaspekte sind die Kombination von Tiefeninformationen und Grauwertdaten einer PMD-Kamera und die Kombination mit Daten einer zusätzlichen RGB-Kamera. Realisierungsmöglichkeiten bestehen in der Projektion der kalibrierten PMD-Daten auf eine 2D-Bildebene [5] oder im Hochskalieren des Tiefenbilds mithilfe von bilateralen Filtern auf die Ortsauflösung des RGB-Bildes [11].
- **Mustererkennung und Klassifikation:** Ausgehend von den vorverarbeiteten Sensordaten

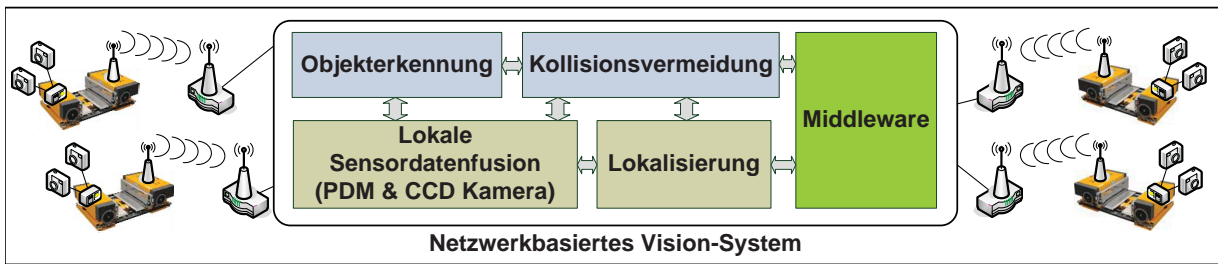


Abbildung 2: Schematische Darstellung der Projektgruppenaufgaben

ist es die Intention, die verschiedenen Fahrzeuge, deren Beladungszustand und die Übergabestationen automatisch zu erkennen. Dies könnte durch unüberwachte als auch überwachte Klassifikationsverfahren geschehen, welche auf Merkmalen arbeiten, die durch Methoden der Mustererkennung, wie z.B. „Pattern-Matching“, aus den Sensordaten extrahiert werden [7].

- **Visualisierung:** Die multiplen Daten der Sensoren und dessen Detektionsergebnisse sollen visualisiert werden – unter Beachtung des Projektfortschritts eventuell auch als interaktives 3D Visualisierungssystem.

Das zweite Teilgebiet der Projektgruppe geht von den o.g. Bildverarbeitungsalgorithmen aus und zielt auf eine effiziente Realisierung dieser Methoden auf Eingebetteten Systemen und auf eine performante Kommunikation zwischen den Fahrzeugen ab. Hierzu sind folgende Arbeitspunkte umzusetzen:

- **Multicore-Programmierung für Eingebettete Systeme:**
 - (a) Grafikkartenimplementierung: Um eine echtzeitfähige Verarbeitung der Bildverarbeitungsalgorithmen zu garantieren und möglichst flexible Verarbeitungsplattformen verfügbar zu haben, sind die Sensordaten der verschiedenen Kamerasysteme mittels GPGPU-Programmierung (General Purpose Computing on GPUs) zu analysieren. Dazu soll OpenCL, eine Programmiersprache für sowohl Grafikkarten bzw. auch homogene Multicoresysteme eingesetzt werden [2].
 - (b) OpenCL-Programmierung: Zur Vereinfachung der OpenCL-Programmierung soll ein Source-to-Source-Compiler angepasst und zusätzlich das OpenCL Embedded Profile für die ausgewählte Plattform auf dem Schwarmfahrzeug umgesetzt werden, um OpenCL-Kernel vor der Ausführung zu kompilieren.
- **Effiziente Kommunikation:** Zur Abstraktion der eigentlichen Netzwerkstruktur und zur Kommunikation der Fahrzeuge untereinander (Aufbau des Sensornetzwerks), soll eine Middleware auf den Fahrzeugen bereitgestellt werden [8], die folgende Aspekte berücksichtigt:
 - (a) Direkte und transparente Kommunikation zwischen den Fahrzeugen mit einem deterministischem Zeitverhalten ohne zentrale Kontrolle
 - (b) Gruppenkommunikation: Bildung und Kommunikation innerhalb einer Adhoc-Gruppe. Diese Funktionalität ermöglicht beispielsweise beim Umfahren eines möglichen Kollisionspunktes, Fahrzeuge im Umfeld zu „benachrichtigen“.
 - (c) Abstraktion der Kommunikationsschnittstelle: Für die Kommunikation mit den Fahrzeugen stehen mehrere Funkschnittstellen zur Verfügung. Diese sollen über eine vereinheitlichte Schnittstelle erreichbar sein.

Bei der Umsetzung dieser Anforderungen in ein Softwaresystem sollen verschiedene Entwurfsphasen beachtet werden [1] und adäquate Werkzeuge (Quellcodeverwaltung, Wiki-System, Bugtracking) eingesetzt werden. Dieses geht mit der angestrebten modularen Umsetzung einher, welche eine spätere einfache Erweiterung des Systems erlaubt, beispielsweise zur globalen Lokalisierung der Fahrzeuge und zur Erstellung einer Umgebungskarte verwendet. Das gewählte Thema gibt einen engen thematischen Anschluss zu verschiedenen Forschungsarbeiten in Richtung digitale

Bildverarbeitung/Mustererkennung am LS VII, Softwareentwicklung für Eingebettete Systeme am LS XII und hoch aktuellen Fragestellungen im Kontext des IMLs und erlaubt damit den Teilnehmenden, sich für eine Reihe von möglichen Diplomarbeiten an den Lehrstühlen zu qualifizieren.

6. Teilnahmevoraussetzungen:

- Eine der Vorlesungen *Mensch-Maschine-Interaktion (Graphische Systeme)*, *Digitale Bildverarbeitung* oder *Graphische Datenverarbeitung* [V]
- Eine der Vorlesungen *Eingebettete Systeme*, *Rechnergestützter Entwurf von Mikroelektronik* oder *Rechnerarchitektur* [V]
- Kenntnisse in einer Programmiersprache, z.B. Java, C++ oder C [V]

Legende: [V] vorausgesetzt, [W] wünschenswert

7. Minimalziel:

- Implementierung eines echtzeitfähigen Bildverarbeitungssystems zur Fahrzeugdetektion und Kollisionsvermeidung
- Implementierung einer Middleware für den Fahrzeugschwarm

8. Literatur:

- [1] H. Balzert. *Lehrbuch der Software-Technik*. Spektrum Akademischer Verlag, 1997.
- [2] Khronos OpenCL Working Group. *The OpenCL Specification*, 2008.
- [3] A. Kolb, E. Barth, R. Koch, and R. Larsen. Time-of-flight sensors on computer graphics. *Proc. Eurographics (State-of-the-Art Report)*, 2009.
- [4] M. Lindner and A. Kolb. Calibration of the Intensity-Related Distance Error of the PMD TOF-Camera. *SPIE: Intelligent Robots and Computer Vision XXV*, 6764:6764–6835, 2007.
- [5] M. Lindner and A. Kolb. Data-Fusion of PMD-Based Distance-Information and High-Resolution RGB-Images. volume 1, pages 121–124, 2007.
- [6] M. Lindner, M. Lambers, and A. Kolb. Sub-Pixel Data Fusion and Edge-Enhanced Distance Refinement for 2D/3D Images. *Dynamic 3D Imaging Workshop*, 2007.
- [7] H. Niemann. *Klassifikation von Mustern*. 2. edition, 2003.
- [8] J. Schmutzler, S. Rohde, and C. Wietfeld. Integration of Wireless Peer-to-Peer Sensor Networks with Embedded Web Services. In *14. ITG Fachtagung -Mobilkommunikation*, volume 215, pages 47–52, Osnabrück, Germany, May 2009. VDE.
- [9] M. ten Hompel and L. Nagel. Zellulare Transportsysteme: Den Dingen Beine machen im "Internet der Dinge". *Information Technology*, it 50, Nr.1:pp. 59–65, 2008.
- [10] M. ten Hompel and T. Schmidt. Warehouse management. *Springer-Verlag, Berlin Heidelberg New York*, 2008.
- [11] Q. Yang, R. Yang, J. Davis, and D. Nister. Spatial-Depth Super Resolution for Range Images. *IEEE Computer Vision and Pattern Recognition*, pages 1–8, 2007.

9. Rechtliche Hinweise:

Die Ergebnisse der Projektarbeit inklusive der dabei erstellten Software sollen der Fakultät für Informatik uneingeschränkt zur freien Forschung und Lehre zur Verfügung stehen. Darüber hinaus sind keine Einschränkungen der Verwertungsrechte an den Ergebnissen der Projektgruppe und keine Vertraulichkeitsvereinbarungen vorgesehen.

Projektgruppenantrag

Lehrstuhl für Förder- und Lagerwesen (FLW)

- 1. Titel** *FiLeCC - Field-Level Computation Cloud:*
Verlagerung von übergeordneten Steuerungsfunktionen in die Feldebene von Materialflusssystemen mithilfe eines Cloud-basierten Ansatzes.
- 2. Zeitraum** Sommersemester 2011 und Wintersemester 2011/2012
- 3. Umfang** 8 SWS pro Semester
- 4. Veranstalter** *Sascha Feldhorst*, EF73, Raum 140, Tel.4073, safe@flw.mb.tu-dortmund.de
Christian Mosblech, EF73, Raum 125, Tel.3450, chmo@flw.mb.tu-dortmund.de
Prof. Dr. Heiko Krumm, Informatik, LS 4

5. Aufgabe

Im Zuge der vorgeschlagenen Projektgruppe soll eine Software-Anwendung konzipiert und realisiert werden, die vorhandene Berechnungs- und Speicherressourcen eines automatisierten Materialflusssystems nutzt, um eine verteilte Ausführungsumgebung für übergeordnete Steuerungs- und Optimierungsfunktionen anzubieten.

5.1 Motivation

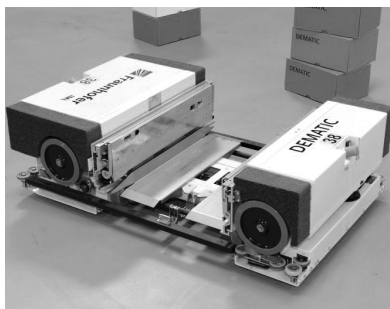
Bei der Auslegung logistischer Systeme muss bereits zu Beginn des Projekts festgelegt werden, ob das System manuell, halbautomatisch oder vollautomatisch betrieben werden soll. Bei dieser Entscheidung gilt es gerade in Hochlohnländern wie Deutschland, die Anschaffungs- und Betriebskosten der verschiedenen Alternativen sorgfältig gegeneinander abzuwägen. Aufgrund ihrer hohen Anschaffungskosten muss für vollautomatische Systeme gewährleistet werden, dass sie für einen möglichst langen Zeitraum betrieben werden können. Die Betriebsdauer einer vollautomatischen Anlage abzuschätzen, gestaltet sich jedoch in den letzten Jahren zunehmend schwieriger. Dies lässt sich darauf zurückführen, dass die Dynamik logistischer Problemstellungen nachhaltig gestiegen ist, die zugrundeliegenden Prozesse häufiger angepasst werden müssen und vollautomatische Systeme diesbezüglich die unflexibelste der drei Varianten darstellen. Grund dafür ist, dass Umbaumaßnahmen an automatisierten Materialflusssystemen häufig aufwändig und teuer sind, da viele Funktionen über verschiedene Systemebenen verteilt sind [4], so dass selbst augenscheinlich lokal beschränkte Änderungen Auswirkungen auf das Gesamtsystem haben können. Dies schlägt sich nicht zuletzt in umbaubedingten Stillstandszeiten nieder. Außerdem erfordern viele Umbaumaßnahmen anlagenspezifisches Wissen des Herstellers, weshalb das durchführende Unternehmen i. d. R. nicht frei am Markt gewählt werden kann [8].

Um dieser Problematik zu begegnen, ist die Modularisierung der Produktions- und Materialflussanlagen ein vielversprechender Ansatz [9, 6]. Die Modularisierung betrifft dabei neben den elektrischen und mechanischen Komponenten auch die Steuerungssoftware. Insbesondere die Dezentralisierung der vormals zentralen Steuerungsfunktionen, wie z. B. Routing oder Stauvermeidung, ist ein wichtiger Bestandteil dieses Ansatzes. Während derzeit die Steuerungsfunktionen in wenigen monolithischen Programmen umgesetzt sind, die entweder in Speicherprogrammierbaren Steuerungen oder Materialflussrechnern ablaufen, erfordert eine strikte Modularisierung eine Verlagerung dieser Funktionen in die gesteuerten Gewerke. Es ist heute möglich, die einzelnen Komponenten eines Materialflusssystems mit eingebetteten Computern auszustatten. Dadurch sind Komponenten, wie z. B. Stetigförderer oder Regalbediengeräte, in der Lage miteinander zu kommunizieren und eigenständig Software auszuführen. Dies befähigt sie, aktuelle Prozessinformationen, wie ihre Betriebsbereitschaft, in Echtzeit zur Verfügung zu stellen und sich aktiv an der Steuerung einer Anlage zu beteiligen.

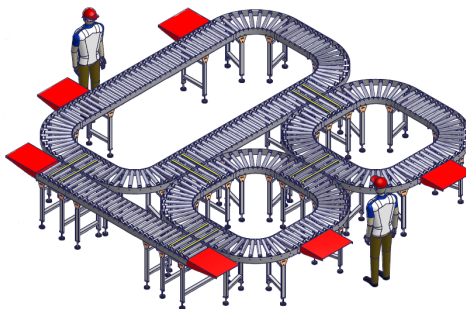
Als Folge werden bereits heute Einzelgewerke oder Anlagenteile mit eingebetteten Industrie-PCs (IPCs) ausgestattet, die zur Ausführung von Steuerungssoftware eingesetzt werden (vgl. Abb. 1). Im Rahmen verschiedener Forschungsprojekte (wie z. B. das BMBF-Projekt *Internet der Dinge*) und auch im Zuge der Projektgruppe 475¹ [2] wurden in Dortmund erste Demonstratoren dezentraler

¹Die PG475 "Management kooperierender Web Services" wurde in Kooperation vom LS4 und dem FLW durchgeführt.

Steuerungssysteme entwickelt, die den gezielten Transport von Materialflussobjekten durch eine Anlage ohne eine zentrale Steuerinstanz realisieren. Als Ausführungsumgebung wird auch hier auf IPCs zurückgegriffen. Durch die ständige Weiterentwicklung der verbauten IPCs steigt die verfügbare Rechenleistung in der Feldebene seit Jahren konstant an. Während noch vor einigen Jahren IPCs mit 266 MHz Prozessoren eingesetzt wurden, sind heute bereits Systeme mit weit über 1 GHz und vereinzelt sogar mit mehreren Prozessorkernen verfügbar. Beim Bau von neuen Anlagen und auch bei der Modernisierung von bestehenden Systemen steht also immer leistungsfähigere Hardware zur Verfügung. Unter Berücksichtigung der gängigen Überdimensionierung von industriellen Systemen zeichnet es sich schon heute ab, dass auf der Feldebene von automatisierten Anlagen zukünftig ein ungenutzter Kapazitätsüberschuss an Berechnungs- und Speicherressourcen entstehen wird.



(a) Multishuttle Move



(b) Modulare Stetigförderanlage



(c) KARIS-Einzelement[4]

Abbildung 1: Beispiele für modulare Fördersysteme mit eigenen Berechnungsressourcen

5.2 Problemstellung

In modularisierten, dezentralen Systemen bringen die einzelnen Gewerke zukünftig neben der Software zu ihrer operativen Steuerung auch Software für die übergeordneten Steuerungsfunktionen mit, d. h. die Module enthalten auch Teile des bisherigen Leitsystems, um möglichst unabhängig zu sein. Die Gesamtfunktionalität der Steuerungssoftware wird also in unabhängige Teilfunktionalitäten zerlegt, welche sich dann zur Laufzeit miteinander integrieren lassen, beispielsweise in Form von lose gekoppelten Diensten. In diesem Zusammenhang ergibt sich die Fragestellung, in welchen Ausführungsumgebungen diese Software-Komponenten ablaufen sollen. Anstatt des Einsatzes zentraler Server, bietet es sich an, die freien Ressourcen der Feldebene zu nutzen. Vor diesem Hintergrund erscheint ein Cloud-basierter Ansatz sinnvoll, in dem die freien Ressourcen unter den vorhandenen Diensten verteilt werden. Die Grundlage des „Cloud Computing“ ist die Bereitstellung von IT-Ressourcen aller Art als Dienste über ein Netzwerk. Diese Ressourcen können vom reinen Anbieten von Speicherplatz über die Nutzung verteilter Rechenleistung bis hin zu virtuellen Computern inklusive Betriebssystem und installierter Software reichen (nach [1]). Ein Cloud-basierter Ansatz für die Verwaltung der Berechnungsressourcen der Feldebene eines automatisierten Systems muss im Wesentlichen zwei Dinge beachten: Zum einen benötigen einige Dienste einen direkten Zugriff auf die automatisierungstechnischen Komponenten des zugehörigen Gewerks, so dass nicht jeder Dienst auf einen anderen IPC verteilt werden kann. Zum anderen darf der Betrieb der operativen Steuerungen durch die Cloud-Dienste nicht beeinträchtigt werden. Deshalb muss für eine angemessene Verteilung von Rechenaufgaben und Speicherplatz innerhalb des Gesamtsystems gesorgt werden.

5.3 Aufgabenstellung

Gesucht ist also eine Cloud-ähnliche Struktur, die über eine genormte Schnittstelle den Zugriff auf Ressourcen wie Speicherplatz und Rechenkapazität bietet. Die Verteilung soll dabei lastabhängig durchgeführt werden, so dass IPCs, die bereits stärker mit Steuerungsaufgaben ausgelastet sind, weniger Aufgaben der zentralen Cloud-Dienste übernehmen. Die grundsätzliche Systemstruktur kann Abb. 2 entnommen werden.

Innerhalb dieses Projektes soll eine Cloud-Architektur genutzt werden, um Services in Form von Softwarebausteinen (SaaS) zur Verfügung zu stellen. Dabei liegt der Fokus der Entwicklung nicht bei den konkreten Services, sondern bei der Cloud-Computing-Plattform. Diese soll es einem Software- bzw. Steuerungsentwickler ermöglichen, auf fest definierte Strukturen und Schnittstellen zuzugreifen. Die Rechenleistung wird dabei innerhalb der Feldebene auf die verfügbare Einzelrechenleistung verteilt.

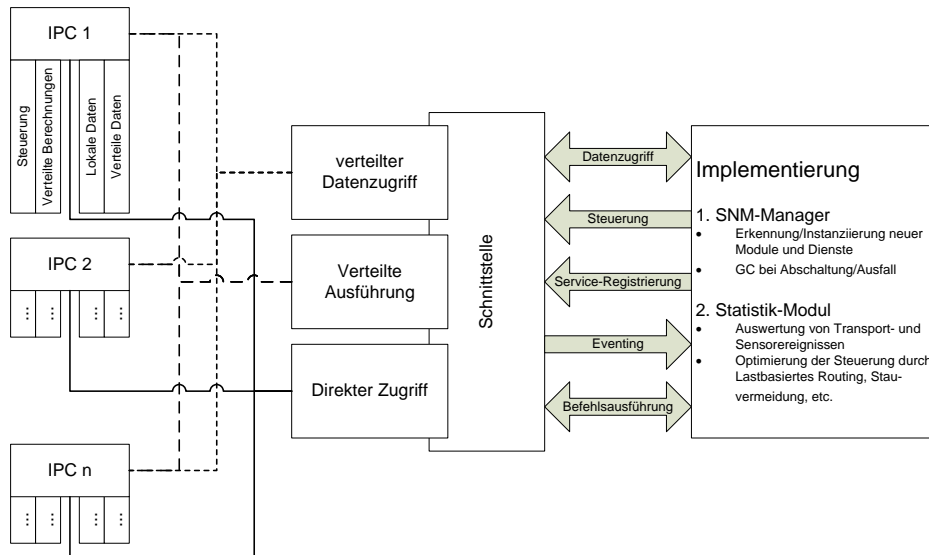


Abbildung 2: Schematischer Aufbau der Software

Diese grundlegende Idee kommt im Bezug auf die Datensicherheit einer sog. „Private Cloud“ gleich, welche sich dadurch auszeichnet, dass sowohl die Logik als auch die Daten innerhalb eines Unternehmens angesiedelt sind. Dadurch wird ermöglicht, dass die z. T. geschäftskritischen Daten eines Unternehmens nicht an Dritte weitergegeben werden, wie z. B. an den Anbieter von Cloud-Ressourcen im Internet. Der Einsatz einer Cloud-Infrastruktur auf Systemen abseits von leistungsfähigen Serversystemen ist derzeit Stand der Forschung. [10] beispielsweise untersucht die Möglichkeiten, die Ressourcen in der Umgebung einer einzelnen Person zu nutzen, um eine „Personal Cloud“ aufzubauen. Dies soll es ermöglichen, die Einsatzgebiete von leistungsstarken Systemen (z. B. Desktops) zusammenzufassen und über leistungsschwächere Systeme (z. B. Mobil-Telefon) zugänglich zu machen.

Die Programmierung der Cloud-Umgebung soll mit der Evaluierung und gegebenenfalls einer geeigneten Systemauswahl beginnen. In diesem Schritt liegt ein besonderer Fokus auf der Recherche von bestehenden, freien (Open Source) Lösungen, die das Potential zur anforderungsgerechten Anpassung bieten. Vielversprechende Systeme sind beispielsweise Gridgain (siehe [5]) oder OpenNebula (siehe [7]). Die zu berücksichtigenden Bewertungskriterien sind über eine gezielte Literatur-Recherche festzulegen und zu gewichten. Randbedingungen sind hierbei zusätzlich der mögliche Einsatz auf portablen Systemen und die Nutzung der Programmiersprache JAVA. Die geleisteten Vorarbeiten am Lehrstuhl FLW setzen auf eine dienstbasierte Kommunikationstechnologie [3], welche innerhalb der verfügbaren Cloud-Systeme auf ihre Einsetzbarkeit untersucht werden soll. Im nächsten Schritt sollen mögliche Service-Implementierungen zusammengestellt und auf ihre Interaktionsanforderungen hin überprüft werden. Im Fokus steht hier die Verwaltung von sogenannten *Smart Neighbourhood Modulen* (SNMs), ein Konzept, das an der TU Dortmund entwickelt wurde. Dem Grundgedanken nach sollen komplette industrielle Anlagen zukünftig aus diesen Modulen bestehen. Ein SNM ist dabei ein modularer Systembaustein bestehend aus Mechanik, Automatisierungskomponenten sowie der zugehörigen Software. Die Anforderungen, die sich beim Management von SNMs ergeben, ermöglichen die Spezifikation der Schnittstelle für die spätere Cloud-Steuerung. Abschließend sollen die gefundenen Spezifikationen in einer Cloud-Steuerungssoftware umgesetzt und zwecks Erprobung in die Versuchsanlage am FLW integriert werden.

6 Teilnahmevoraussetzungen

Für die Teilnahme an der Projektgruppe sind folgende Voraussetzungen zu erfüllen („Mindestens eine Veranstaltung verpflichtend (P)“, „Wünschenswert (W)“, „Voraussetzung (V)“):

- Methodische Grundlagen des Software Engineering (P)
- Modellierung und Analyse eingebetteter und verteilter Systeme (P)
- Rechnernetzanwendungen (P)
- Kenntnisse in OOP mit Java (V)
- Webtechnologien (W)
- Service Computing (W)

7 Minimalziel

Die Projektgruppe hat die folgende Zielsetzung, wobei die ersten beiden Ziele das Minimalziel darstellen und das dritte Ziel lediglich optional ist:

- *Entwicklung und Realisierung einer funktionierenden Software zur Cloud-Steuerung:* Diese Steuerung soll auf der Versuchsanlage des Lehrstuhls für Förder- und Lagerwesen implementiert und erprobt werden.
- *Implementierung von mindestens einem Modul:* Dieses Modul soll die Schnittstelle und damit die Ressourcen der Cloud nutzen. Funktional soll es das Management von „*Smart Neighbourhood Modulen*“ ermöglichen und somit beispielsweise eine dynamische Erweiterung bzw. Reduktion des Systems erkennen und die betroffenen Dienste laden bzw. entladen können.
- *Optionale Implementierung eines Statistik-Moduls:* Dieses Modul soll Laufzeitdaten von den logistischen Gewerken erfassen und statistisch auswerten können. Die so gewonnenen statistischen Kennwerte sollen in die Steuerung einfließen und den Materialfluss optimieren.

8 Referenzen

- [1] C. Baun, M. Kunze, J. Nimis, and S. Tai, *Cloud Computing: Web-basierte dynamische IT-Services*, ser. Informatik im Fokus. Berlin: Springer, 2010.
- [2] B. Brill, M. Cagliyan, S. Feldhorst, M. Fiedler, J. Gaeb, M. Heinemann, T. Karakoyun, J. Konieczny, T. Mayer, H. Niehoff, T. Peper, and L. Rüsing, “PG475: Management kooperierender Web Services: RoastedKit,” TU Dortmund, Tech. Rep., 2006.
- [3] S. Feldhorst, S. Libert, M. ten Hompel, and H. Krumm, “Integration of a legacy automation system into a SOA for devices,” in *ETFA: Proceedings of the 14th IEEE International Conference on Emerging Technologies & Factory Automation*. IEEE Press, 2009, pp. 782–789.
- [4] K. Furmans, C. Nobbe, and F. Schönung, “Dezentrale Materialflusststeuerungen,” in *Tagungsband zum 5. BVL Wissenschaftssymposium Logistik Strukturwandel in der Logistik*, ser. Schriftenreihe Wirtschaft & Logistik, W. Delfmann, Ed. DVV Media Verl., 2010, pp. 307–315.
- [5] I. GridGain Systems, “Gridgain = compute + data + cloud,” Website, 13.10.2010 2010. [Online]. Available: <http://www.gridgain.com/>
- [6] W. Günthner and M. ten Hompel, *Internet der Dinge in der Intralogistik*, ser. VDI-Buch. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2010.
- [7] O. P. L. (OpenNebula.org), “Opennebula: The open source toolkit for cloud computing,” Website, 13.10.2010 2010. [Online]. Available: <http://www.opennebula.org/>
- [8] V. Sadowsky and S. Feldhorst, “Paket Royale - Dezentrales Steuerungskonzept für das Internet der Dinge,” in *Tagungsband zum 5. BVL Wissenschaftssymposium Logistik Strukturwandel in der Logistik*, ser. Schriftenreihe Wirtschaft & Logistik, W. Delfmann, Ed. DVV Media Verl. Dt. Verkehrs-Verl., 2010, pp. 296–306.
- [9] M. ten Hompel, R. Jünemann, L. Nagel, and T. Schmidt, *Materialflusssysteme: Förder- und Lagertechnik*, 3rd ed., ser. VDI-Buch. Springer-Verlag Berlin Heidelberg, 2007.
- [10] X. Wu, W. Wang, B. Lin, and K. Miao, “Composable IO: A Novel Resource Sharing Platform in Personal Clouds,” in *Cloud Computing*, ser. Lecture notes in computer science, M. Jaatun, G. Zhao, and C. Rong, Eds. Springer Berlin / Heidelberg, 2009, vol. 5931, pp. 232–242.

9 Rechtliche Hinweise

Die Ergebnisse der vorgeschlagenen Projektgruppe und die dabei erstellte Software sollen der Fakultät für Informatik und der Fakultät Maschinenbau uneingeschränkt für Lehr- und Forschungszwecke zur freien Verfügung stehen. Darüber hinaus sind keine Einschränkungen der Verwertungsrechte an den Ergebnissen der Projektgruppe und keine Vertraulichkeitsvereinbarungen vorgesehen.