

Dresden, den 28. Januar 2019

Übergabe Teilprojekt C05 (SFB Transregio)

Ziel des Dokuments:

- Übersicht Ist-Stand in der Außen- und Innensicht sowie Arbeitspakete der Phase 3
- Zusammenfassung von (möglichen) Forschungsthemen aus WI-Sicht
- Interne Hinweise zur Zusammenarbeit innerhalb des SFB
- Verweis auf wichtige Quellen

Zielstellung des Teilprojekts:

- Bewertung der im SFB Transregio 96 erarbeiteten Korrektur- und Kompensationsverfahren
- Bewertung aus technischen Sicht (Steigerung der Genauigkeit)
- Bewertung aus wirtschaftlicher Sicht (Aufwand, organisatorische Konsequenzen)
- Herausforderungen:
 - Grundlagenforschung (SFB) vs. Anwendungsbezug (C05)
 - Umfassende Bewertung kaum möglich
 - Fremde Fachdomäne, Abhängigkeit von Zuarbeit
 - Kommunikation im Gesamtprojekt
 - Implementierung einer eigenen „Hidden Research Agenda“ (nach außen im SFB sind die Erwartungen an uns andere als die, welche wir in der WI-Forschung platzieren können!)

Arbeitspakete und Aufgaben:

Arbeitspaket	Wichtigste Aufgaben	Mögliche WI-Forschungsthemen
AP 1 (Bewertungsmethode)	<ul style="list-style-type: none"> • Bewertungsmetrik tatsächlich konzipieren und implementieren (!) • Einbindung in Gesamtmethode • Beschreibung der Gesamtmethode • Implementierung über „Recommender“-Modul • Priorität: Sehr hoch (elementares Projektziel!) 	<ul style="list-style-type: none"> • Methodenentwicklung, multi-kriterielle Bewertung über mehrere Dimensionen hinweg • Agile Artefakte als Tools (Design Science Research)
AP 2 (Domain Manger)	<ul style="list-style-type: none"> • Grundsätzlich: Dauerhafter Ausbau der Datenbasis • Iterative Weiterentwicklung • Operativ: Deployment verbessern, Upgrades Typo3 • Student auf Modell-Transformationsprozess ansetzen (UML → SQL → Typo3-TCA), ggf. Alternativbewertung • Priorität: Gering (muss gemacht werden, ist aber wissenschaftlich gelöst; Transformation an Studenten rausgeben) 	<ul style="list-style-type: none"> • (Konfigurations-Management) • „Real Model-Driven Engineering“, Transformationsprozess
AP 2-2 (Rule Engine)	<ul style="list-style-type: none"> • Studium der DMN-Spezifikation und von FEEL • Abschließende Bewertung der Eignung • Implementierung in DOCASYS (Rule Engine) • Dateneinarbeitung • Priorität: Hoch (hochinteressantes Forschungsthema) 	<ul style="list-style-type: none"> • Grundlagenforschung zur Regelspezifikation und zur Anwendung in Bewertungsprojekten (ggf. Recherche zu

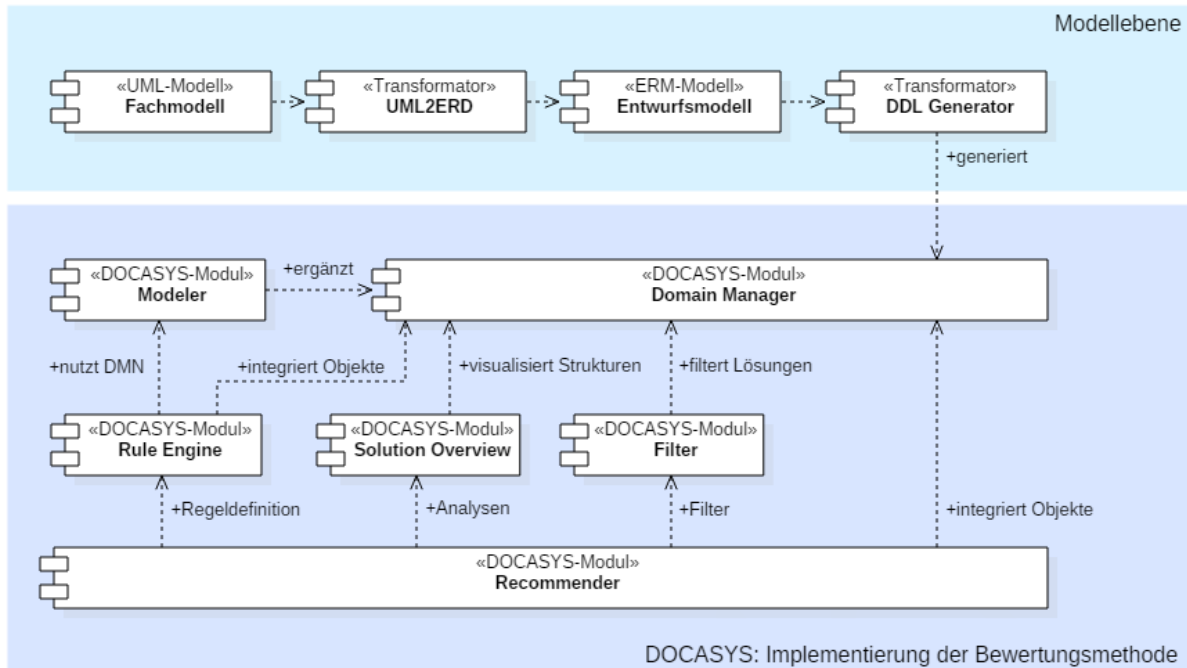
	innerhalb der WI, für SFB allerdings etwas weniger relevant)	Expertensystemen etc. erforderlich!)
AP 2-3 (Recommender)	<ul style="list-style-type: none"> Konzeption und Implementierung der Bewertungsmetrik vor der Hintergrund der praktischen Nachfrage Wenn genügend Daten: KI-Ansatz zum automatischen Lernen und Verbessern („Nice to have“-Thema) Priorität: Mittel (hier eher Tooling, die zugrunde liegenden Regeln liegen im AP 2-2 und die wiederum zugrunde liegende Metrik in AP 1, wo jeweils die konzeptionellen Lösungen erarbeitet werden müssen) 	<ul style="list-style-type: none"> Siehe AP 1 (KI und Machine Learning)
AP 3 (Referenz-Anwendungsfälle und Lastfälle)	<ul style="list-style-type: none"> Spezifikation Standardbeschreibung auf Basis der Domainobjekte im Ist-Zustand Anleiten der IWM-Kollegen (Fachinhalte müssen von dort kommen) Begleiten der Dateneingabe im DOCASYS (muss primär vom IWM gemacht werden!) Priorität: Sehr hoch (elementare Konzepte innerhalb des SFB, Potential zu wichtigem Direktergebnis des SFB) 	<ul style="list-style-type: none"> Nichts Direktes
AP 5 (Langzeitwirkung)	<ul style="list-style-type: none"> Bei uns eher ein Ausbau des Domain Managers Anleiten der IWM-Kollegen (Grundlagen gibt es in Phase 1 mit der VDMA-Richtlinie) Priorität: Eher gering 	<ul style="list-style-type: none"> Nichts Direktes
AP 6 (Kombinierbarkeit)	<ul style="list-style-type: none"> Abstimmung mit IWM Integration der Gesamtaufwände und ggf. Erweiterung der Bewertungsmethode Priorität: Mittel 	<ul style="list-style-type: none"> Nichts Direktes

Interne Hinweise:

- **Domain Model:**
 - Das Fachmodell der Domäne und dessen Implementierung in DOCASYS ist der **Kern von allem**
 - Was dort nicht existiert und beschrieben ist, existiert auch nicht in unserem Projekt (!)
 - Anders kann die Kommunikation nicht vereinheitlicht und der homogene Vergleichsrahmen nicht aufgebaut werden
- **Zusammenarbeit mit IWM im Teilprojekt:**
 - Seitens der IWM-Projektpartner muss in Phase 3 mehr Zuarbeit kommen, diese ist ggf. nachdrücklich einzufordern (in Phase 2 war diese unzureichend und die Ergebnislast lag bei uns)
 - Dies betrifft v.a.:
 - Einarbeitung von Daten im Domain Manager (v.a. bei technischen Aspekten, die wir nicht bewerten können)
 - Einarbeitung der Untersuchungen und Vermessungen seitens IWM
 - Aussagen zur technischen Leistungsfähigkeit von Verfahren
 - Eigene Publikationsleistungen (formell hat WiSe in Phase 1 und 2 über 80% der geforderten Publikationsleistung erbracht!)
 - Ggf. ist die Zuarbeit auf höherer Ebene einzufordern (Chef-Ebene)
- **Stellung innerhalb des SFB:**
 - Im Plenum schlagen uns meist gewisse Skepsis und Vorbehalte entgegen (BWL-Perspektive, Fachfremdheit, Unlösbarkeit des Problems, Beratersicht etc.), die

nicht offen ausgesprochen wird, aber latent – z.B. in Fragen nach Vorträgen – zum Tragen kommt

- Im bilateralen Austausch auf Ebene der Projektbearbeiter finden wir aber eine offene und aufgeschlossene Art vor; es ist aber immer die Zielstellung und unsere Perspektive zu betonen, um die Kollegen „abzuholen“
- **Generelle Argumentationslinie (bei Vorträgen, Vollversammlungen etc.):**
 - Bewertung erfordert Beschreibbarkeit
 - Bewertung → Vergleich → Homogenisierung als Vergleichsrahmen
 - Beschreibbarkeit → Beschreibung
 - Ansatz: Modellgestütztes Management und Domain-Driven Engineering
 - Der Weg ist das Ziel (Methode)
 - Erstes Ziel: Sichtbarmachung von Faktoren, Aspekten, Zusammenhängen
 - Zweites Ziel: Tatsächlicher Vergleich
 - Generell sollte das immer etwas hart und urteilende „Bewerten“ abgeschwächt werden und der Fokus auf die Sichtbarmachung des impliziten, in den Teilprojekten sowie in der Domain liegenden Wissens gelegt werden
- **Stellung von DOCASYS:**
 - DOCASYS ist nach außen hin der Outcome des Teilprojekts
 - Im Näheren ist es jedoch eher ein Tool und Mittel zum Zweck, um alle relevanten Daten der Teilprojekte in unserem Domänenrahmen zu verorten und „Fakten zu schaffen“
 - Wissenschaftlich ist DOCASYS nicht (!) unser Hauptziel
 - Unser Hauptziel ist die Gesamtmethode und die Konzeption der (im Tool umgesetzten) Module wie z.B. die Bewertungsmetrik oder das Domänenmodell insgesamt
 - Im Design Science kann das ggf. als iterative, artefakt-gestützte Forschung eingehen, eine direkte Positionierung ist noch nicht erfolgt
- **Bezug zu aktuellen Entwicklungen und Relevanz-Betonung:**
 - Die Relevanz der Projektarbeit ist vor dem Hintergrund aktueller Trends und Themen der Digitalisierung (Industrie 4.0, KI, Machine Learning etc.) zu betonen
 - Diese Themen erfordern IT-gestützte Strukturierungs- und Dokumentationsrahmen (zuerst losgelöst vom folgenden Zweck wie z.B. einer Bewertung)
 - C05 liefert einen solchen Ansatz, der de facto generisch einsetzbar ist
- **Publikationen:**
 - In Phase 1 und Phase 2 wurden seitens WiSe vor allem generische Publikationen im Kontext von Modellierungssprachen eingebracht, die streng genommen keinen unmittelbaren sondern einen mittelbaren Projektbezug haben (siehe Publikationsliste)
 - In Phase 3 sollte diese Strategie dahingehend geändert werden, dass auch Publikationen im ingenieurwissenschaftlichen Metier avisiert werden (hier kann eher auf das Tooling abgestellt werden, nicht auf unsere WI-Methoden)
 - In der WI sollten dann weiterhin eher methodische Themen eingebracht werden und der SFB jeweils als Demonstrationsbeispiel dienen
- **Einordnung in Grundlagenforschung (DFG):**
 - DFG-Projekte (z.B. ein SFB) sind Grundlagenforschungsprojekte
 - C05 adressiert aber de facto aus dem Problem heraus die Anwendung
 - Argumentativ ist hier auf die grundlegende Forschungsfrage in unserem Teilprojekt abzustellen und auf die fehlenden Ansätze und Methoden zu verweisen
 - Bewertung findet meist rein-numerisch statt
 - Umgang mit sich entwickelnden Domänen unklar



- Frühere Bewertung bzw. Vergleichsführung nicht möglich

Technische Aspekte in Bezug auf DOCASYS:

- Argumentationslinie pro Webapplikation (Cloud):
 - Verteilter Zugriff, Flexibilität, moderne Technologien (alles verlagert sich in die Cloud), Nutzung von Standards, Rückgriff auf bestehende Architekturen, frühe Lauffähigkeit und kurze Deployment-Zyklen (siehe agile Prinzipien)
- Architekturansatz:
 - Standards und Standardsoftware (Eigenentwicklungen minimieren, Austauschbarkeit sicherstellen, keine typische „Uni-Software“ bauen)
 - Typo3 Content Management Framework (Standard, Verbreitung und v.a. strukturierte Erweiterbarkeit)
- Implementierung:
 - Typo3 Version 7.6 LTS (wird seit 12.2018 nicht mehr weiterentwickelt!)
 - Upgrade und Migration auf Typo3 8 LTS oder Typo3 9 LTS mittelfristig nötig (Migrationsaufwand)
 - Einführung Versionierungssystem und Deploymentsystem prüfen (Versionierung, Backup, Applikationsmanagement etc.), derzeit per „Hand“
 - Kernaufbau: Eine Datenbank, ein Dateisystem
- Tools
 - Fachmodellierung: StarUML (standardkonformes UML-Modellieren) mit eigener UML2ERD-Extension (Erzeugung ERD-Modelle) und DDL-Generator (Code- Erzeugung SQL)
 - Webentwicklung: z.B. PhpEd (Remote auf Server programmieren)
 - Modellierung mit BPMN, CMMN, DMN: Camunda Modeler (standardkonform; Plugin in DOCASYS verbaut)
- Alternative Architekturen:
 - Dedizierte Eigenentwicklung zur Umsetzung des Transformationsprozesses ist erfolgt auf Basis des Zend Frameworks (PHP)
 - Architektur: Zend, Doctrine, Doctrine Reverse Mapper (SQL zu ORMs)
 - Wurde 2016/2017 entwickelt (Prototyp) und final als ungeeignet bewertet (hat sehr viel Zeit gekostet!)
 - Implementierungsaufwand zu hoch (es musste alles entwickelt werden!)

- Alternative: Generische Ansätze in der Web-Entwicklung mit geeigneten Tools, um schnell Modelle zu erzeugen (z.B. <https://github.com/ZF-Commons/ZfcAdmin>)
- Alternative: Andere Frameworks (Symfony, ruby-basierte Frameworks)
- Alternative: Java-basierte Frameworks (Hibernate etc.)
- Alternative: NEOS CMS (generische Data Nodes, allerdings wenig Dokumentation)

Wichtige Publikationen des Teilprojekts:

Objekt	Zweck
Antrag Phase 2	Struktur, Arbeitspakete und Argumentation generell
Braun et al. (2015): Proposal for Requirements-Driven Design Science Research	Könnte für einen DSR-orientierten Paper relevant sein, Stichwort "Artefakte als Vehikel für wissenschaftlichen Progress"
Braun & Esswein (2014): Entwicklung einer BPMN-Extension für ressourcen-intensive Prozesse im Maschinenbau	Prozess und Ressourcen, Implementierung in der BPMN (ggf. relevant für modellbasierte Ressourcen-Analyse)
Wiemer et al. (2014): Reduzierung thermisch bedingter Verlagerungen des TCP von spanenden Werkzeugmaschinen	Gesamtansatz (ist teilweise schon etwas überholt)
Wiemer et al. (2015) Technical and Economic Benchmarking Guideline for the Compensation and Correction of Thermally Induced Machine Tool Errors	Gesamtansatz (ist teilweise schon etwas überholt)

Weiterführung erforderliches Wissen:

- Modellierungssprachen und Modellierungsmethoden
- Verständnis der Meta-Ebenen, Abstraktion, Grundprinzipien
- Syntax und Semantik (zu Semantik gibt es ein paar Abschnitte in meiner Diss)
- Verschiedene Arten der Semantik (Unterscheidung formale, informale und Gradierungen)
→ ständiges Forschungsthema
- Softwareentwicklung: MVC-Prinzip
- Grundlagen WI bzw. Typ und Instanz sowie Automatisierungsgraden
- Grundlagen WZM

Quellen und Kontakte

- Projektverzeichnis auf wisem-Server: Alle Projektdateien aller Phasen
- Wichtig: Passwortliste derzeit im kdbx-Format (Keypass, ggf. auf Mac zu migrieren)
- Sharepoint SFB: <https://sharepoint.tu-dresden.de/sites/cimtt/SFBTR96>
- Fr. Ott (Geschäftsführung SFB): Ansprechpartner bei globalen Fragen
- Fr. Müller (Assistenz Geschäftsführung): Ansprechpartner bei kleineren Anliegen
- Dr. Wiemer: Teilprojektleiter C05 auf IWM-Seite
- Innerhalb des SFB an den Kollegen der RWTH Aachen orientieren!

Grobe Architektur von DOCASYS:

- Ziel: Grobübersicht auf Basis des Dateisystems
- Zugang zum System folgt noch

Ordner	Typ	Zweck
Generelle Struktur:		

fileadmin	Typo3-Standard	Datei-Upload im Backend
fileadmin-protected	DOCASYS	Datei-Upload aus DOCASYS heraus, Zielordner für Modeler (Hintergrund: möglicher separater Zugriffsschutz)
typo3	Typo3-Standard	Typo3-Bibliothek (Ablösen durch Symlink)
typo3conf	Typo3-Standard	Typo3-Konfiguration und Speicherort für alle Extensions
typo3temp	Typo3-Standard	Temporäre Systemdateien (Cache etc.)
uploads	Typo3-Standard	User-Uploads, die nicht im Fileadmin landen
Extensions:		
extensionbuilder	Typo3-Extension	Erzeugung von Typo3-Extension, u.a. mit Domain Modeler (erspart das händische Erzeugen von Extensions, insbesondere auf Datenebene wichtig) Derzeit leider nur für Typo3 7.6 verfügbar (ein Student könnte mal einen Weiterentwicklung vornehmen und ggf. einen Roundtrip implementieren)
gridelements	Typo3-Extension	Grid-Struktur im Backend (optional, in Typo3 9 LTS obsolet)
DOCASYS:		
wise_docasys_backend	DOCASYS	Alle Backend-Funktionen und Module
wise_docasys_domain_designer	DOCASYS	Domain Model und somit Domain Manager (alle Datenobjekte und Listen), somit der Kern
wise_docasys_theme	DOCASYS	Gerüst für Frontend-Modul Backend-Konfiguration (Eingrenzung der Formulare auf diversen Listen-Seiten)
wise_docasys_domain_designer		
Classes/Domain/Model		Objekt-relationale Mapper (ORM), generiert aus Extension Builder, entstammen de facto dem Fachmodell
Classes/Domain/Repository		Repositories zu einzelnen Domain Models, d.h. komplexe Abfragen und Queries (hier, nicht im Controller!)
Classes/Domain/Userfuncs		Kleinere Snippets, um den Anzeigenamen der Domain Objects zu spezifizieren (z.B. mit Präfix)
Configuration/ExtensionBuilder		Kommt aus dem Extension Builder, kann ignoriert werden
Configuration/TCA		Definition der Formulare zu allen Domänenobjekten (initial aus Extension Builder generiert, aber manuell angepasst) Beim Update im Extension Builder sollte hier immer der TCA-Ordner gesichert werden und dann gemergt werden (wird obsolet bei Versionierungskontrolle).
Documentation.tmpl		Dokumentation, bisher ignoriert

Ressources/		Sprachsegmente (wichtig für einzelige Auswahlfelder), Icons etc.
Test/		Hier unwichtig
<i>wise_docasys_backend</i>		
Classes/Controller		Controller für jedes Modul mit entsprechenden Actions (indexAction etc.)
Classes/Utility		Helfer, siehe Typo3-Dokumentation
Classes/ ViewHelpers/Be		Backend-Helfer
Configuration/		Kann hier ignoriert werden (liegt alles im Domain Manager!)
Documentation.tmpl		Dokumentation, bisher ignoriert
Ressources/Public		Öffentlich erreichbare Ressourcen, insbesondere Bilder , Stil-Dateien und externe Bibliotheken
Ressources/Public/StyleSheets		CSS-Formatierungen
Ressources/Public/JavaScript		Zentrale Sammlung der JS-Funktionen (referenzieren dann auf gesonderte Module, siehe unten)
Ressources/Public/Icons		SVG-Grafiken
Ressources/Public/DmnJs		Bibliothek zur Modellierung mit DMN im Browser (vgl. https://github.com/bpmn-io/dmn-js)
Ressources/Public/ CytospaceJs		Bibliothek zur Darstellung der Input-Output-Netzwerke
Ressources/Public/CmmnJs		Bibliothek zur Modellierung mit CMMN im Browser (vgl. https://github.com/bpmn-io/cmmn-js)
Ressources/Public/ ChartJs		Bibliothek für Diagramme (Untersuchungsergebnisse)
Ressources/Public/BpmnJs		Bibliothek zur Modellierung mit BPMN im Browser (vgl. https://github.com/bpmn-io/bpmn-js)
Ressources/Public/Bootstrap		Bootstrap-Framework für Responsive-Funktionen (Anpassung an verschiedene Gerätegrößen)