



Objektorientierte und Funktionale Programmierung (OFP)



SS 2014



Madjid Fathi, Alexander Holland, Univ. Siegen

Email: {fathi,alex}@informatik.uni-siegen.de

Tel.: 0271/740-2311/-2276, Büro: H-A 8112/8113

Objektorientierte und Funktionale Programmierung

SS 2014

0 Organisatorisches





- Studium der Informatik an der Univ. Dortmund
 - dort 1991 promoviert,
 - 2001 habilitiert an der Techn. Univ. Illmenau
- Seit 2004 Prof. für Wissensbasierte Systeme und Wissensmanagement an der Univ. Siegen
- Forschung: Wissensmanagement Anwendungen, Wissensbasierte Systeme, Applikationen in Medizin und Maschinenbau, Computational Intelligence
- e-mail: fathi@informatik.uni-siegen.de
- Web: <http://www.uni-siegen.de/fb12/ws/mitarbeiter/>
- Tel.: 0271/740-2311
- Büro: H - A 8411
- Sprechstunde: Mo., 16:15 - 17:15 Uhr



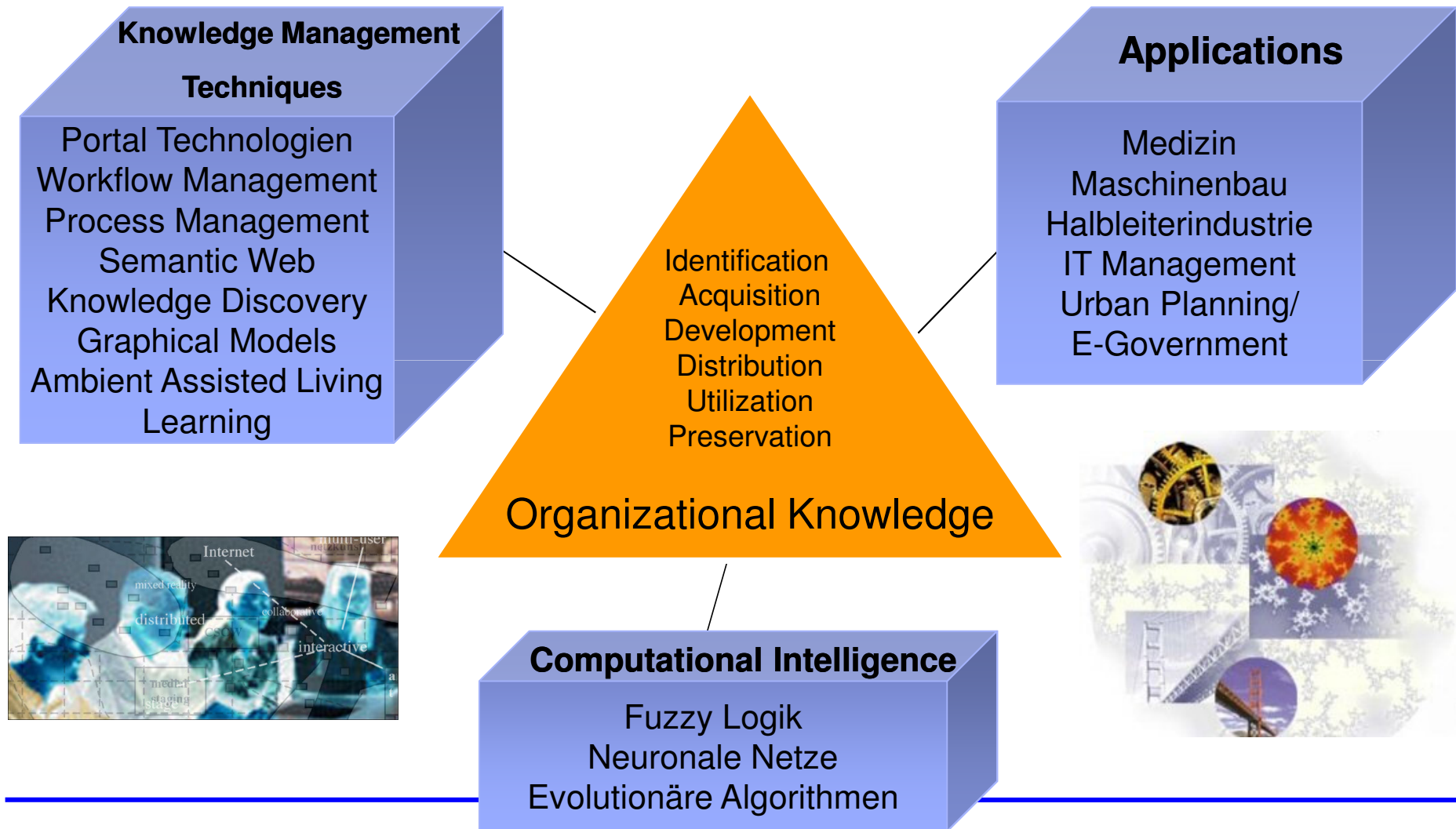
Wissensbasierte Systeme und Wissensmanagement

➤ Lehre:

- Wissensbasierte Systeme I, II (je 2V + 1Ü, 4LP)
- Wissensmanagement I, II (je 2V + 1Ü, 4LP)
- Software Engineering (je 2V + 1Ü, 4LP)
- Seminare / Proseminare: Medizinische Gruppe
- Projektgruppen im Bereich Ambient Assisted Living (u.a. Demenz), Kompetenz-Management

➤ Forschung:

- PLM und CM
- Quality Management, Business Intelligence
- Wissensmanagement, IT-Management



Vorlesungen

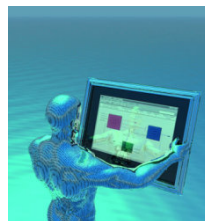
Wissensbasierte Systeme I
Wissensbasierte Systeme II
Wissensmanagement I
Wissensmanagement II
Maschinelles Lernen
Decision Making

Seminare

Knowledge Discovery
Semantic Technologies
IT Management
Advanced Knowledge
Management

Projektgruppen

Anwendungen zum Wissensmanagement
Anwendungen zum Medical Engineering

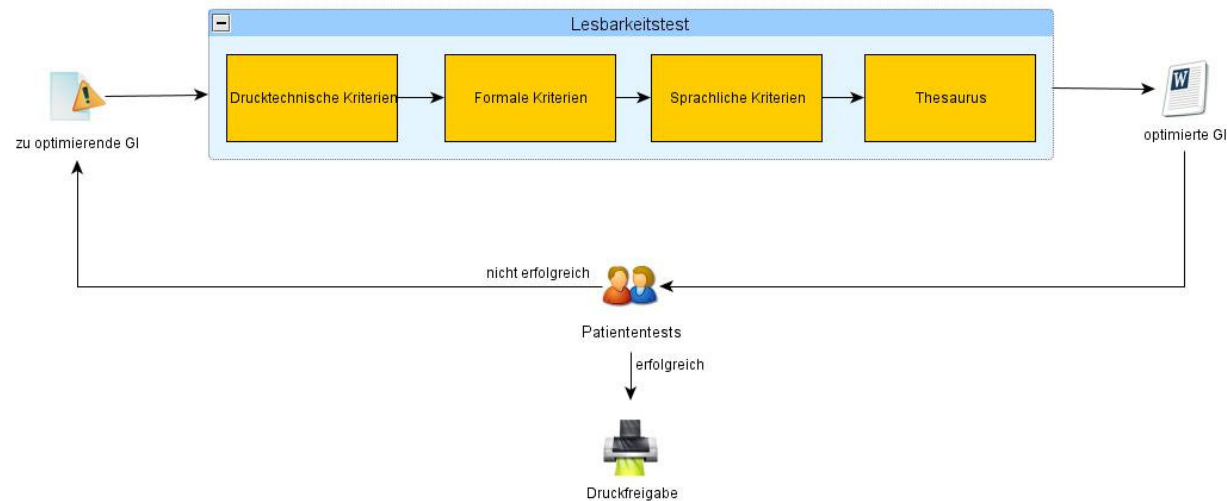


**Multi Domain Strategy for Data Integration of Business
Process Data as a Controlling Instrument with Applied Data
Mining & Visualization Methods**
(seit 2010)



**Aufbereitung und Verfügbarmachung medizinischer
Informationen in Packungsbeilagen durch Lesbarkeitstestung
zur gezielten und nachhaltigen Nutzung durch den Anwender**

BMBF
(seit 2011)



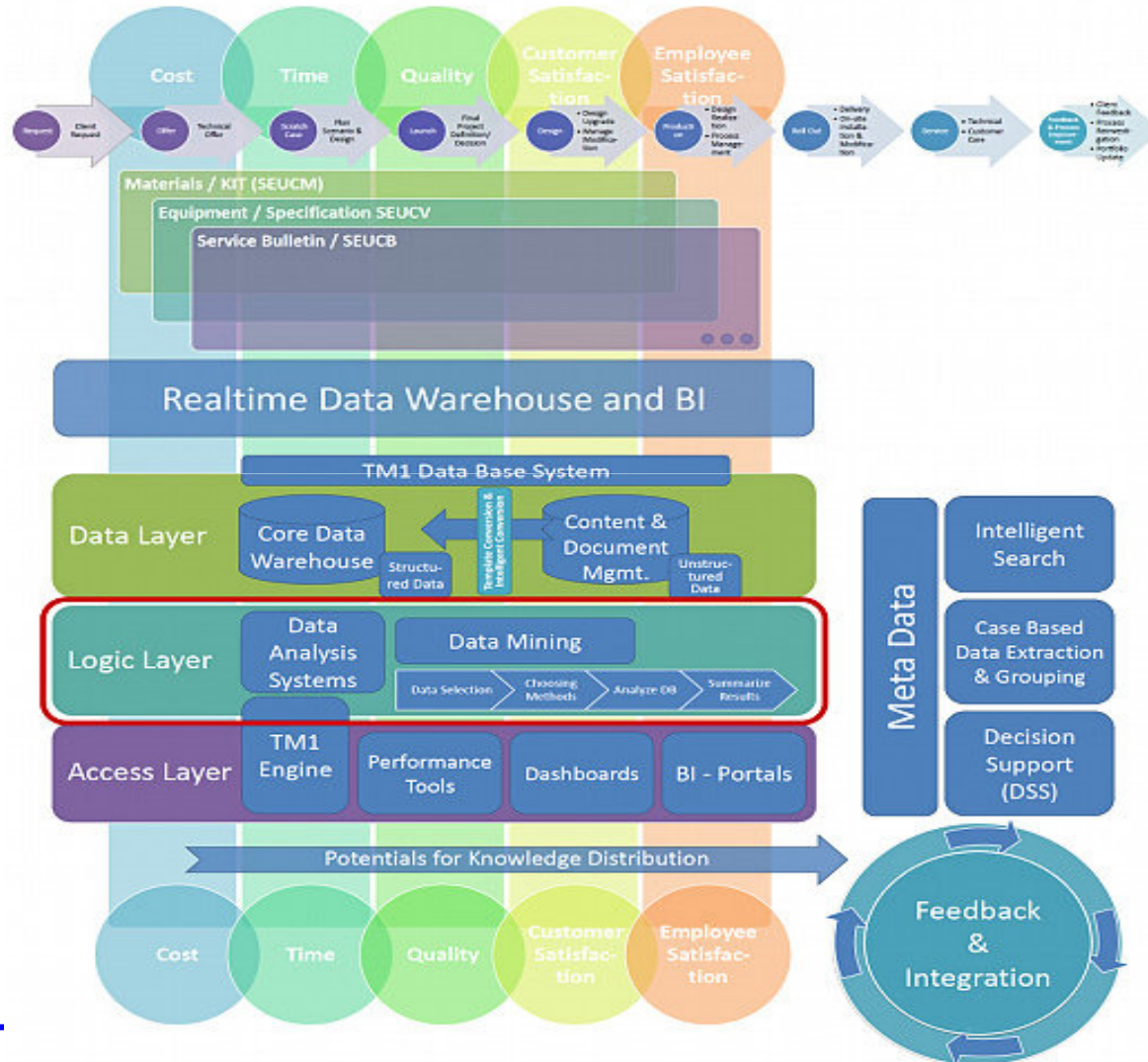


Projekte

Multi Domain Strategy for Data Integration of Business Process Data as a Controlling Instrument with Applied Data Mining & Visualization Methods



REPLIC - Retrieval and Extraction Process for Lasting Information Channeling





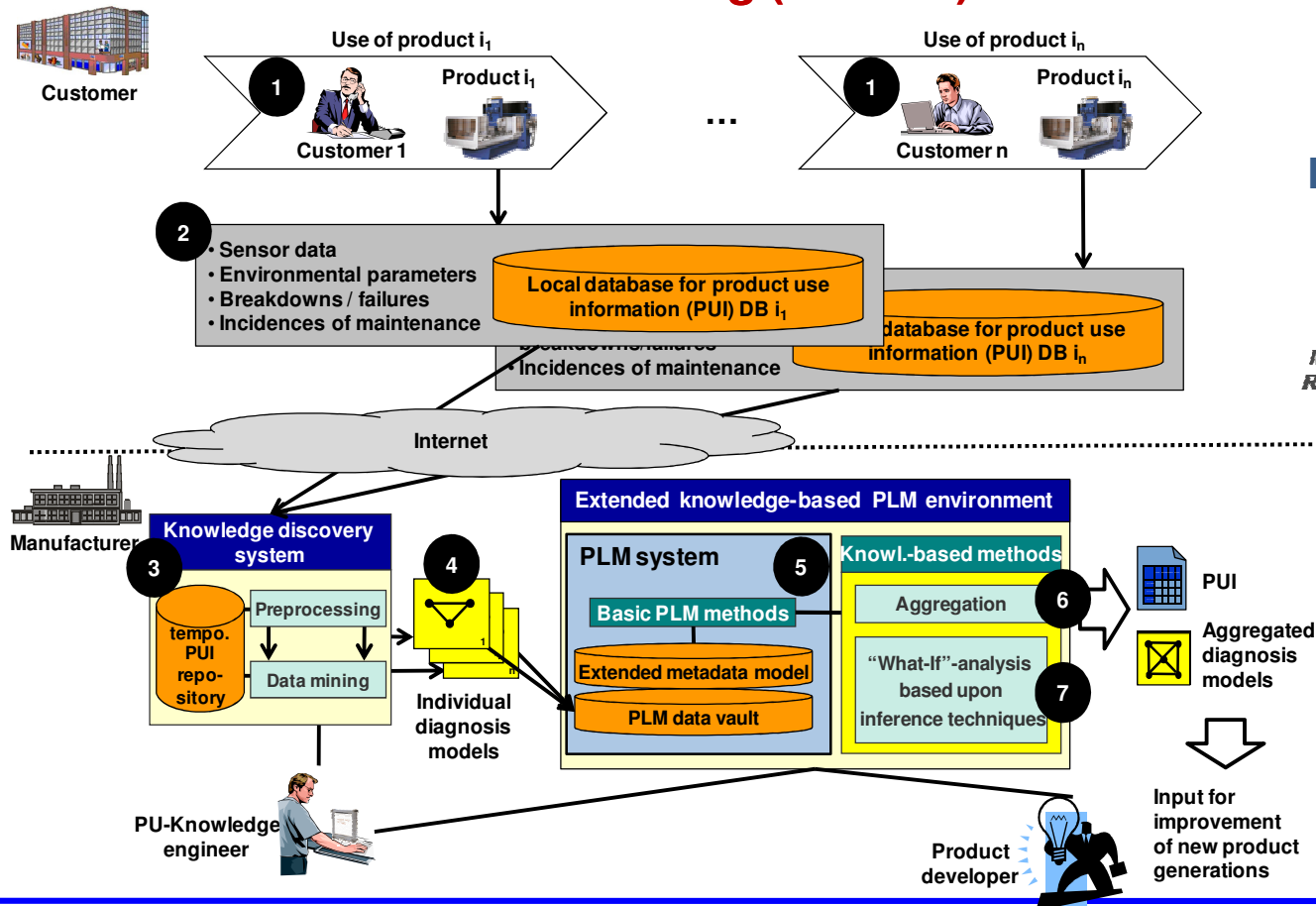
Projekte

Erweiterung des Product Lifecycle Managements durch wissensbasierte Rückführung von Produktnutzungs-Informationen in die Produktentwicklung (WiRPro)



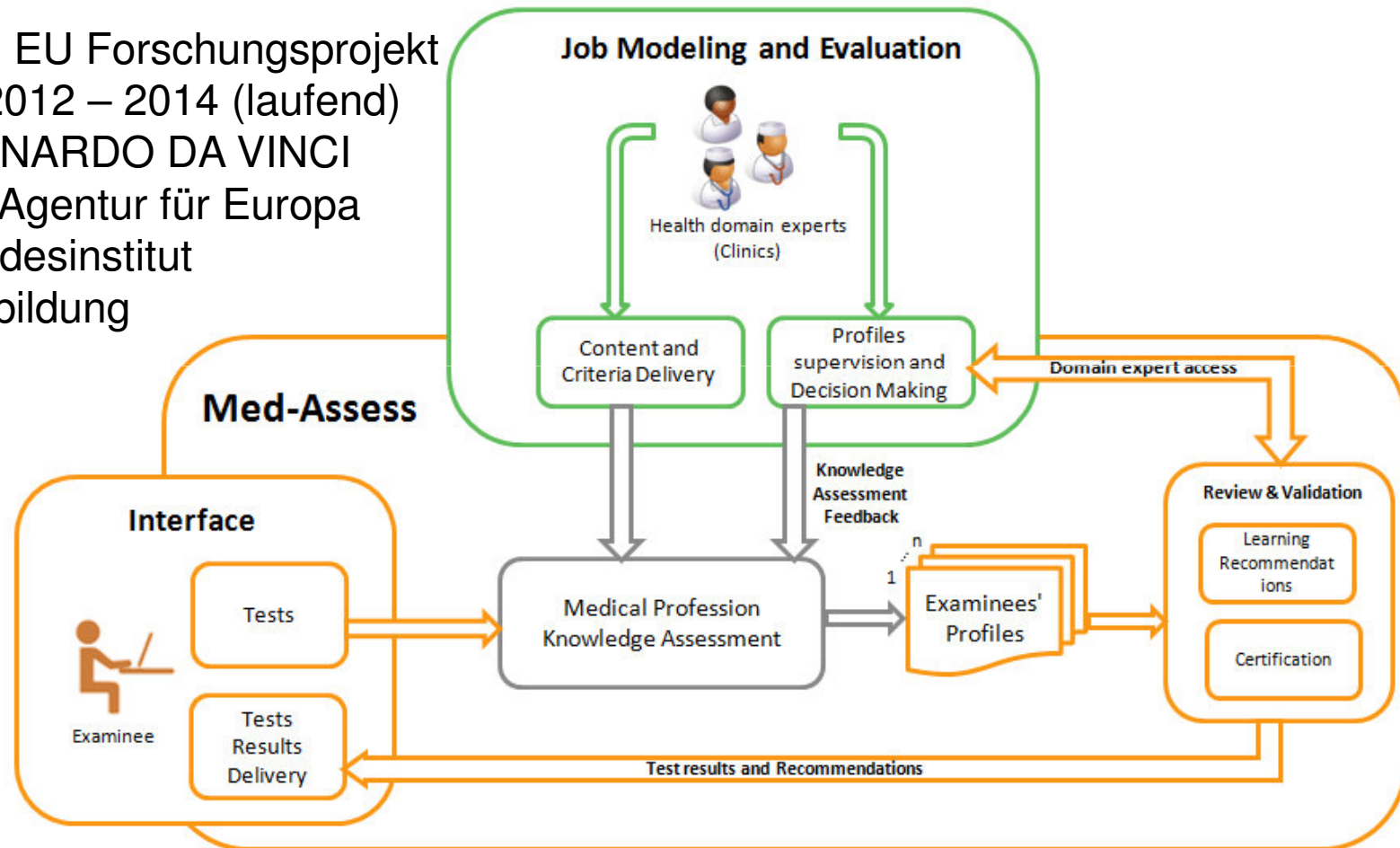
(2009-2012)

Projektpartner:



Med-Assess Adaptive Medical Profession Assessor

Kategorie: EU Forschungsprojekt
 Laufzeit: 2012 – 2014 (laufend)
 EU – LEONARDO DA VINCI
 Nationale Agentur für Europa
 beim Bundesinstitut
 für Berufsbildung



NeuroCare

Ganzheitliches Konzept zur Unterstützung und Entlastung von ambulanten Pflegekräften und Angehörigen bei der Betreuung von kognitiv beeinträchtigten Senioren



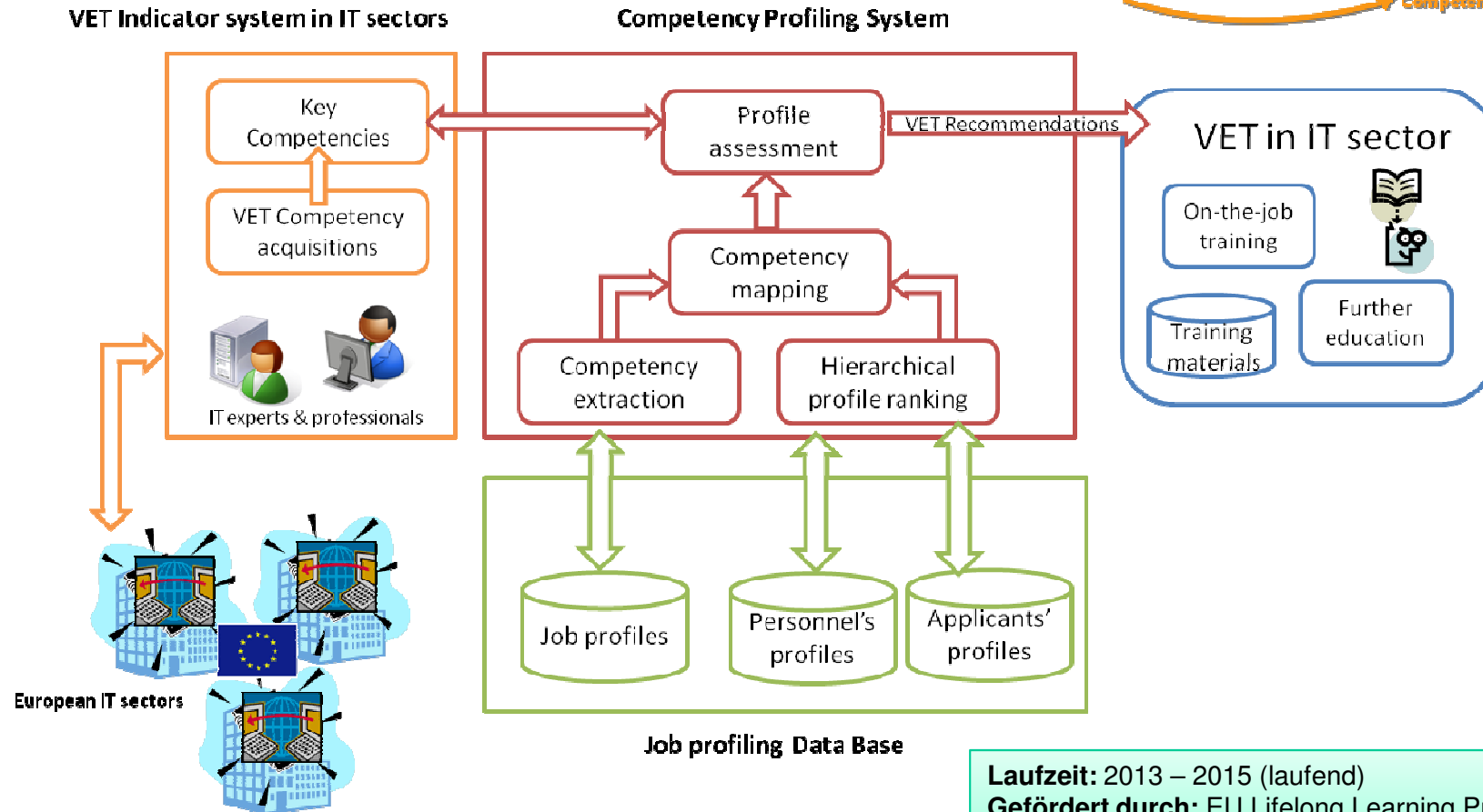
Das ganzheitliche Konzept NeuroCare besteht aus folgenden Elementen:

- Screening (Instrument zur Erfassung kognitiver Beeinträchtigungen)
- Training für Nutzer/Betroffene zum Erhalt und zur Stärkung der kognitiven Fähigkeiten
- Erkennung kognitiver Veränderungen (Krankheitsfortschritt/Verschlechterung)
- Bereitstellen von Informationen für alle Beteiligten (Betroffene, Angehörige, Ärzte etc.)
- Austausch und Kommunikation zwischen den Beteiligten
- Vernetzung zwischen allen beteiligten Akteuren in NeuroCare
- Umfassende Schulung, Fort- und Weiterbildung zu AAL-Beratern
- Serviceportal mit Beratern

Laufzeit: 2013-2016 (laufend)
Gefördert durch: BMBF (Ministerium für Bildung und Forschung)
Projekträger: VDI/VDE Innovation + Technik GmbH

ComProFITS

Competence Profiling Framework for IT Sector in Spain



Laufzeit: 2013 – 2015 (laufend)
Gefördert durch: EU Lifelong Learning Programme
Projekträger: NA-BiBB, Bonn, Deutschland

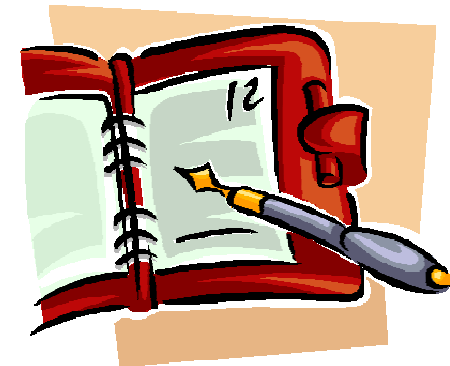
➤ **Vorlesungstermine:**

- Montag, 12:00 – 13:30, Roter Hörsaal (AR-D 5103)
- Montag, 14:15 – 15:45, Roter Hörsaal (AR-D 5103)

➤ **Vorlesungsfrei:** 21.04. (Ostern)

09.06. (Pfingsten)

➤ **Vorlesungsende:** 14.07.





Zur Vorlesung ...

➤ **Informationen: auf der WWW-Seite der Vorlesung**

http://www.eti.uni-siegen.de/ws/lehre/lehre14/ofp_2014/index.html

wichtige Ankündigungen und Termine!

➤ Vorlesungsfolien (im PDF-Format)

➤ zwei Versionen:

➤ Originalgröße mit Animationen

➤ verkleinert, ohne Animationen, mit Inhaltsverzeichnis

➤ zum Ausdrucken: Druckservice des Fachschaftsrats!

➤ Übungsblätter (über web-basiertes Übungssystem **DUESIE**)



Inhalt und Lernziele:

- Grundkenntnisse des Software-Entwurfs und vertiefte Kenntnisse in der Programmierung

- objektorientierte Modellierung (Analyse und Entwurf)
- objektorientierte Programmierung mit Java
 - inklusive wichtiger Teile der Java-Klassenbibliothek
- objektorientierte Entwurfsmuster
- funktionale Programmierung mit SML
 - Ziel: Kenntnis und Verständnis der grundlegenden Konzepte (neue Denkweisen!)
 - weniger: Nutzung von SML als Programmiersprache



Zielsetzung:

- Anwendung und Vertiefung der Vorlesungsinhalte
- erste Erfahrungen mit objektorientierter Analyse, Entwurf und Programmierung (in Java)
- Verinnerlichung der objektorientierten und der funktionalen Denkweise

Ihre Aufgaben:

- Nacharbeiten des Vorlesungsstoffs
 - allein oder in Gruppen; auch mit Literatur!
- Bearbeitung der Übungsaufgaben
 - allein oder ggf. auch in Gruppen; zu Hause oder im Labor
 - aber: jeder muss **EIGENE** Lösung abgeben!!!
- aktive Teilnahme an den Übungen
 - hier ist auch Platz für Fragen zur Vorlesung



Durchführung der Tutorübungen:

- Sie werden aktiv und erklären Ihre Lösungen!
- die Tutoren machen auf Fehler aufmerksam und beantworten Fragen
 - Das "Vorrechnen" der Aufgaben ist nicht Sinn der Übung
- es gibt bewußt keine ausgearbeiteten Musterlösungen

Differenzierte Übungsaufgaben und -gruppen:

- für Programmieranfänger (z.B. Erstsemester)
 - einfache Aufgaben im Klausurstil
- für Studenten mit etwas Programmiererfahrung
 - etwas komplexere / interessantere Aufgaben



Das Team:

- Übungsleitung: **Dr. Andreas Hoffmann**, Raum H-B 8405
- Tutoren: Namen und Email-Adressen siehe WWW-Seite

Termineinteilung:

- jetzt: Festlegung der Übungszeiten
 - voraussichtlich 10-12 Übungsgruppen
- Anmeldung zu den Übungsgruppen über DUESIE:
ab Mittwoch, den 16.04.14, ab 18 Uhr

Übungsbeginn:

- Mo, 21.04.2014 Beginn der Bearbeitung 1. Übungsblatt
- Mi, 23.04.2014 Organisatorisches, Übungsbeginn

Vorläufige Termine für die Übungen ...



OFP/(EI-2) - SoSe 14					
	Montag	Dienstag	Mittwoch	Donnerstag	Freitag
08-10	Gruppe 1 H-C7324	Gruppe 2 H-C7324	Gruppe 4 H-F 001	Gruppe 6 H-C 6336/37	Gruppe 8 H-C 6336/37
10-12	Gruppe 12 H-B8409/10	Gruppe 14 H-A 8107	- kein Raum zur Verfügung -	Gruppe 7 H-C 7324	Gruppe 11 H-B8409/10
12-14	Vorlesung	- kein Raum zur Verfügung -	- kein Raum zur Verfügung -	Gruppe 13 H-B 8409/10	Gruppe 9 H-C7324
14-16	Vorlesung	Gruppe 16 H-B8409/10	Gruppe 15 H-B 8409/10	Gruppe 18 H- A 8107 (?)	Gruppe 19 H- A 8107 (?)
16-18	Gruppe 17 H- A 8107 (?)	Gruppe 3 H-F 112	Gruppe 5 H-C7324	Gruppe 10 H-C 7324	- kein Tutor zur Verfügung -



Anmeldung zu den Übungsgruppen:

- http://www.eti.uni-siegen.de/ws/lehre/lehre14/ofp_2014/index.html
 - “Objektorientierte und Funktionale Programmierung (OFP)” → Übungsblatt 0
 - DUESIE-Passwort für die Vorlesung: **Turing14**
 - Anmeldungen ab **16.04.2014, ab 18 Uhr** möglich
- Funktionen:
 - Anmeldung zu einer Übungsgruppe
 - Um- und Abmeldung nur nach Absprache mit Tutor
 - Auslastung der Übungsgruppen anzeigen
 - Angabe Ihrer Email-Adresse
 - für wichtige Nachrichten
- Teilnehmerzahl pro Gruppe ist auf 25 limitiert
 - aber: es gibt insgesamt genügend Plätze für alle!



➤ Web-basiertes Übungssystem DUESIE

➤ <https://duesie.bs.informatik.uni-siegen.de/>

➤ Bereitstellung der Übungsblätter

➤ Bearbeitung kann online oder offline erfolgen

➤ die Übungsblätter sind ggf. selbst auszudrucken

➤ Abgabe der Lösungen

➤ nur bis zum festgesetzten Abgabetermin!

➤ Bereitstellung der korrigierten Lösungen mit Bewertung

➤ Hinweise:

➤ das allererste (nicht bewertete) Übungsblatt wird direkt auf der Vorlesungsseite bereitgestellt

➤ Einweisung in **DUESIE** in der ersten Übungsstunde



- Kennungen im Linux-Pool H-C 8327 aus A&D sind noch gültig
 - Kennungen werden für alle Teilnehmer erstellt
 - Kennung und Passwort abholen
 - im Sekretariat: Fr. Wenderoth, H-A 8111
 - Kartenschlüssel verlängern lassen oder neu beantragen
 - im Sekretariat: Fr. Wenderoth, H-A 8111
 - Abholung dann bei Hr. Kiel, H-B 5404 !
 - ggf. werden feste Termine bekanntgegeben

- Programmieraufgaben sind auch zu Hause lösbar
 - Grundsätzlich ist nur Internetverbindung und Webbrowser nötig
 - benötigte Software: Editor, Java-SDK, SML-Compiler. eUML4DUESIE
 - über die WWW-Seite aus dem Internet ladbar
 - oder auf CD bei der Fachschaft
 - jeweils für Linux und Windows



- Zweistündige Kompaktklausur (120 Minuten)
 - kurz nach Vorlesungsende
 - **Termine werden noch bekannt gegeben (Anfang August)**
- Wiederholungsmöglichkeit voraussichtlich Anfang Oktober
- Termine, Räume etc. werden in Vorlesung u. WWW angekündigt
- Durchführung ohne Hilfsmittel, aber Java-Klassenreferenz und SML Kurzreferenz als Anlage (siehe WWW)
- Zum Bestehen sind $\geq 50\%$ der erreichbaren Punkte nötig
- Probeklausur (zweiteilig) ist im WWW verfügbar
- Bekanntgabe der Ergebnisse:
 - durch Aushang (neben H-A 8113) und im WWW
 - Einsichtnahme in die Klausur ist möglich (Termin im WWW)



Prüfungsvorleistungen

- **ALLE** Studierenden haben Übungspflicht!
- Die Studierenden die bereits eine Zulassung zur EI2-Klausur haben (aus vorherigen Semestern) haben **KEINE** Übungspflicht!
- Dazu müssen Sie mindestens 75% der Übungsblätter tauglich (d.h. "ausreichend oder besser") bearbeiten
- Wegen der differenzierten Übungsaufgaben:
 - die Übungsblätter enthalten ca. doppelt so viele Aufgaben als Sie bearbeiten müssen
 - Sie suchen sich davon diejenigen 50% aus, die Ihrem Niveau entsprechen
 - zum Bestehen eines Übungsblatts brauchen Sie daher nur 35% der maximal möglichen Punkte



LA-Studierende mit LPO 2003:

- **Lehramts-Studenten** benötigen nur einen Übungsschein
 - Sie müssen nicht an der Klausur teilnehmen!
- Voraussetzung dazu: **aktive** Teilnahme an der Übung
 - **Anwesenheit** bei mindestens 75% der Übungen
 - **Abgabe einer tauglichen Lösung** für mindestens 75% der Übungsblätter
 - wie auf Folie 15 beschrieben
 - zweimal im Semester: **Vorstellung der Lösung** einer Aufgabe in der Übung (mit "ausreichend oder besser" bewertet)
- Der Übungsschein selbst ist **unbenotet**

Studierende Bachelor-Lehramt:

➤ **KLAUSUR!!! (laut Modulhandbuch Informatik!)**



Verantwortung der Studierenden

- Informationen **aktiv** beschaffen
 - Vorlesungs-Webseite: Termine, Ankündigungen, Übungen, ...
 - Prüfungsamt: Anmeldefristen, Klausuranmeldung, ...
 - Bibliothek, ...
- Übungen **aktiv** bearbeiten
- Vorlesung **schritthaltend** nacharbeiten
 - auch mit weiterführender Literatur!
- Aufbau von Lerngruppen

- An der Universität läuft Ihnen niemand nach!



Häufige Fehleinschätzungen

- *”Es reicht, wenn ich unmittelbar vor der Klausur lerne“*
 - wenn Sie nicht am Ball bleiben, verlieren Sie sehr schnell den Anschluß!
- *”Ich muß die Übungen nicht selbst machen, die werden ja sowieso besprochen“ oder ”Ich verstehe die Musterlösung, also kann ich es doch“*
 - gerade Programmieren lernen Sie **nur** durch Übung, d.h. selber machen!!
- *”Hauptsache, ich bestehe die Prüfung irgendwie“*
 - die Programmierkenntnisse aus der EI II sind eine Grundlage für Ihr ganzes späteres Studium!



- Software-Entwicklung [1-2]
 - Vorgehensmodelle, Programmentwicklung, objektorientierte (OO) Modellierung

- Objektorientierte Analyse mit UML [2]
 - Objekte, Klassen, Attribute, Operationen
 - Assoziation, Aggregation, Komposition, Generalisierung
 - Modellierung dynamischer Abläufe

- Objektorientierter Entwurf mit UML und Java [3-4]
 - Datentypen, Parameter, Klassendefinition, Konstruktor und Destruktor, Methodenaufrufe, Polymorphie, Sichtbarkeiten, Schnittstellen und Pakete, ...

Zahlen in []: ungefähre Umfang in Doppelstunden



Inhalt der Vorlesung ...

- Objektorientierte Programmierung mit Java
 - Sprachkonstrukte von Java, Ausnahmen, Garbage Collection [2-3]
 - Java Collection Framework [2]
 - Dateien, Streams und Serialisierung [1-2]
 - Programmierung graphischer Bedienoberflächen [1-2]
 - Threads [1]

- Entwurfsmuster [1-2]

- Funktionale Programmierung [4-5]
 - Konzepte funktionaler Programmiersprachen, Rekursion, Funktionen höherer Ordnung, Typinferenz, polymorphe Funktionen, Termauswertung, ...



Basisliteratur



[Ba99] Helmut Balzert: *Lehrbuch Grundlagen der Informatik*, Spektrum-Verlag, 1999. (*)

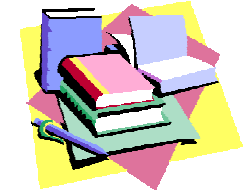
- Einführung in Java und UML
- deckt nicht-funktionalen Teil der Vorlesung i.W. ab

[Ba05] Heide Balzert: *Lehrbuch der Objektmodellierung*, Spektrum, 2005. (*)

- Vertiefung zu UML, objektorientierter Modellierung und Entwurfsmustern
- für EI II relevant: LE 1, 2, 3, 11, 12 und 14

[Er99] Martin Erwig: *Grundlagen funktionaler Programmierung*, Oldenbourg, 1999. (*)

- für EI II relevant: Kap. 1 und 2



Ergänzende Literatur zu Objektorientierung und UML

[BK03] David J. Barnes, Michael Kölling: *Objektorientierte Programmierung mit Java*, Pearson Studium, 2003.

- schöne objektorientierte Einführung in Java, mit Programmierwerkzeug BlueJ

[Oe05] Bernd Oesterreich: *Objektorientierte Softwareentwicklung – Analyse und Design mit der UML*, Oldenbourg, 2005. (+)

- Vertiefung zu UML, Objektorientierung, Analyse und Entwurf
- für EI II relevant: Kap. 2, 4.3, 4.4

[St05] Harald Störrle: *UML 2 für Studenten*, Pearson Studium, 2005.

- ausführliche Behandlung der vollständigen UML
- für EI II relevant: Kap. 1-5



Literatur ...



Ergänzende Literatur zu Java und Objektorientierung

[SB06] K. Sierra, B. Bates: *Java von Kopf bis Fuß*, O'Reilly, 2006.

- locker, witzig und motivierend geschrieben
- mit vielen Beispielen und Übungen

[GS02] Heinz-Peter Gumm, Manfred Sommer: *Einführung in die Informatik*, Oldenbourg, 2002. (*)

- viele Themen der Informatik, relativ knappe Darstellung
- für EI II relevant: Kap. 3 und 12
- als Nachschlagewerk

[Sc05] Reinhard Schiedermeier: *Programmieren mit Java*, Pearson Studium, 2005.

- auch als Nachschlagewerk geeignet



Ergänzende Literatur zu Java und Objektorientierung ...



[Bi01] Judy Bishop: *Java lernen*, Pearson Studium, 2001.

- relativ vollständig, eigenwilliger Aufbau, zum komplett Lesen, kein Nachschlagewerk

[HC05] Cay S. Hostmann, Gary Cornell: *Core Java 2*, Band 1 und 2, Addison-Wesley, 2005. (+)

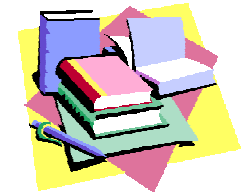
- erschöpfendes Nachschlagewerk zu Java (insgesamt über 2000 Seiten)

[LL05] John Lewis, William Loftus: *Java Software Solutions*, Addison Wesley, 2005.

- schön aufgemachtes Buch mit vielen Beispielen, auch als Nachschlagewerk geeignet



Ergänzende Literatur zu funktionaler Programmierung



[Kr02] F. Kröger: *Informatik I*, Skriptum zur Vorlesung, LMU München, WS02/03.
<http://www.pst.informatik.unimuenchen.de/personen/kroeger/infolWS0203.pdf>

- funktionale Programmierung mit SML

[Pa00] Peter Padawitz: *Grundlagen und Methoden funktionaler Programmierung*, Skript Univ. Dortmund, 2000. <http://fldit-www.cs.uni-dortmund.de/~peter/ProgNeu.ps.gz>

- für EI II relevant: i.W. Kap. 1 und 2

[Pa91] Laurence C. Paulson: *ML for the Working Programmer*, Cambridge University Press, 1991.

- das Standardwerk zu SML



Ergänzende Literatur zu Entwurfsmustern



[GH+96] E. Gamma, R. Helm, R. Johnson, J. Vlissides:
Entwurfsmuster, Addison Wesley, 1996. (+)

➤ das Standardwerk zu Entwurfsmustern

Anmerkungen

(*) Diese Bücher waren die Grundlage für die Ausarbeitung einiger Abschnitte der Vorlesung

(+) Diese Bücher wurden bei der Ausarbeitung der Vorlesung zur Vertiefung genutzt



Weitere Informationsquellen

- Ggf. weitere Literaturangaben im Verlauf der Vorlesung
 - Literaturangaben stehen auch im WWW
- Viele WWW-Ressourcen
 - siehe WWW-Seite zur Vorlesung
- Diskussionsforum: News-Gruppe zur EI I und EI II
 - <news://news.uni-siegen.de/uni-siegen.ei>

